

# FRP介绍和使用

## FRP介绍

FRP (Fast Reverse Proxy) 是一个使用 Go 语言开发的高性能的反向代理应用，可以轻松地进行内网穿透，对外网提供服务。FRP 支持 TCP、UDP、HTTP、HTTPS等协议类型，他支持点对点的转发，还有全流量转发，FRP最大的一个特点是使用SOCKS代理，而SOCKS是加密通信的，类似于做了一个加密的隧道，可以把外网的流量，通过加密隧道穿透到内网。效果有些类似于VPN。

下载地址: <https://github.com/fatedier/frp/releases>

v0.51.3






Latest

### Features

- Support Go 1.21.

### Assets

19

 <a href="#">frp_0.51.3_darwin_amd64.tar.gz</a>	11.9 MB	3 days ago
 <a href="#">frp_0.51.3_darwin_arm64.tar.gz</a>	11.4 MB	3 days ago
 <a href="#">frp_0.51.3_freebsd_386.tar.gz</a>	10.8 MB	3 days ago
 <a href="#">frp_0.51.3_freebsd_amd64.tar.gz</a>	11.4 MB	3 days ago
 <a href="#">frp_0.51.3_linux_386.tar.gz</a>	10.9 MB	3 days ago
 <a href="#">frp_0.51.3_linux_amd64.tar.gz</a>	11.4 MB	3 days ago
 <a href="#">frp_0.51.3_linux_arm.tar.gz</a>	10.8 MB	3 days ago
 <a href="#">frp_0.51.3_linux_arm64.tar.gz</a>	10.4 MB	3 days ago
 <a href="#">frp_0.51.3_linux_mips.tar.gz</a>	10.3 MB	3 days ago
 <a href="#">frp_0.51.3_linux_mips64.tar.gz</a>	10.1 MB	3 days ago
 <a href="#">Source code (zip)</a>		3 days ago
 <a href="#">Source code (tar.gz)</a>		3 days ago
<a href="#">Show all 19 assets</a>		

使用手册: <https://gofrp.org/docs/>

frp

文档 版本历史 Github 中文 站内搜索...

站内搜索...

文档

概览

安装

概念

示例

功能特性

参考

FAQ

文档

了解如何部署、使用 frp。

概览

一些概述，便于您快速的了解 frp。

安装

关于如何安装 frp 的说明。

概念

一些概念，理解它们有助于您更好地了解和使用 frp。

示例

这里包括多个常见的使用场景和配置示例，你可以用来亲自部署和体验这些示例。

功能特性

了解 frp 所支持的一些功能特性。

参考

FAQ

查看页面源码








编辑此页

添加子页面

提交项目问题

FRP分为两个文件

将 frps 及 frps.ini 放到具有公网 IP 的机器上。将 frpc 及 frpc.ini 放到处于内网环境的机器上。

 frpc	2023 12:02	文件	15,157 KB
 frpc.ini	2023 12:09	配置设置	1 KB
 frpc_full.ini	2023 12:09	配置设置	13 KB
 frps	2023 12:02	文件	18,401 KB
 frps.ini	2023 12:09	配置设置	1 KB
 frps_full.ini	2023 12:09	配置设置	6 KB
 LICENSE	2023 12:09	文件	12 KB

## FRP启动

### 服务端配置

Frps 默认给出两个服务端配置文件，一个是简版的 frps.ini，另一个是完整版本 frps\_full.ini，服务端一般就是搭建在公网的机器上（黑客的机器）

代理配置文件

```
# frps.ini
[common]
bind_port = 7000
# 默认配置中监听的是 7000 端口，可根据自己实际情况修改
```

代理启动文件

```
./frps -c ./frps.ini
```

使用FRPS 运行 frps.ini 配置文件

```
C:\Users\Administrator\Desktop\FRP>frps.exe -c frps.ini
2023/08/17 14:32:37 [I] [root.go:204] frps uses config file: frps.ini
2023/08/17 14:32:37 [I] [service.go:206] frps tcp listen on 0.0.0.0:7000
2023/08/17 14:32:37 [I] [root.go:213] frps started successfully
```

