

内网渗透 | NPS内网穿透工具的使用

原创 想走安全的小白 HACK学习呀

2020-11-23原文

一、NPS工具介绍

NPS工具是一款使用go语言编写的轻量级、功能强大的内网穿透工具。支持TCP、UDP流量转发，支持内网HTTP、SOCKS5代理，同时支持snappy压缩(节省带宽和流量)、站点保护、加密传输、多路复用、header修改等。同时还支持web图形化管理。该工具使用简单，相比于FRP，NPS是图形化界面，因此配置更加简单。

二、NPS工具原理介绍

注意:NPS工具的工作原理和FRP工具的工作原理相似，因此我们只需要对其中某一款工具的原理十分熟悉即可，由于之前写过一篇十分详细的FRP的工作原理和使用方法，因此，在这不再赘述，大家可以去看这篇文章：

内网渗透 | FRP代理工具详解

1.NPS客户端和服务端配置

NPS工具由NPS服务器端和NPS客户端组成，我们一般将NPS服务器端放在具有公网IP的VPS上，并且会开启一个端口等待NPS的客户端进行连接(一般会在NPS服务器的配置文件中说明)，而NPS的客户端一般会被放在我们已经拿下的内网主机上，我们会指定NPS服务器的客户端需要连接的NPS服务器的IP和端口，这样，我们就成功的将NPS的服务器端和NPS的客户端连接了起来。

2.通过NPS进行内网穿透

按照上面的方法，我们已经配置好了NPS服务，现在服务器端和客户端是可以连通的，但是，我们又怎么可以通过NPS进行内网穿透呢？其实，NPS是会在配置文件里面设置图形化界面的登录后台，我们通过登录NPS的后台，然后使用配置文件中设置的账号密码进行登录，登录后台之后，首先添加一个客户端，这个客户端会自动生成一个唯一验证密钥，我们需要在配置文件中输入这个唯一验证密钥，这样就可以将NPS的客户端和服务端连接起来了，随后我们可以根据我们的需求添加隧道，如HTTP隧道、SOCKS隧道等多条隧道，我们通过隧道设置的端口进行访问，即可访问到内网主机。

三、NPS配置

1.NPS下载链接

NPS下载链接:<https://github.com/ehang-io/nps/releases/tag/v0.26.9>

NPS官方说明文档:<https://ehang-io.github.io/nps/#/api>

2.NPS服务端配置

(1):查看服务器版本

```
arch
```

```
[root@iZbp13s58ab22ea4iuwr0dZ ~]# arch
x86_64
```

HACK学习呀

(2):下载对应版本的NPS服务器

 [linux_amd64_client.tar.gz](#)

 [linux_amd64_server.tar.gz](#)

 [linux_arm64_client.tar.gz](#)

 [linux_arm64_server.tar.gz](#)

HACK学习呀

(3):上传到服务器端进行解压

```
tar -xvzf linux_amd64_server.tar.gz
```

```
[root@iZbp13s58ab22ea4iuwr0dZ nps_server]# tar -xvzf linux_amd64_server.tar.gz
conf/nps.conf
conf/tasks.json
conf/clients.json
conf/hosts.json
conf/server.key
conf/server.pem
web/views/
web/views/index/
web/views/index/add.html
web/views/index/list.html
web/views/index/hedit.html
web/views/index/hlist.html
web/views/index/hadd.html
```


