使用FRP做穿透服务器，自定义自己的服务器，放弃Ngrok

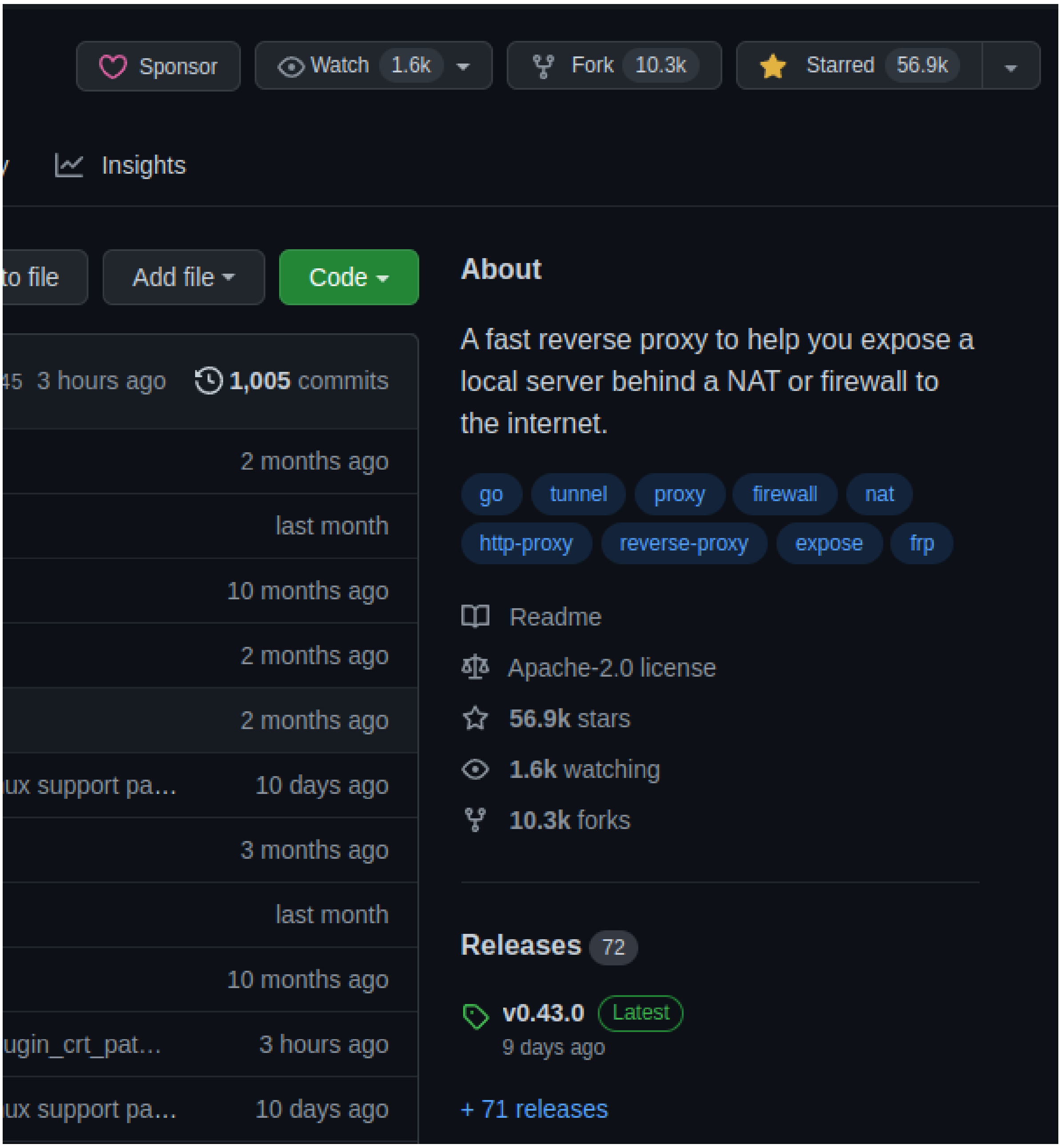
- ☑ 一台公网服务器，linux系统

FRP服务器搭建

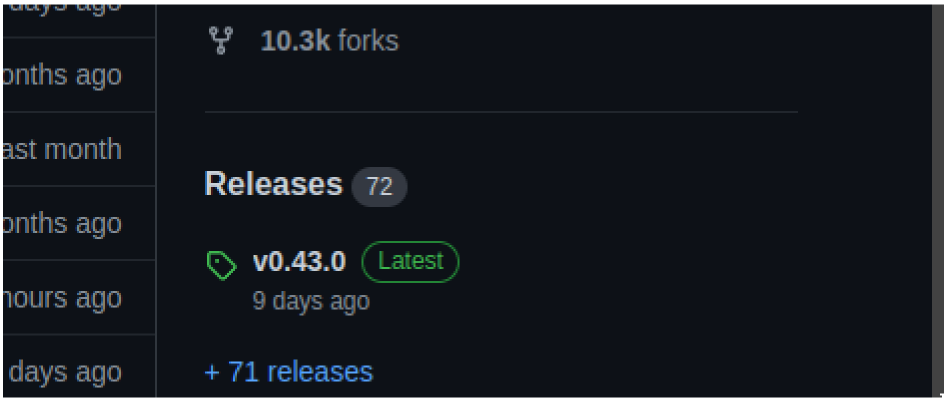
首先我们使用wget命令在服务器上下载最新的FRP

我们打开FRP的GitHub仓库

https://github.com/fatedier/frp



A fast reverse proxy to help you expose a local server behind a NAT or firewall to the internet.