### 客户端配置

FRP 默认也给出两个客户端配置文件，一个是简版的 frpc.ini，另一个是完整版本 frpc\_full.ini。这里同样以简版的 frpc.ini 文件为例，假设 FRP 服务端所在服务器的公网 IP 为 X.X.X.X：

配置文件如下

```
# frpc.ini
[common]
server_addr = x.x.x.x
server_port = 7000
```

```
./frpc -c ./frpc.ini
```

# FRP的配置文件介绍

配置文件非常重要，需要什么功能就可以进行不同的配置，来进行代理

参考文档：<https://gofrp.org/docs/reference/server-configures/>

服务端：

## 基础配置

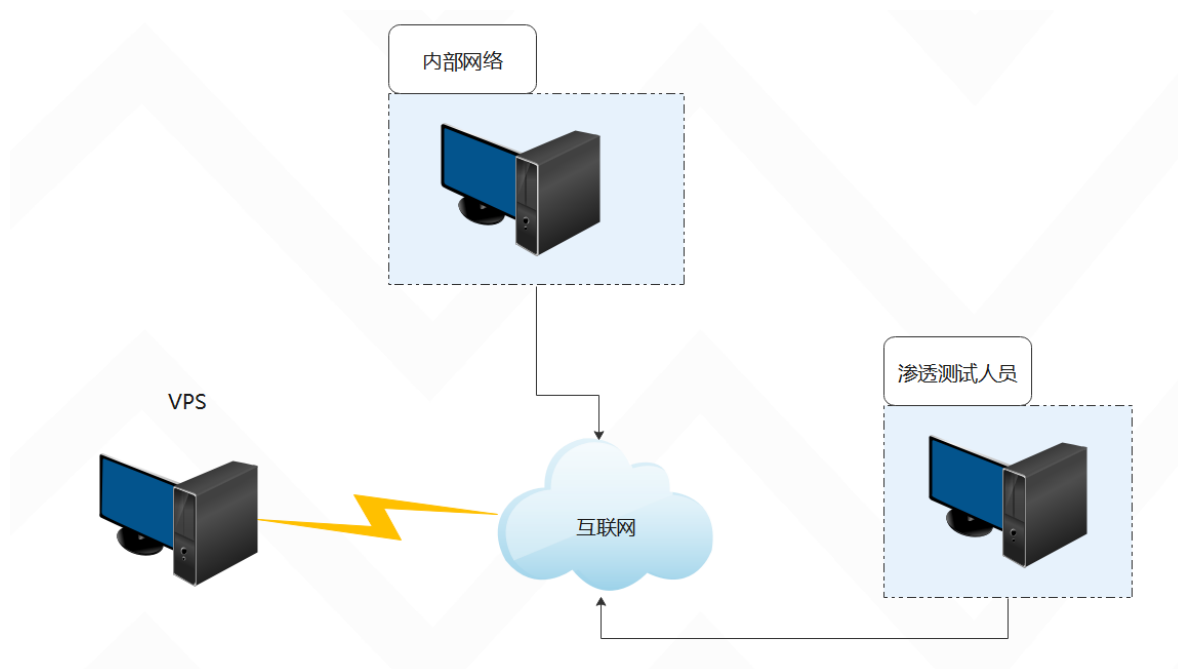
参数	类型	说明	默认值	可选值	备注
bind_addr	string	服务端监听地址	0.0.0.0		
bind_port	int	服务端监听端口	7000		接收 frpc 的连接
kcp_bind_port	int	服务端监听 KCP 协议端口	0		用于接收采用 KCP 连接的 frpc
quic_bind_port	int	服务端监听 QUIC 协议端口	0		用于接收采用 QUIC 连接的 frpc
quic_keepalive_period	int	quic 协议 keepalive 间隔，单位: 秒	10		
quic_max_idle_timeout	int	quic 协议的最大空闲超时时间，单位: 秒	30		
quic_max_incoming_streams	int	quic 协议最大并发 stream 数	100000		
proxy_bind_addr	string	代理监听地址	同 bind_addr		可以使代理监听在不同的网卡地址
log_file	string	日志文件地址	./frps.log		如果设置为 console，会将日志打印在标准输出中