 HACK学习呀

(4):安装NPS

```
./nps install #Linux
```

```
nps.exe install #windows
```

```
[root@iZbp13s58ab22ea4iuwr0dZ nps_server]# ls
conf  linux_amd64_server.tar.gz  nps  web
[root@iZbp13s58ab22ea4iuwr0dZ nps_server]# ./nps install
2020/11/21 16:44:21 copy file ../nps_server/conf/clients.json to /etc/nps/conf/clients.json
2020/11/21 16:44:21 copy file ../nps_server/conf/hosts.json to /etc/nps/conf/hosts.json
2020/11/21 16:44:21 copy file ../nps_server/conf/nps.conf to /etc/nps/conf/nps.conf
2020/11/21 16:44:21 copy file ../nps_server/conf/server.key to /etc/nps/conf/server.key
2020/11/21 16:44:21 copy file ../nps_server/conf/server.pem to /etc/nps/conf/server.pem
2020/11/21 16:44:21 copy file ../nps_server/conf/tasks.json to /etc/nps/conf/tasks.json
2020/11/21 16:44:21 copy file ../nps_server/web/views/client/add.html to /etc/nps/web/views/client/add.html
2020/11/21 16:44:21 mkdir: /etc/nps/web/views/client/
```

 HACK学习呀

(5):查看配置文件

```
cd conf/
```

```
[root@iZbp13s58ab22ea4iuwr0dZ nps_server]# cd conf/
[root@iZbp13s58ab22ea4iuwr0dZ conf]# ls
clients.json  hosts.json  nps.conf  server.key  server.pem  tasks.json
```

 HACK学习呀

配置文件中的内容如下

```
appname = nps
```

```
#Boot mode(dev/pro)
```

```
runmode = dev
```

```
#HTTP(S) proxy port, no startup if empty
```

```
http_proxy_ip=0.0.0.0
```

`http_proxy_port=80` #域名代理http代理监听端口

`https_proxy_port=443`

#域名代理https代理监听端口(一般会修改这两个端口, 避免端口冲突)

`https_just_proxy=true`

#default https certificate setting

`https_default_cert_file=conf/server.pem`

`https_default_key_file=conf/server.key`

##bridge

`bridge_type=tcp` #客户端与服务端连接方式kcp或tcp

`bridge_port=8024`

#服务端客户端通信端口, 也就是说客户端通过访问服务端的这个端口可以进行连接

`bridge_ip=0.0.0.0`

Public password, which clients can use to connect to the server

After the connection, the server will be able to open relevant ports and parse related domain names according to its own configuration file.

`public_vkey=123`

#客户端以配置文件模式启动时的密钥, 设置为空表示关闭客户端配置文件连接模式

#Traffic data persistence interval(minute)

#Ignorance means no persistence

`#flow_store_interval=1`

#服务端流量数据持久化间隔，单位分钟，忽略表示不持久化

*# Log Level LevelEmergency->0 LevelAlert->1 LevelCritical->2
LevelError->3 LevelWarning->4 LevelNotice->5 LevelInformational->6 LevelDebug->7*