onths ago
ast month
onths ago
hours ago
days ago

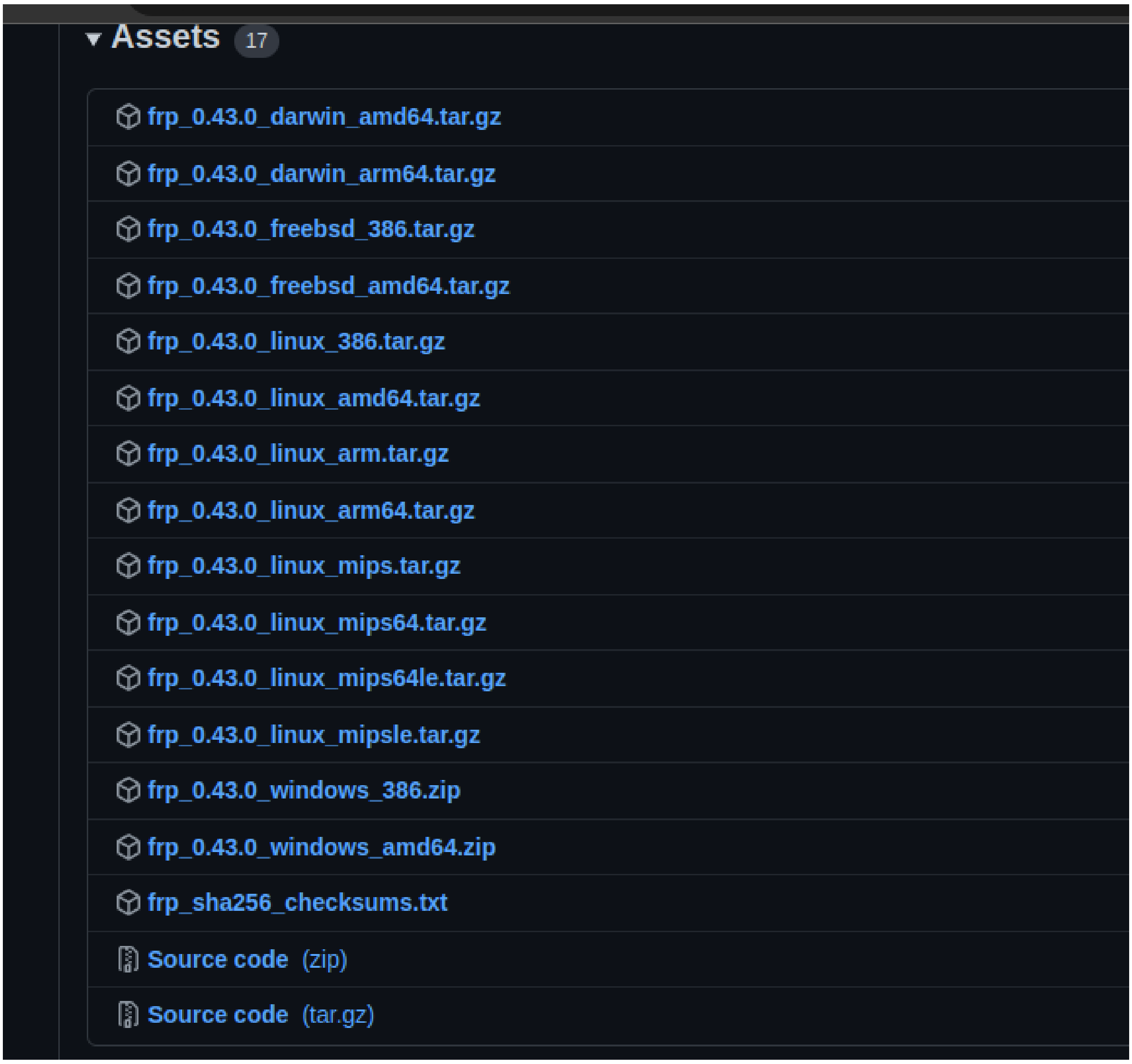
10.3k forks

Releases 72

v0.43.0 Latest
9 days ago

+ 71 releases

我们进入Releases



▾ Assets 17

- frp_0.43.0_darwin_amd64.tar.gz
- frp_0.43.0_darwin_arm64.tar.gz
- frp_0.43.0_freebsd_386.tar.gz
- frp_0.43.0_freebsd_amd64.tar.gz
- frp_0.43.0_linux_386.tar.gz
- frp_0.43.0_linux_amd64.tar.gz
- frp_0.43.0_linux_arm.tar.gz
- frp_0.43.0_linux_arm64.tar.gz
- frp_0.43.0_linux_mips.tar.gz
- frp_0.43.0_linux_mips64.tar.gz
- frp_0.43.0_linux_mips64le.tar.gz
- frp_0.43.0_linux_mipsle.tar.gz
- frp_0.43.0_windows_386.zip
- frp_0.43.0_windows_amd64.zip
- frp_sha256_checksums.txt
- Source code (zip)
- Source code (tar.gz)