客户端：

server_addr	string	连接服务端的地址	0.0.0.0		
server_port	int	连接服务端的端口	7000		
nat_hole_stun_server	string	xtcp 打洞所需的 stun 服务器地址	stun.easyvoip.com:3478		
connect_server_local_ip	string	连接服务端时所绑定的本地 IP			
dial_server_timeout	int	连接服务端的超时时间	10		
dial_server_keepalive	int	和服务端底层 TCP 连接的 keepalive 间隔时间，单位秒	7200		负数不启用
http_proxy	string	连接服务端使用的代理地址			格式为 (protocol)://user:passwd@192.168.1.128:8080 protocol 目前支持 http、socks5、ntlm
log_file	string	日志文件地址	./frpc.log		如果设置为 console，会将日志打印在标准输出中
log_level	string	日志等级	info	trace, debug, info, warn, error	
log_max_days	int	日志文件保留天数	3		
disable_log_color	bool	禁用标准输出中的日志颜色	false		

# 点对点转发

下载有如下的场景



内部网络无法直接访问该端口，可以通过FRP进行端口转发到公网

## 1、服务端

配置如下

```
[common]
server_addr = 118.178.134.226
server_port = 7000
[3389]
type = tcp
local_ip = 127.0.0.1
local_port = 3389
remote_port = 6000
```

启动服务端

```
C:\Users\Administrator\Desktop\frp_0.51.2_windows_amd64>frpc.exe -c frpc.ini
2023/08/17 15:32:37 [I] [root.go:220] start frpc service for config file [frpc.i
ni]
2023/08/17 15:32:37 [I] [service.go:301] [c9551fe206d01aad] login to server succ
ess, get run id [c9551fe206d01aad]
2023/08/17 15:32:37 [I] [proxy_manager.go:150] [c9551fe206d01aad] proxy added: [
3389]
2023/08/17 15:32:37 [I] [control.go:172] [c9551fe206d01aad] [3389] start proxy s
uccess
```

## 2、客户端

配置如下

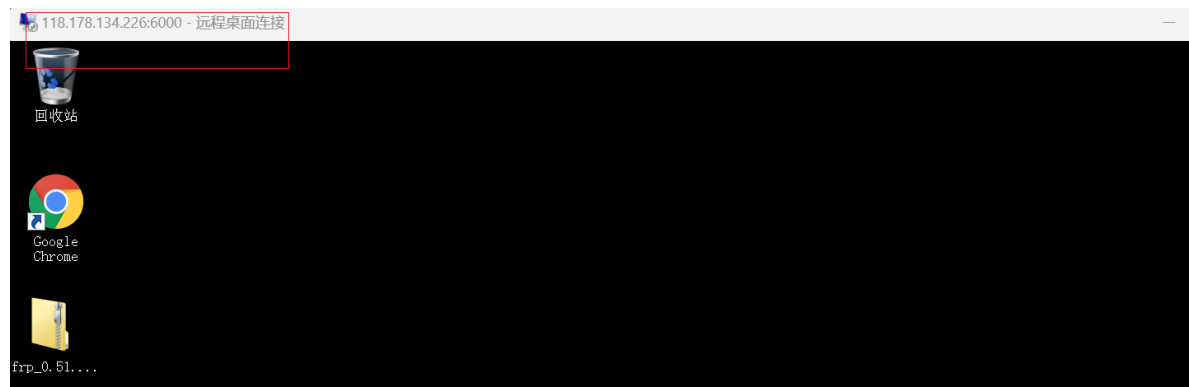
```
frps.ini - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
[common]bind_port = 7000
```