`log_level=7` *#日志输出级别*

`#log_path=nps.log`

#Whether to restrict IP access, true or false or ignore

`#ip_limit=true` *#是否限制ip访问，true或false或忽略*

`#p2p`

`#p2p_ip=127.0.0.1` *#服务端IP，使用p2p模式必填*

`#p2p_port=6000` *#p2p模式开启的udp端口*

`#web`

`web_host=a.o.com`

`web_username=admin` *#web界面管理账号*

`web_password=123` *#web界面管理密码*

`web_port = 8080` *#web管理端口，通过访问该端口可以访问NPS后台*

`web_ip=0.0.0.0`

`web_base_url=` *#web管理主路径，用于将web管理置于代理子路径后面*

`web_open_ssl=false`

`web_cert_file=conf/server.pem`

```
web_key_file=conf/server.key

# if web under proxy use sub path. Like http://host/nps need
this.

#web_base_url=/nps


#Web API unauthenticated IP address(the len of auth_crypt_key
must be 16)

#Remove comments if needed

#auth_key=test          #web api 密钥

auth_crypt_key =1234567812345678
#获取服务端authKey时的aes加密密钥，16位


#allow_ports=9001-9009,10001,11000-12000


#Web management multi-user login

allow_user_login=false

allow_user_register=false

allow_user_change_username=false


#extension

allow_flow_limit=false

allow_rate_limit=false

allow_tunnel_num_limit=false

allow_local_proxy=false
```

```
allow_connection_num_limit=false
```

```
allow_multi_ip=false
```

```
system_info_display=false
```

```
#cache
```

```
http_cache=false
```

```
http_cache_length=100
```

```
#get origin ip
```

```
http_add_origin_header=false
```

```
#pprof debug options
```

```
#pprof_ip=0.0.0.0          #debug pprof 服务端IP
```

```
#pprof_port=9999          #debug pprof 端口
```

```
#client disconnect timeout
```

```
disconnect_timeout=60    #客户端连接超时，单位 5s，默认值 60，即  
300s = 5mins
```

注意:在上面的配置文件中，我们主要是要注意以下方面:

- ①:一般会修改域名代理的端口，避免端口冲突
- ②:NPS的web页面默认端口是8080，默认用户名密码是admin/123
- ③:NPS的服务端和客户端进行连接的默认端口是8024，这个端口可以进行修改，修改之后，在连接时注意使用修改后的端口
- ④:NPS服务端开启的端口(也就是我们需要访问的VPS的端口)不在配置文件中，需要我们web界面中进行配置

3.NPS客户端配置

(1):下载对应版本的NPS客户端

windows_amd64_client

名称	修改日期	类型	大小
conf	2020/11/21 17:28	文件夹	
npc.exe	2020/10/6 18:01	应用程序	11,951 KB

(2):客户端连接方式

①:使用vkey进行连接

Windows: `npc.exe -server=ip:port -vkey=服务端生成的key`

Linux: `./npc -server=ip:port -vkey=服务端生成的key`

②使用配置文件进行连接

windows: `npc.exe -config=npc配置文件路径`

linux: `./npc -config=npc配置文件路径`

(3):客户端配置文件

友情提示:这里将配置文件写出来主要是为了让大家了解配置文件的内容,如果觉得配置文件太过繁琐,大多数情况下只需要关注server_addr、conn_type、和vkey这三个参数即可。

[common]

server_addr=127.0.0.1:8024

conn_type=tcp

vkey=123

auto_reconnection=true

max_conn=1000

flow_limit=1000

rate_limit=1000

basic_username=11

basic_password=3

web_username=user

web_password=1234

crypt=true

compress=true

#pprof_addr=0.0.0.0:9999

disconnect_timeout=60

[health_check_test1]

health_check_timeout=1

health_check_max_failed=3

health_check_interval=1

health_http_url=/

health_check_type=http

health_check_target=127.0.0.1:8083,127.0.0.1:8082

[health_check_test2]

health_check_timeout=1

health_check_max_failed=3

health_check_interval=1

health_check_type=tcp

health_check_target=127.0.0.1:8083,127.0.0.1:8082

[web]

host=c.o.com

target_addr=[127.0.0.1:8083](#),[127.0.0.1:8082](#)

[tcp]

mode=tcp

target_addr=[127.0.0.1:8080](#)

server_port=[10000](#)

[socks5]

mode=socks5

server_port=[19009](#)

multi_account=multi_account.conf

[file]

mode=file

server_port=[19008](#)

local_path=/Users/liuhe/Downloads

strip_pre=/web/

[http]

mode=httpProxy

server_port=[19004](#)

[udp]

mode=udp

```
server_port=12253
```

```
target_addr=114.114.114.114:53
```

```
[ssh_secret]
```

```
mode=secret
```

```
password=ssh2
```

```
target_addr=123.206.77.88:22
```

```
[ssh_p2p]
```

```
mode=p2p
```

```
password=ssh3
```

```
[secret_ssh]
```

```
local_port=2001
```

```
password=ssh2
```

```
[p2p_ssh]
```

```
local_port=2002
```

```
password=ssh3
```

```
target_addr=123.206.77.88:22
```