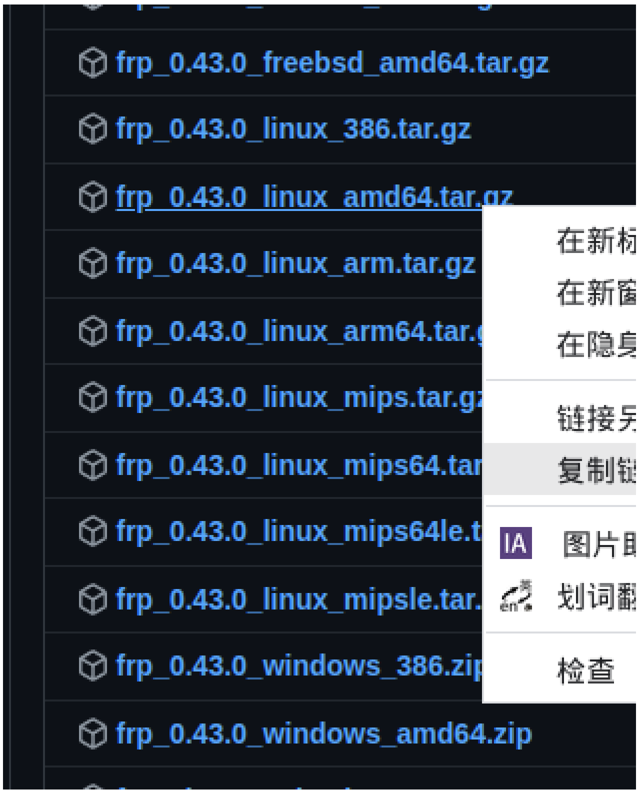
选择我们需要的版本，如果你的linux基于树莓派，是arm的，我们就下载arm版本

但是服务器和一般linux都是amd64版本！

所以我们下载`frp_0.43.0_linux_amd64.tar.gz`！如果你的客户端是树莓派

或者服务器是树莓派，arm架构的CPU，那你应该下载`frp_0.43.0_linux_arm64.tar.gz`！

我们右键要下载的文件，选择复制下载链接



frp_0.43.0_freebsd_amd64.tar.gz

frp_0.43.0_linux_386.tar.gz

frp_0.43.0_linux_amd64.tar.gz

frp_0.43.0_linux_arm.tar.gz

frp_0.43.0_linux_arm64.tar.g

frp_0.43.0_linux_mips.tar.gz

frp_0.43.0_linux_mips64.tar

frp_0.43.0_linux_mips64le.t

frp_0.43.0_linux_mipsle.tar.

frp_0.43.0_windows_386.zip

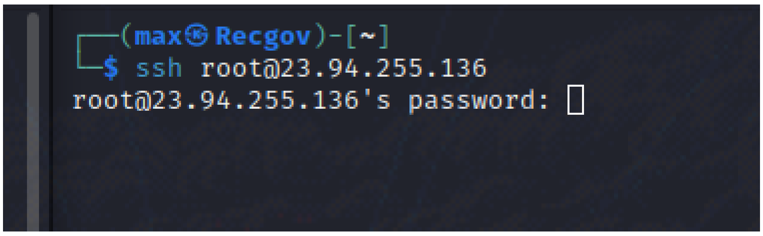
frp_0.43.0_windows_amd64.zip

之后我们连接上我们的服务器，使用ssh客户端，或者使用下面的命令

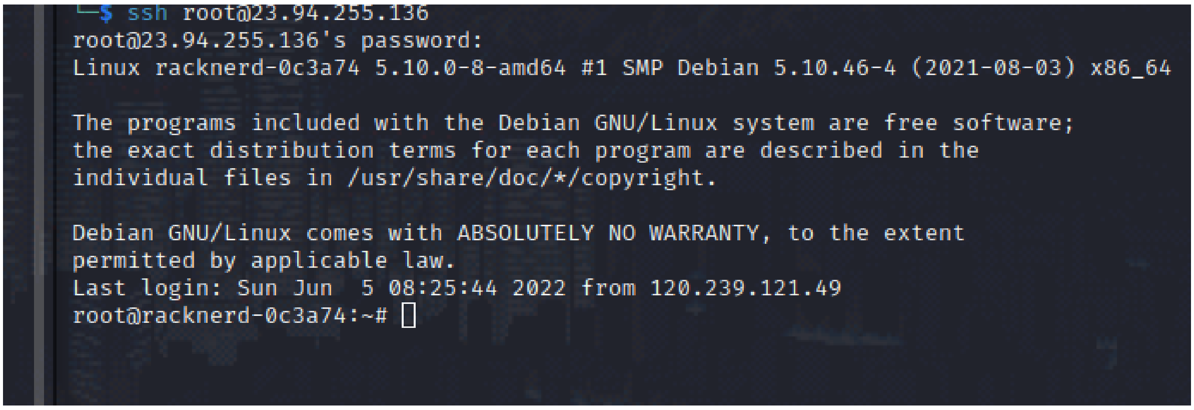
`ssh root@服务器ip`

如果它询问你是否接受证书`yes|no|..`我们输入yes并回车!

之后它就会请你输入服务器的登录密码!