启动客户端

```
2023/08/17 15:32:31 [I] [root.go:204] frps uses config file: frps.ini
2023/08/17 15:32:31 [I] [service.go:206] frps tcp listen on 0.0.0.0:7000
2023/08/17 15:32:31 [I] [root.go:213] frps started successfully
2023/08/17 15:32:38 [I] [service.go:539] [c9551fe206d01aad] client login info
: ip [175.9.140.2:27302] version [0.51.2] hostname [] os [windows] arch [amd64]
2023/08/17 15:32:38 [I] [tcp.go:81] [c9551fe206d01aad] [3389] tcp proxy listen port [6000]
2023/08/17 15:32:38 [I] [control.go:497] [c9551fe206d01aad] new proxy [3389] type [tcp] success
```

### 3、结果

配置如下



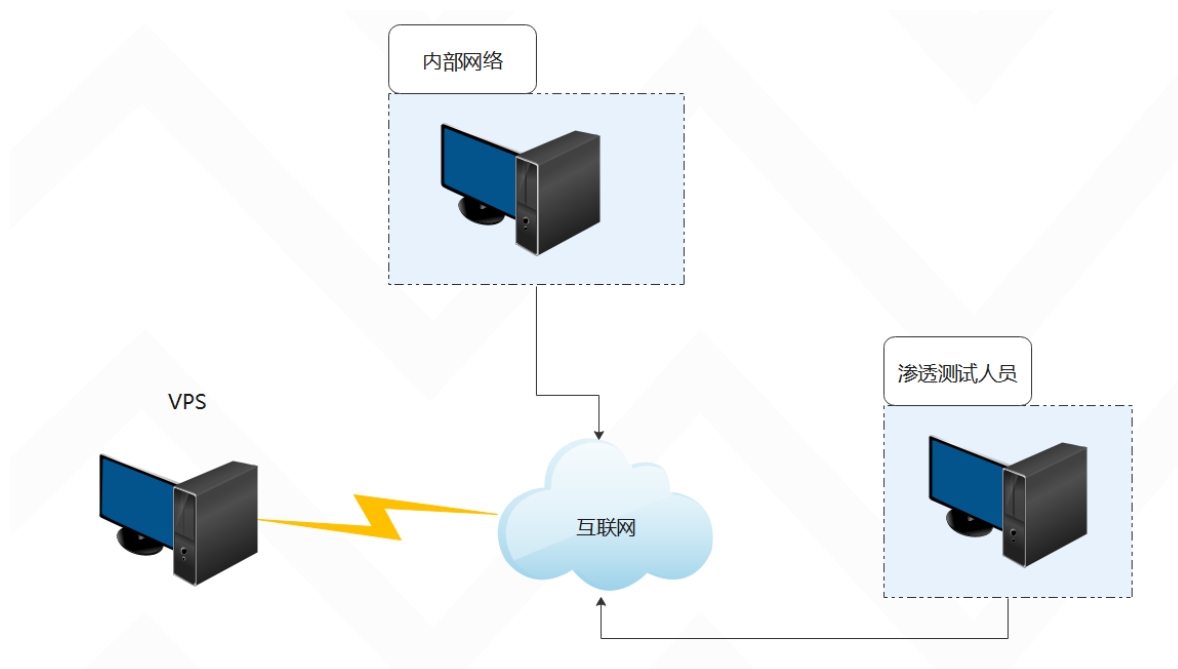
### 4、流量分析

因为我们配置的是TCP流量没有进行加密

4	2023-08-17 1...	118.178.134.226	192.168.41.250	TCP	39
5	2023-08-17 1...	192.168.41.250	118.178.134.226	TCP	8
6	2023-08-17 1...	118.178.134.226	192.168.41.250	TCP	5
7	2023-08-17 1...	192.168.41.250	118.178.134.226	TCP	8
8	2023-08-17 1...	118.178.134.226	192.168.41.250	TCP	5
9	2023-08-17 1...	192.168.41.250	118.178.134.226	TCP	11
	2023-08-17 1...	118.178.134.226	192.168.41.250	TCP	5
	2023-08-17 1...	192.168.41.250	118.178.134.226	TCP	8
	2023-08-17 1...	192.168.41.250	118.178.134.226	TCP	9
	2023-08-17 1...	118.178.134.226	192.168.41.250	TCP	5
	2023-08-17 1...	118.178.134.226	192.168.41.250	TCP	5
	2023-08-17 1...	118.178.134.226	192.168.41.250	TCP	8

## 搭建socks隧道

使用场景如下



现在我们搭建socks隧道实现对目标机器的全端口访问

客户端

配置文件

```
frps.ini - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
[common]bind_port = 7000
```

启动文件

```