注意:NPS的客户端启动有两种启动方式,一种是不需要配置文件,直接输入相关命令即可启动,另一种是使用配置文件启动NPS客户端。如果需要配置文件来启动NPS客户端,那么需要配置如下内容(其余内容可以忽略)。

```
server_addr    #服务端ip/域名:port
```

conn_type #与服务端通信模式(tcp或kcp)

vkey #服务端配置文件中的密钥

首先server_addr是需要填写NPS服务端的IP和端口, conn_type选择合适的类型(一般选择TCP), vkey的值设置为服务端配置文件的密钥。这样服务端和客户端就可以进行连接了。

四、NPS使用实例

1.NPS服务端配置

首先按照上面的内容在VPS上下载并安装NPS的服务端。

(1):修改NPS服务端配置

```
appname = nps
#Boot mode(dev|pro)
runmode = dev

#HTTP(S) proxy port, no startup if empty
http_proxy_ip=0.0.0.0
http_proxy_port=8000
https_proxy_port=4430
https_just_proxy=true
#default https certificate setting
https_default_cert_file=conf/server.pem
https_default_key_file=conf/server.key
```

为了避免冲突修改端口

 HACK学习呀

(2):重载配置文件

./nps reload

```
[root@iZbp13s58ab22ea4iuwr0dZ nps_server]# ./nps reload
```

这块加载失败了, 目前还不清楚原因。

(3):启动NPS服务端

./nps start

```
[root@iZbp13s58ab22ea4iuwr0dZ nps_server]# ./nps start
[root@iZbp13s58ab22ea4iuwr0dZ nps_server]#
```

 HACK学习呀

./nps stop #停止nps 服务

./nps restart #重启nps 服务

(4):访问NPS服务端

23.8080/login/index

NPS

一款轻量级、高性能、功能强大的内网穿透代理服务器

- 协议支持全面，兼容几乎所有常用协议，例如tcp、udp、http(s)、socks5、p2p、http代理...
- 全平台兼容(linux、windows、macos、群辉等)，支持一键安装为系统服务
- 控制全面，同时支持服务端和客户端控制
- https集成，支持将后端代理和web服务转成https，同时支持多证书
- 操作简单，只需简单的配置即可在web ui上完成其余操作
- 展示信息全面，流量、系统信息、即时带宽、客户端版本等
- 扩展功能强大，该有的都有了（缓存、压缩、加密、流量限制、带宽限制、端口复用等等）
- 域名解析具备自定义header、404页面配置、host修改、站点保护、URL路由、泛解析等功能
- 服务端支持多用户和用户注册功能