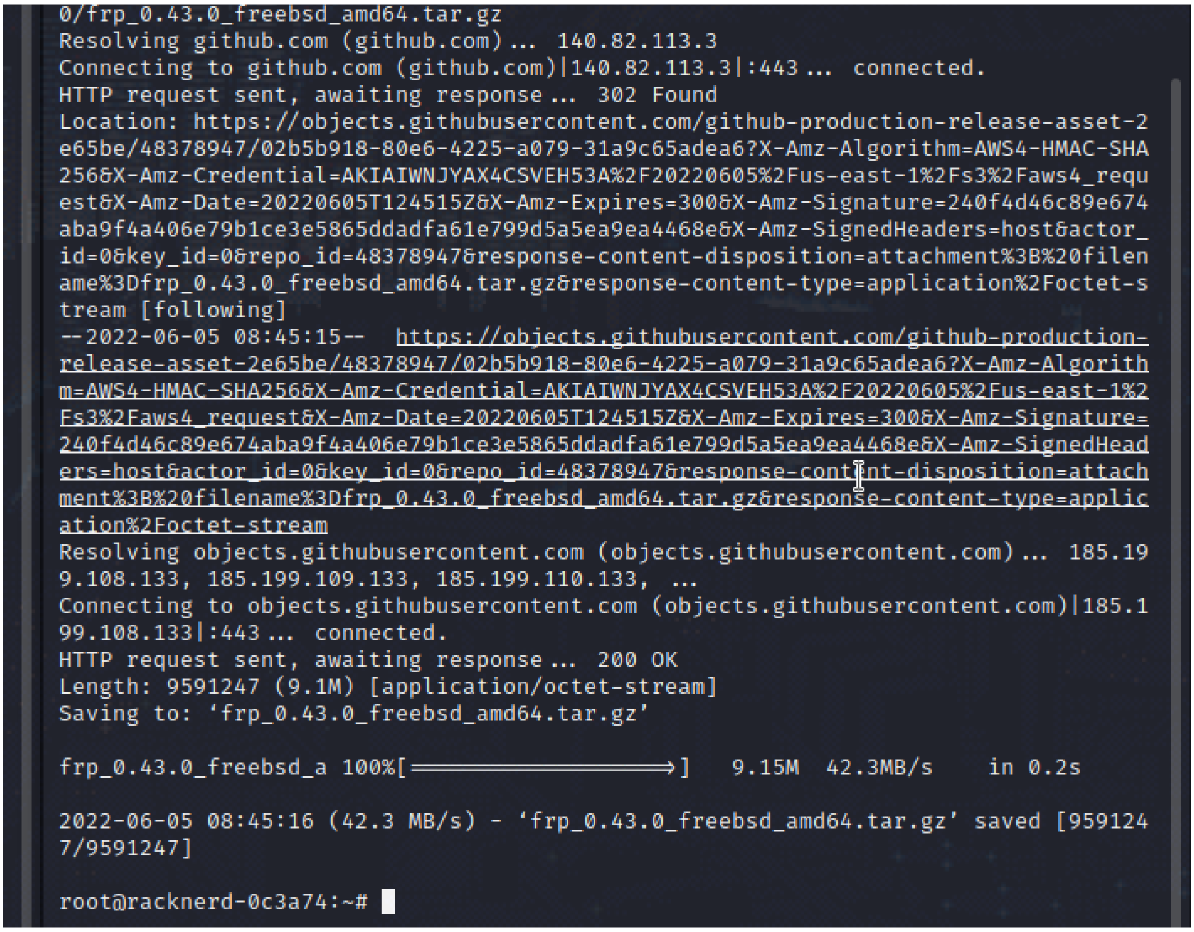
```
┌──(max㉿Recgov)-[~]
└─$ ssh root@23.94.255.136
root@23.94.255.136's password: 
```



```
└─$ ssh root@23.94.255.136
root@23.94.255.136's password:
Linux racknerd-0c3a74 5.10.0-8-amd64 #1 SMP Debian 5.10.46-4 (2021-08-03) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jun  5 08:25:44 2022 from 120.239.121.49
root@racknerd-0c3a74:~# 
```

登陆成功!

