golang端口转发工具

坏虾 / 2017-10-25 07:39:12 / 浏览数 3997 安全工具 工具 顶(0) 踩(0)

初学golang，写了一个端口转发工具。其实就是lcx中的tran功能。

使用场景 A ===》 B A能访问B
B ===》C B能访问C

但是 A C 不通。

这样就可以在B上执行lcx_go.exe 0.0.0.0:1987 B的IP:3389

这时候只要 mstsc A的IP:1987 就可以连上B的RDP了。

目前只写了这一个小功能。

主要是学习一下go的语法。

Go的性能接近C++，但是大小是一个缺陷。所以最后我用UPX压缩了一下。还是好大。

代码如下。之后再把剩下的slave和listen功能补充上。

[code]package main

```
import (
	"fmt"
	"io"
	"net"
	"sync"
	"os"
)

var lock sync.Mutex
var trueList []string
var ip string

var list string

func main() {
	if len(os.Args) != 3 {
		fmt.Fprintf(os.Stderr, "Usage: %s 0.0.0.0:8888 ip:3389\n", os.Args[0])
		os.Exit(1)
	}
	ip = os.Args[1]
	rip := os.Args[2]
	server(rip)
}

func server(rip string) {
	fmt.Printf("Listening %s\n", ip)
	lis, err := net.Listen("tcp", ip)
	if err != nil {
		fmt.Println(err)
		return
	}
	defer lis.Close()

	 for {
		conn, err := lis.Accept()
		if err != nil {
			fmt.Println("建立连接错误:%v\n", err)
			continue
		}

		 fmt.Println("Connecting from " ,conn.RemoteAddr())
		go handle(conn, rip)
	}
}
```

```
func handle(sconn net.Conn,rip string) {
	defer sconn.Close()
	dconn, err := net.Dial("tcp", rip)
	if err != nil {
		fmt.Printf("连接%v失败:%v\n", ip, err)
		return
	}
	ExitChan := make(chan bool, 1)
	go func(sconn net.Conn, dconn net.Conn, Exit chan bool) {
		io.Copy(dconn, sconn)
		ExitChan <- true
	}(sconn, dconn, ExitChan)

	 go func(sconn net.Conn, dconn net.Conn, Exit chan bool) {
		io.Copy(sconn, dconn)
		ExitChan <- true
	}(sconn, dconn, ExitChan)
	<-ExitChan
	dconn.Close()
}[/code]
```

点击收藏 | 0 关注 | 0

上一篇：如何利用反弹 shell 构建你的... 下一篇：第三季度安全客季刊发布

1. 2 条回复


hades 2017-10-25 07:42:54

越来越会玩了~

0 回复Ta


坏虾 2017-10-25 07:49:31

活到老学到老嘛，基友。。。

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录