用户名

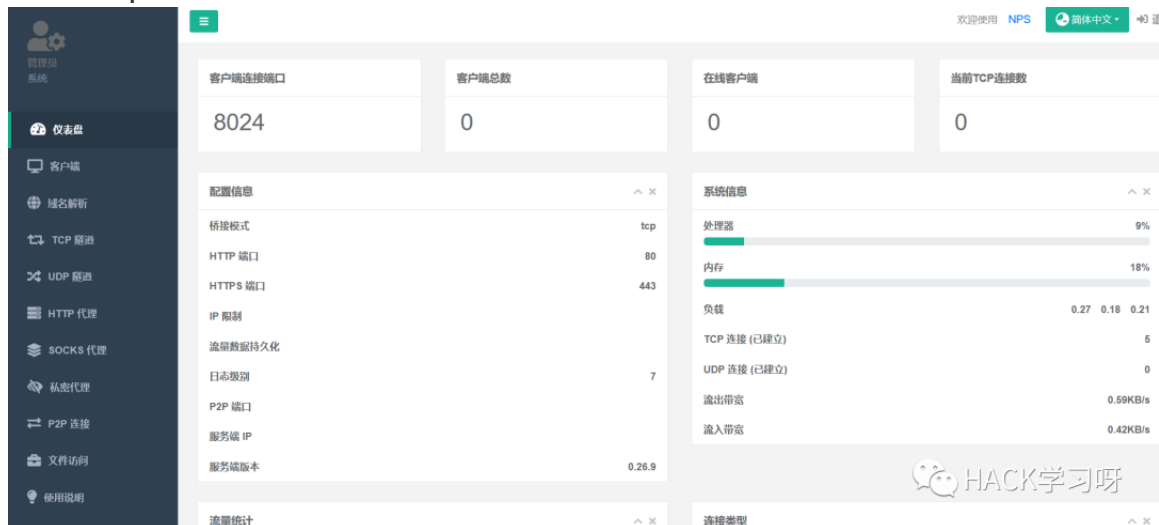
密码

登录

HACK学习呀

(5):使用账号密码登录成功

如下为nps控制台。



(6):新增一个客户端

这块新建的客户端主要是要使用生成的唯一验证密钥，通过这个唯一验证密钥才能将NPS的服务端和客户端连接起来，因此至少需要添加一个客户端。

管理员系统

仪表盘

客户端

域名解析

TCP 隧道

UDP 隧道

HTTP 代理

SOCKS 代理

私密代理

P2P 连接

文件访问

使用说明

新增客户端

备注
test

Basic 认证用户名
留空表示不受限制
仅限Socks5、Web、HTTP转发代理

Basic 认证密码
留空表示不受限制
仅限Socks5、Web、HTTP转发代理

唯一验证密钥
留空表示不受限制
唯一值，不填将自动生成

允许客户端通过配置文件连接
是

压缩
否

加密
否

HACK学习呀

新增

这个ID下面会用到

搜索

ID	备注	版本	唯一验证密钥	客户端地址	入口流量	出口流量	网速	状态	连接	选项	查看
2	test		dcom11fo7ztl43i		0B	0B	0B/S	开放	连接	<div></div>	<div>转发</div> <div>主机</div>

显示第 1 到第 1 条记录，总共 1 条记录

后面设置的Key会用到

HACK学习呀

(7):添加SOCKS代理

管理员系统

仪表盘

客户端

域名解析

TCP 隧道

UDP 隧道

HTTP 代理

SOCKS 代理

私密代理

P2P 连接

新增

模式
使用场景: 将公网服务器 1.1.1.1 的 8003 端口作为 SOCKS5 代理, 访问内网任意设备或者资源。
SOCKS 代理 选择SOCKS代理

客户端 ID
2 选择客户端ID, 该ID是上面生成的

备注
留空表示不受限制

服务端端口
2333 设置服务端端口,通过访问VPS的该端口即可访问内网

新增

HACK学习呀

新增

搜索

ID	客户端 ID	备注	模式	端口	目标 (IP:端口)	唯一标识密钥	状态	运行状态	客户端状态	选项
1	2		SOCKS 代理	2333			开放	开放		<div></div>