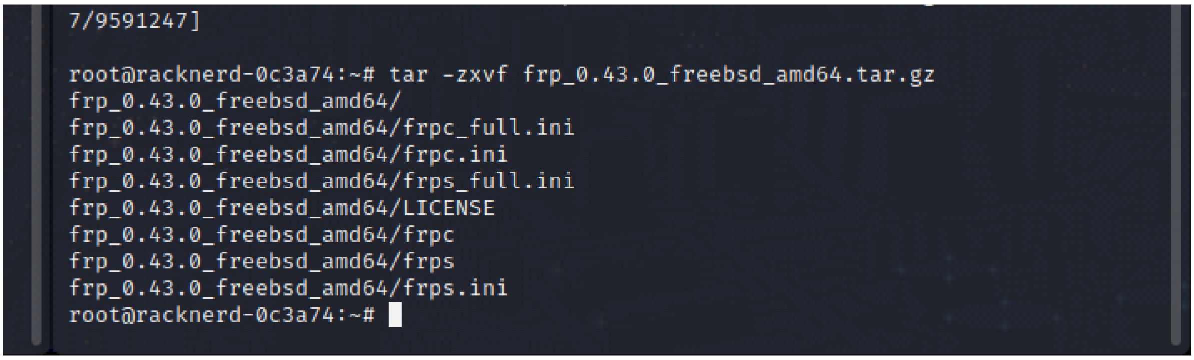
之后我们使用wget命令来下载FRP客户端



```
0/frp_0.43.0_freebsd_amd64.tar.gz
Resolving github.com (github.com)... 140.82.113.3
Connecting to github.com (github.com)|140.82.113.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2
e65be/48378947/02b5b918-80e6-4225-a079-31a9c65adea6?X-Amz-Algorithm=AWS4-HMAC-SHA
256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20220605%2Fus-east-1%2Fs3%2Faws4_requ
est&X-Amz-Date=20220605T124515Z&X-Amz-Expires=300&X-Amz-Signature=240f4d46c89e674
aba9f4a406e79b1ce3e5865ddadfa61e799d5a5ea9ea4468e&X-Amz-SignedHeaders=host&actor_
id=0&key_id=0&repo_id=48378947&response-content-disposition=attachment%3B%20filen
ame%3Dfrp_0.43.0_freebsd_amd64.tar.gz&response-content-type=application%2Foctet-s
tream [following]
--2022-06-05 08:45:15--  https://objects.githubusercontent.com/github-production-
release-asset-2e65be/48378947/02b5b918-80e6-4225-a079-31a9c65adea6?X-Amz-Algorith
m=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20220605%2Fus-east-1%2
Fs3%2Faws4_request&X-Amz-Date=20220605T124515Z&X-Amz-Expires=300&X-Amz-Signature=
240f4d46c89e674aba9f4a406e79b1ce3e5865ddadfa61e799d5a5ea9ea4468e&X-Amz-SignedHead
ers=host&actor_id=0&key_id=0&repo_id=48378947&response-content-disposition=attach
ment%3B%20filename%3Dfrp_0.43.0_freebsd_amd64.tar.gz&response-content-type=applic
ation%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.19
9.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.1
99.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9591247 (9.1M) [application/octet-stream]
Saving to: ‘frp_0.43.0_freebsd_amd64.tar.gz’

frp_0.43.0_freebsd_a 100%[===================>]   9.15M  42.3MB/s    in 0.2s

2022-06-05 08:45:16 (42.3 MB/s) - ‘frp_0.43.0_freebsd_amd64.tar.gz’ saved [959124
7/9591247]

root@racknerd-0c3a74:~# 
```

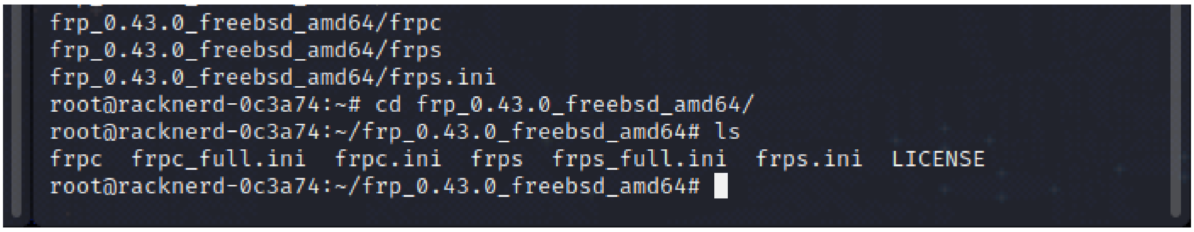
下载成功，我们解压到本地，使用如下命令

`tar -zxvf xxxxxxx.tar.gz`



```
7/9591247]

root@racknerd-0c3a74:~# tar -zxvf frp_0.43.0_freebsd_amd64.tar.gz
frp_0.43.0_freebsd_amd64/
frp_0.43.0_freebsd_amd64/frpc_full.ini
frp_0.43.0_freebsd_amd64/frpc.ini
frp_0.43.0_freebsd_amd64/frps_full.ini
frp_0.43.0_freebsd_amd64/LICENSE
frp_0.43.0_freebsd_amd64/frpc
frp_0.43.0_freebsd_amd64/frps
frp_0.43.0_freebsd_amd64/frps.ini
root@racknerd-0c3a74:~# 
```

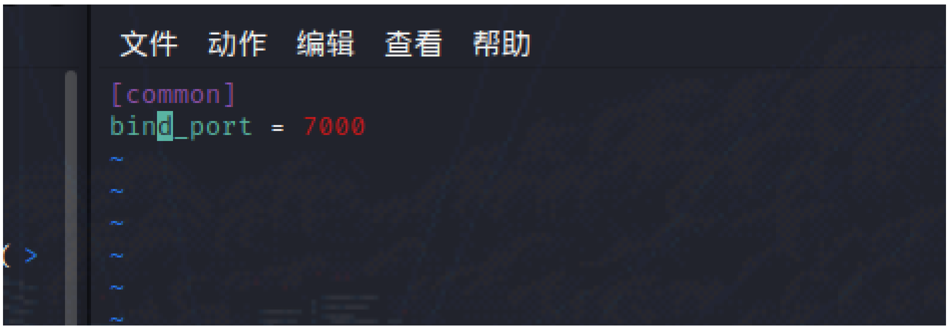
解压成功！我们进去刚刚解压出来的文件夹里



```
frp_0.43.0_freebsd_amd64/frpc
frp_0.43.0_freebsd_amd64/frps
frp_0.43.0_freebsd_amd64/frps.ini
root@racknerd-0c3a74:~# cd frp_0.43.0_freebsd_amd64/
root@racknerd-0c3a74:~/frp_0.43.0_freebsd_amd64# ls
frpc  frpc_full.ini  frpc.ini  frps  frps_full.ini  frps.ini  LICENSE
root@racknerd-0c3a74:~/frp_0.43.0_freebsd_amd64# 
```