C:\Users\Administrator\Desktop\frp_0.51.2_windows_amd64>frpc.exe -c frpc.ini
2023/08/17 15:56:19 [I] [root.go:220] start frpc service for config file [frpc.ini]
2023/08/17 15:56:19 [I] [service.go:301] [746c1582c809152b] login to server success, get run id [746c1582c809152b]
2023/08/17 15:56:19 [I] [proxy_manager.go:150] [746c1582c809152b] proxy added: [3389]
2023/08/17 15:56:19 [I] [control.go:172] [746c1582c809152b] [3389] start proxy success
```

服务端

配置文件

```
[common]
server_addr = 118.178.134.226
server_port = 7000
[3389]
type = tcp
plugin=socks5
remote_port = 6000
```

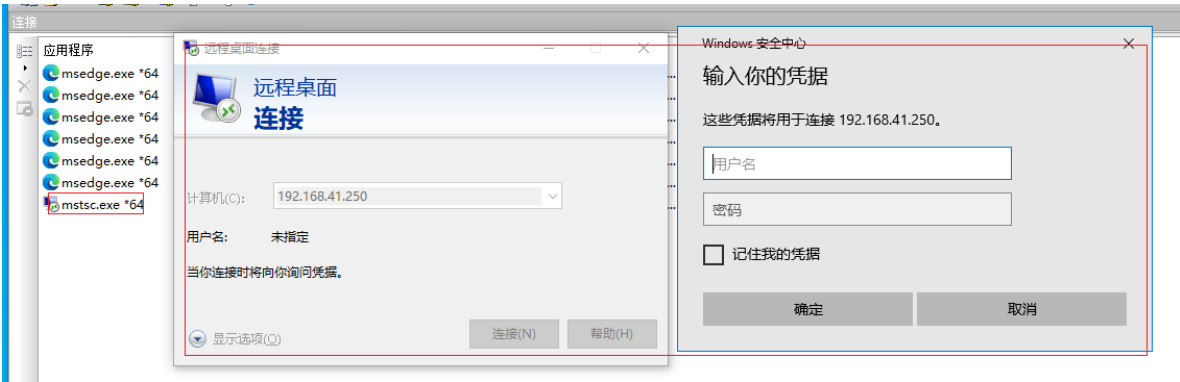
启动文件

```
[root@root FRP]# ./frps -c frps.ini
2023/08/17 15:55:54 [I] [root.go:204] frps uses config file: frps.ini
2023/08/17 15:55:54 [I] [service.go:206] frps tcp listen on 0.0.0.0:7000
2023/08/17 15:55:54 [I] [root.go:213] frps started successfully
2023/08/17 15:55:58 [I] [service.go:539] [1e34e5e98ff6bda0] client login info: ip [175.9.140.2:28275] version [0.5
1.2] hostname [] os [windows] arch [amd64]
2023/08/17 15:55:58 [I] [tcp.go:81] [1e34e5e98ff6bda0] [3389] tcp proxy listen port [6000]
2023/08/17 15:55:58 [I] [control.go:497] [1e34e5e98ff6bda0] new proxy [3389] type [tcp] success
2023/08/17 15:56:18 [I] [control.go:332] [1e34e5e98ff6bda0] control writer is closing
2023/08/17 15:56:18 [I] [proxy.go:110] [1e34e5e98ff6bda0] [3389] proxy closing
2023/08/17 15:56:18 [W] [proxy.go:196] [1e34e5e98ff6bda0] [3389] listener is closed: accept tcp [::]:6000: use of
closed network connection
```