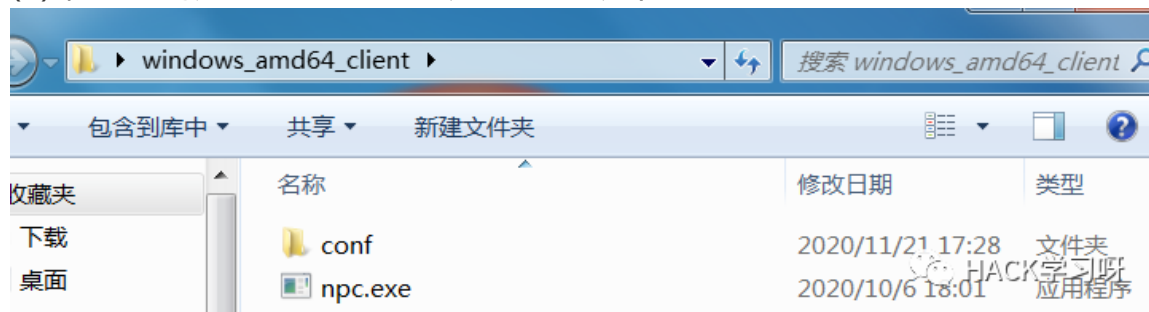
显示第 1 到第 1 条记录，总共 1 条记录

HACK学习呀

2.客户端配置

第一种方法:无配置文件

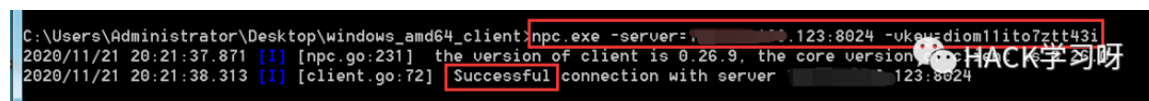
(1):将NPS对应版本的文件上传到内网主机中



(2):执行如下命令

Windows: `npc.exe -server=ip:port -vkey=服务端生成的key`

Linux: `./npc -server=ip:port -vkey=服务端生成的key`



客户端连接成功。

(3):使用浏览器设置代理访问内网主机

192.168.223.155

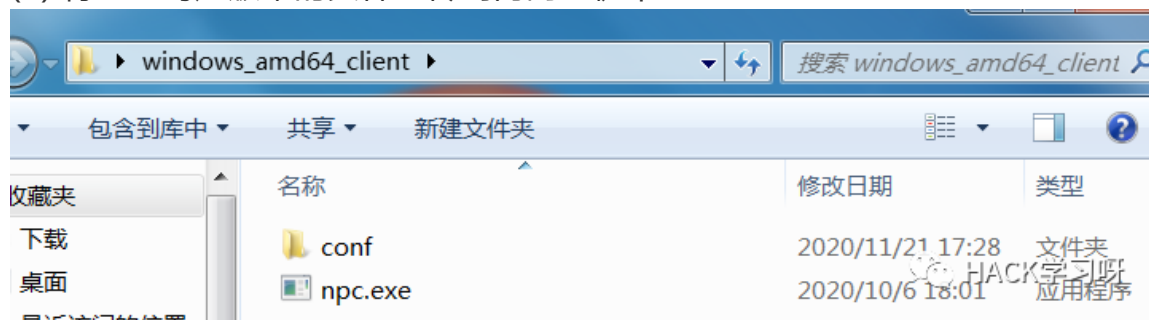


成功访问到内网主机的通达OA，但是在笔者测试的过程中发现似乎NPS的速度和稳定性不如FRP。

第二种方法:有配置文件

此模式使用nps的公钥或者客户端私钥验证，各种配置在客户端完成，同时服务端web也可以进行管理

(1):将NPS对应版本的文件上传到内网主机中



(2):修改配置文件


```
[common]
server_addr=192.168.223.106:8024 VPSIP:port
conn_type=tcp 连接类型
vkey=diom11ito7ztt43i 设置key
auto_reconnection=true
max_conn=1000
```