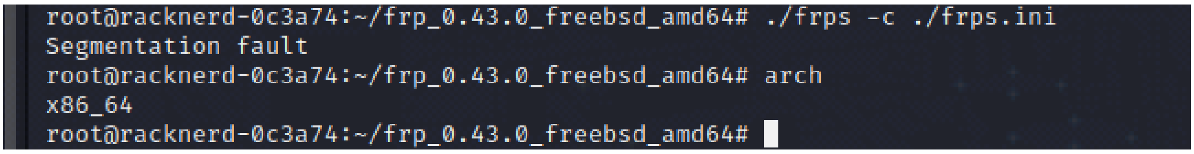
- 注意！:我们可以看到既有frpc也有frps，我们在服务器只需要使用到frps，s是server（服务器）的意思，而在我们本地linux，我们则需要使用frpc!

之后让我们来配置一下Frps.ini服务器文件！



```
[common]
bind_port = 7000
~
~
~
~
~
~
```

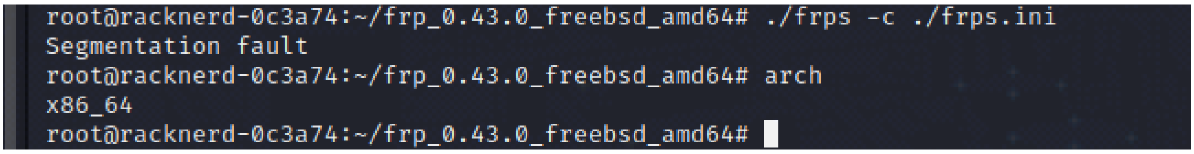
我们往里面添加如下参数



```
root@racknerd-0c3a74:~/frp_0.43.0_freebsd_amd64# ./frps -c ./frps.ini
Segmentation fault
root@racknerd-0c3a74:~/frp_0.43.0_freebsd_amd64# arch
x86_64
root@racknerd-0c3a74:~/frp_0.43.0_freebsd_amd64# 
```

```
[common]
bind_port = 7000          //要与后面配置的本地要穿透的客户端.ini文件统一，用于连接到服务器

dashboard_port = 9999    //网页访问的端口，我设置了9999，所以浏览器这样访问ip:9999
dashboard_user = admin    //网页登录账号
dashbard_pwd = 123456     //网页登录密码
```

之后我们保存后，使用 `./frps -c ./frps.ini` 命令即可启动



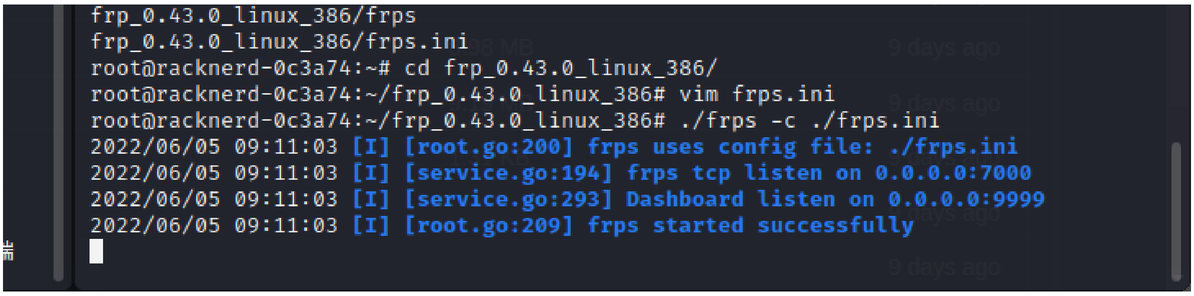
```
root@racknerd-0c3a74:~/frp_0.43.0_freebsd_amd64# ./frps -c ./frps.ini
Segmentation fault
root@racknerd-0c3a74:~/frp_0.43.0_freebsd_amd64# arch
x86_64
root@racknerd-0c3a74:~/frp_0.43.0_freebsd_amd64# 
```