配置代理

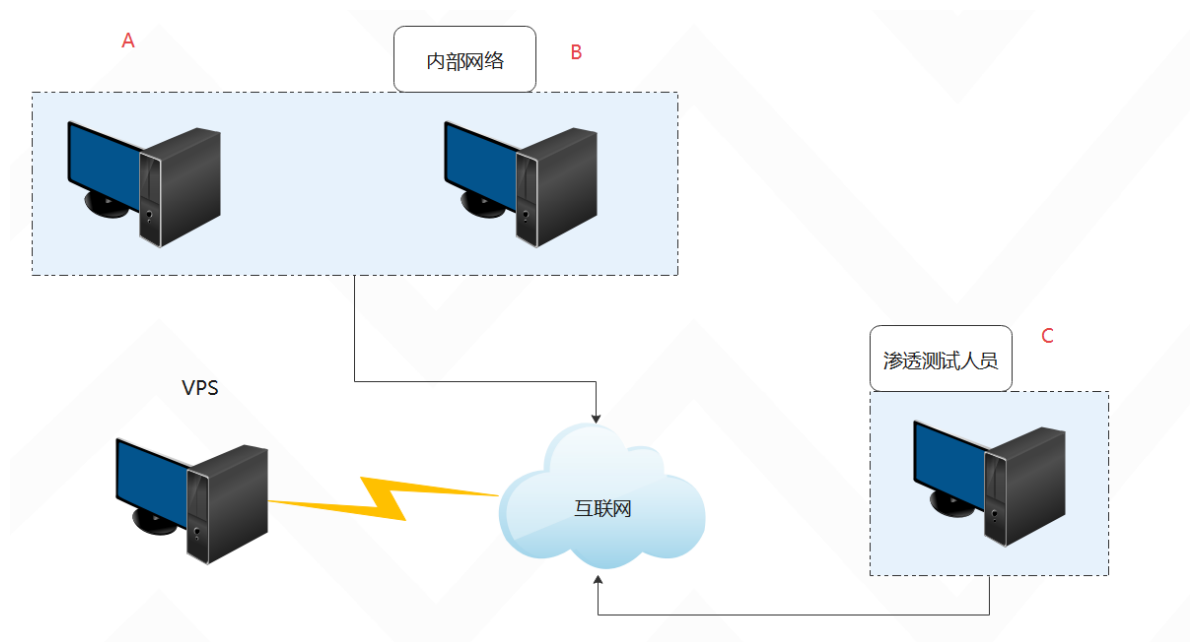


测试连接



## 搭建socks多级隧道

多级socks隧道搭建网络拓扑如下



首先搭建A->B的隧道

A执行如下

```
[common]
server_addr = 192.168.41.142
server_port = 7001
[sock5]
type = tcp
plugin = socks5
remote_port = 6000
```

B执行如下

```
[common]

bind_port = 7001
```

搭建好隧道之后接着使用端口转发

B上执行如下

```
[common]
server_addr = 118.178.134.226
server_port = 7000
[portforward]
type = tcp
local_ip = 127.0.0.1
remote_port = 6000
local_port = 6000
```



C上执行如下



```
frps.ini - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

[common]bind_port = 7000
```

结果