(3):执行如下命令

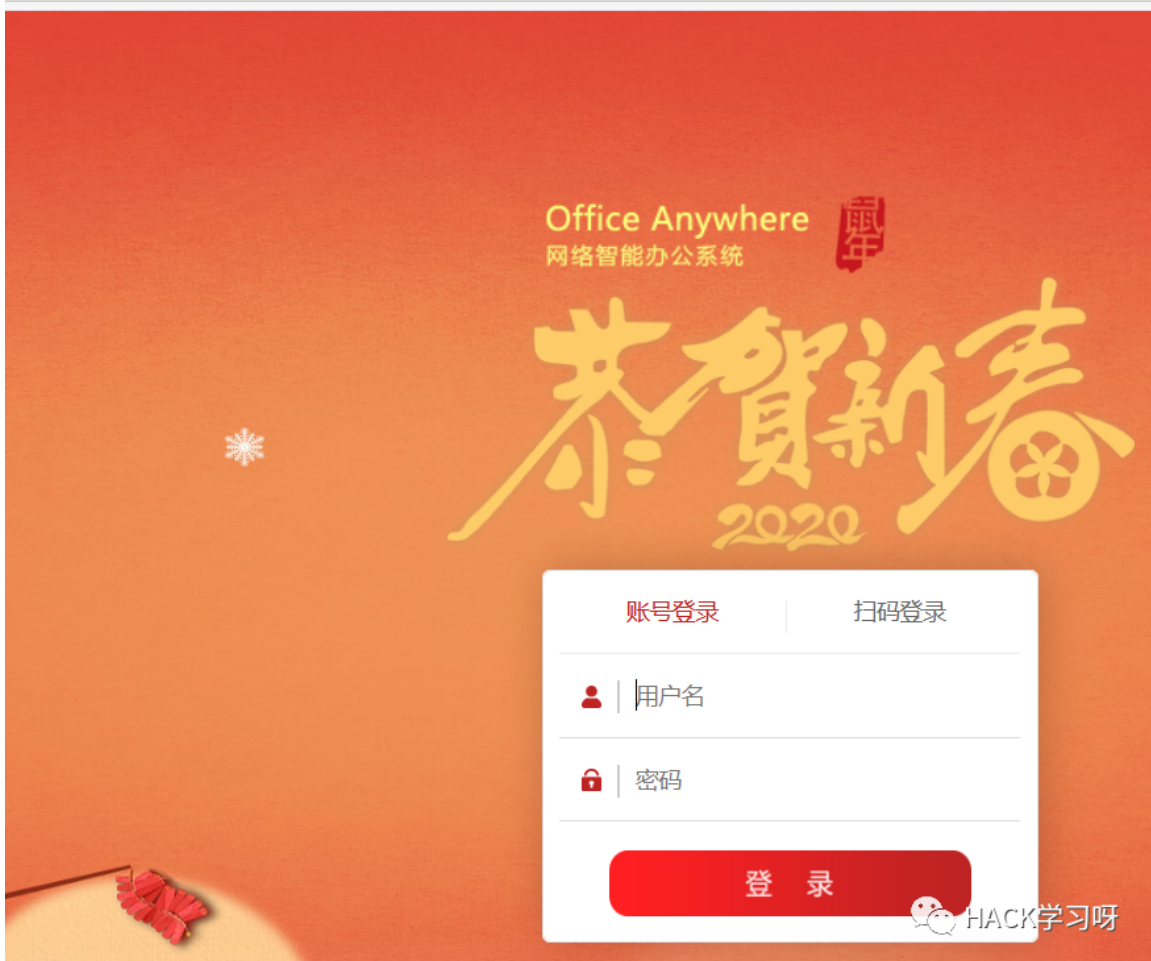
windows: npc.exe -config=npc配置文件路径

linux: ./npc -config=npc配置文件路径

```
C:\Users\Administrator\Desktop\windows_amd64_client>npc.exe -config=../conf/npc.conf
2020/11/21 20:40:14.206 [I] [npc.go:231] the version of client is 0.26.9, the control version of client
2020/11/21 20:40:14.221 [I] [control.go:97] Loading configuration file ../conf/npc.conf successfully
```

(4):浏览器通过代理成功访问目标主机

192.168.223.155



五、NPS其他场景使用

1.使用NPS代理SSH服务

(1):在内网主机上进行下载解压

```
root@kali:~/NPS# tar -zxvf linux_386_client.tar.gz
npc
conf/npc.conf
conf/multi_account.conf
root@kali:~/NPS# ls
conf linux_386_client.tar.gz npc
```