我们发现启动失败，我们使用`arch`命令看看系统的架构

我们发现是x86_64，由此可知，我们的系统是X86架构，我们应该下载的版本是

`frp_0.43.0_linux_386.tar.gz`

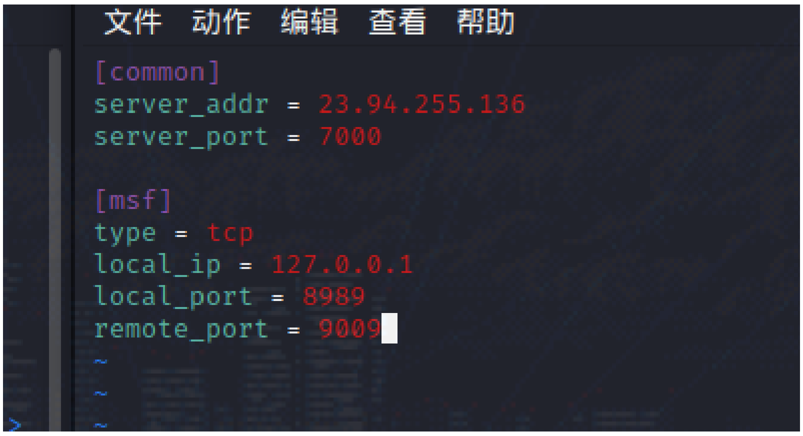
我们重新下载并重新配置一下



```
frp_0.43.0_linux_386/frps
frp_0.43.0_linux_386/frps.ini
root@racknerd-0c3a74:~# cd frp_0.43.0_linux_386/
root@racknerd-0c3a74:~/frp_0.43.0_linux_386# vim frps.ini
root@racknerd-0c3a74:~/frp_0.43.0_linux_386# ./frps -c ./frps.ini
2022/06/05 09:11:03 [I] [root.go:200] frps uses config file: ./frps.ini
2022/06/05 09:11:03 [I] [service.go:194] frps tcp listen on 0.0.0.0:7000
2022/06/05 09:11:03 [I] [service.go:293] Dashboard listen on 0.0.0.0:9999
2022/06/05 09:11:03 [I] [root.go:209] frps started successfully
```

运行成功！！

客户端部署

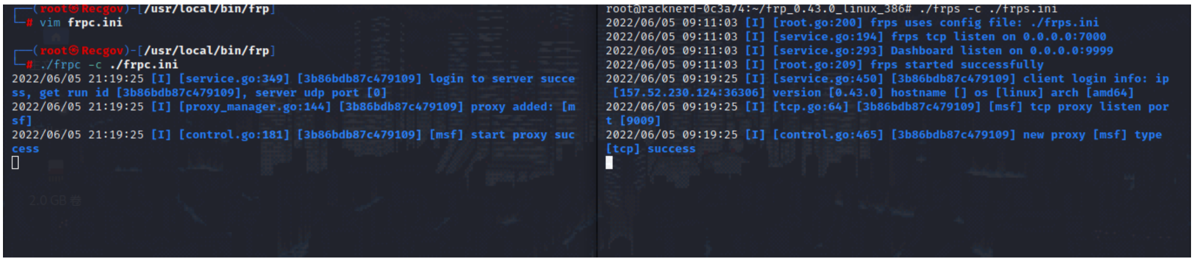


```
[common]
server_addr = 23.94.255.136
server_port = 7000

[msf]
type = tcp
local_ip = 127.0.0.1
local_port = 8989
remote_port = 9009
~
~
~
```

```
server_addr = 服务器IP
server_port = 用于与服务器通讯，需要和服务器的IP地址一致

type  = 看你需要转发什么流量，如果我使用TCP的payloads，我就使用TCP类型
local_port = 127.0.0.1 将本地端口转发
local_port = 将本地某个端口转发（注意，msf监听就是监听这里的127.0.0.1IP和转发出去的端口，但是生成的木马，需要指向到你的服务器IP和下面的端口）
remote_port = 服务器某一个端口
```

之后我们使用命令`./frpc -c ./frpc.ini`启动

之后我们在服务器这个ssh连接终端可以看到下面的内容



```
┌──(root㉿Recgov)-[/usr/local/bin/frp]
└─# vim frpc.ini

┌──(root㉿Recgov)-[/usr/local/bin/frp]
└─# ./frpc -c ./frpc.ini
2022/06/05 21:19:25 [I] [service.go:349] [3b86bdb87c479109] login to server succe
ss, get run id [3b86bdb87c479109], server udp port [0]
2022/06/05 21:19:25 [I] [proxy_manager.go:144] [3b86bdb87c479109] proxy added: [m
sf]
2022/06/05 21:19:25 [I] [control.go:181] [3b86bdb87c479109] [msf] start proxy suc
cess
```

```
root@racknerd-0c3a74:~/frp_0.43.0_linux_386# ./frps -c ./frps.ini
2022/06/05 09:11:03 [I] [root.go:200] frps uses config file: ./frps.ini
2022/06/05 09:11:03 [I] [service.go:194] frps tcp listen on 0.0.0.0:7000
2022/06/05 09:11:03 [I] [service.go:293] Dashboard listen on 0.0.0.0:9999
2022/06/05 09:11:03 [I] [root.go:209] frps started successfully
2022/06/05 09:19:25 [I] [service.go:450] [3b86bdb87c479109] client login info: ip
 [157.52.230.124:36306] version [0.43.0] hostname [] os [linux] arch [amd64]
2022/06/05 09:19:25 [I] [tcp.go:64] [3b86bdb87c479109] [msf] tcp proxy listen por
t [9009]
2022/06/05 09:19:25 [I] [control.go:465] [3b86bdb87c479109] new proxy [msf] type
[tcp] success
```