HACK学习呀

(2):在服务端创建一条TCP隧道

新增

模式
使用场景: 通过公网服务器1.1.1.1的8001端口, 连接内网机器10.1.50.101的22端口, 实现SSH连接。
TCP 隧道 隧道类型

客户端 ID
2 客户端ID

备注
留空表示不受限制

服务端端口
222 服务器端口

目标 (IP:端口)
192.168.223.160:22 内网主机的22端口

代理到本地可以只填写端口号, 只有TCP模式支持负载均衡

新增

HACK学习呀

ID	客户端 ID	备注	模式	端口	目标 (IP:端口)	唯一标识密钥	状态	运行状态	客户端状态	选项
16	2		TCP 隧道	222	192.168.223.160:22		开放	开放	在线	  

显示第 1 到第 1 条记录, 总共 1 条记录

HACK学习呀

(3):启动客户端连接服务端

./npc -server=ip:port -vkey=服务端生成的key

```
root@kali:~/NPS# ./npc -server=1.1.1.1:8001 -vkey=diom11ito7ztt431
2020/11/21 21:09:15.174 [I] [npc.go:231] the version of client is 0.26.9, the version of server is 0.26.0
2020/11/21 21:09:15.605 [I] [client.go:72] Successful connection with server 1.1.1.1:8001
```

HACK学习呀

(4):使用另一台VPS访问该VPS的222端口连接内网主机

```
root@iZ1x8w59kze53cZ:~# ssh root@116.62.106.123 -p 222 访问VPS的222端口
root@116.62.106.123's password:
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: Sat Nov 21 21:14:16 2020 from 192.168.223.160
root@kali:~# 成功登录内网主机kali
```

HACK学习呀

成功访问到内网主机。



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](#)

2020年性价比最高安全课程

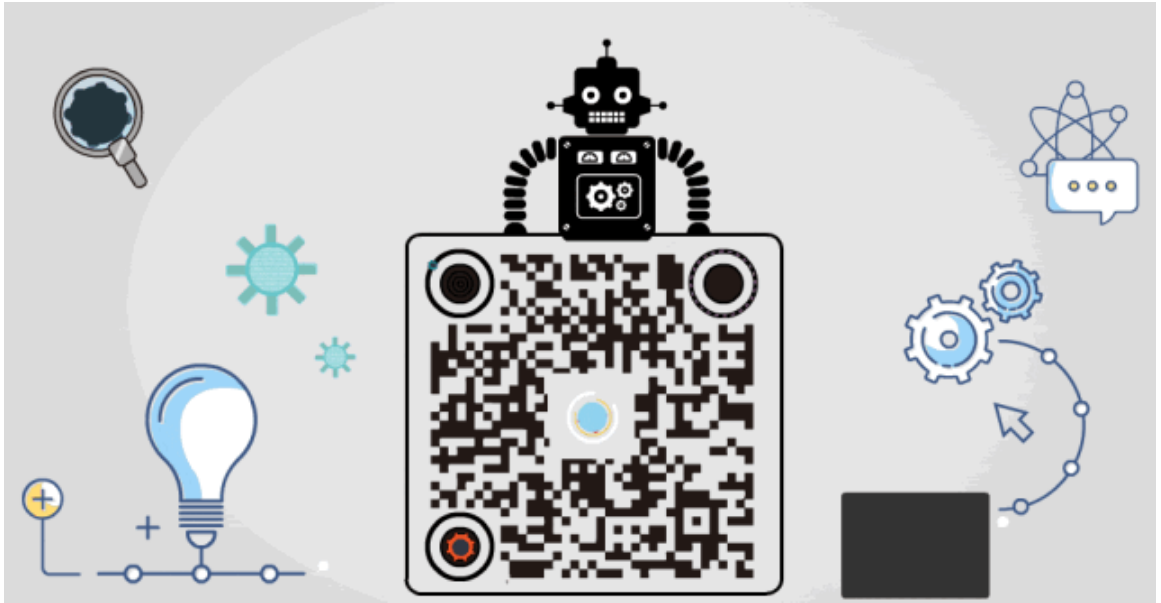
报名线上学习

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