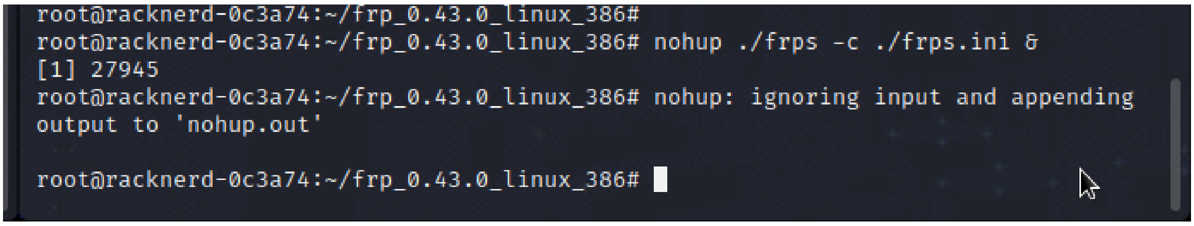
啊哈，显然我们已经连接成功了！

- 但我们服务器ssh连接断开后，就无法正常连接了

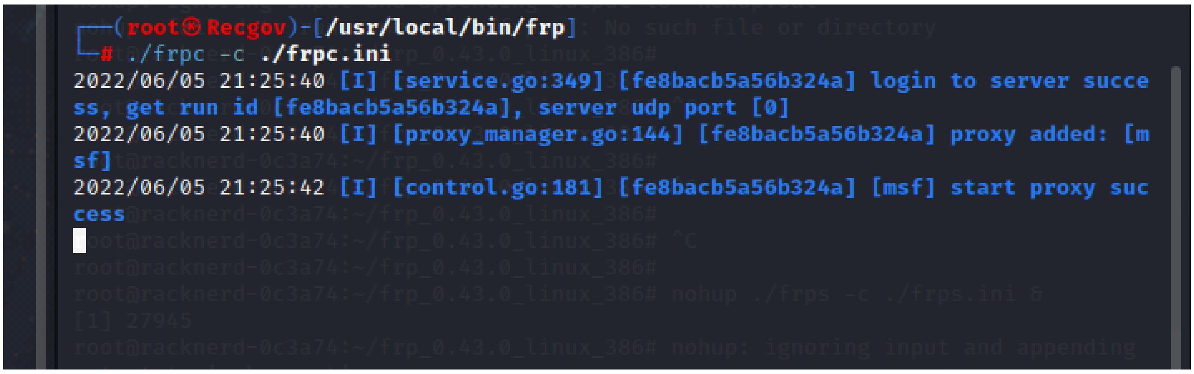
持久化命令，使得我们断开ssh后，命令仍在运行

我们使用下面的命令来完成

`nohup ./frps -c ./frps.ini &`



```
root@racknerd-0c3a74:~/frp_0.43.0_linux_386#
root@racknerd-0c3a74:~/frp_0.43.0_linux_386# nohup ./frps -c ./frps.ini &
[1] 27945
root@racknerd-0c3a74:~/frp_0.43.0_linux_386# nohup: ignoring input and appending
output to 'nohup.out'

root@racknerd-0c3a74:~/frp_0.43.0_linux_386#
```

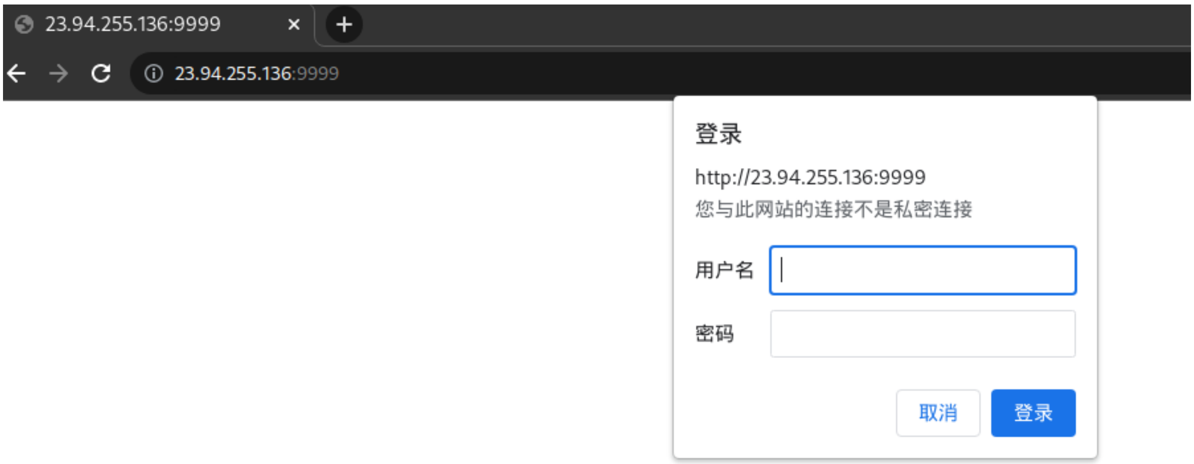


```
┌──(root㉿Recgov)-[/usr/local/bin/frp]
└─# ./frpc -c ./frpc.ini
2022/06/05 21:25:40 [I] [service.go:349] [fe8bacb5a56b324a] login to server succe
ss, get run id [fe8bacb5a56b324a], server udp port [0]
2022/06/05 21:25:40 [I] [proxy_manager.go:144] [fe8bacb5a56b324a] proxy added: [m
sf]
2022/06/05 21:25:42 [I] [control.go:181] [fe8bacb5a56b324a] [msf] start proxy suc
cess
```

我们可以看到，服务器即使断开连接，我们本地仍可以连接

我们打开WebFRP看看

IP:你自己设置的端口，我设置了9999



登录

http://23.94.255.136:9999
您与此网站的连接不是私密连接

用户名

密码

成功，我们登陆一下

之后我们即可在服务器列表内找到自己的服务器，比如我转发了TCP，所以我可以在TCP处看到自己的服务器处于在线状态！