网络安全协议及分析
代理安全Socks

密码与网络安全系　刘虹

2025年春季学期

课程体系

第一章 概述

第二章 链路层扩展L2TP

第三章 IP层安全IPSec

第四章 传输层安全SSL和TLS

第五章 会话安全SSH

第六章 代理安全Socks

第七章 网管安全SNMPv3

第八章 认证协议Kerberos

第九章 应用安全

本章学习目标

- 代理的定义
- Socks框架
- Socks4和Socks5

提纲

代理

- 代理服务器是用户接入互联网的重要手段之一，使用代理服务器接入方式，通信双方的数据流都由代理服务器转发。
- 代理服务器：
  - 应用层代理：位于TCP/IP协议栈的应用层，例如HTTP代理和FTP代理。
  - 互联网连接共享（Internet Connection Sharing，ICS）：位于IP层，实现多个私用IP地址共享同一公共地址。
  - Socks代理（防火墙安全会话转换协议，Protocol for sessions traversal across firewall securely)

代理

应用层代理

1. 通常具备缓存功能，当它发现客户端所请求的资源已经被缓存时，将直接返回这些资源，由此提高通信效率并减轻应用服务器的负担；
2. 经过代理服务器转发的报文首部将服务器本身的IP地址作为请求的源地址，可以用于突破基于IP地址的访问限制；
3. 应用层网关使用的是应用层待机技术，它是目前常用的一种防火墙技术，能够检查进出的数据包，通过网关复制传递数据，防止在受信器户与不受信的服务器与主机间直接建立联系。

代理

互联网连接共享ICS

- 对家庭网络或小型网络所提供的一种连接共享服务。
- 这种代理依托网络地址转换（Network Address Translation，NAT）技术，可以实现多个私有地址共享同一公共地址。

代理

- Socks：防火墙安全会话转换协议
  - 使用端口1080，工作于应用层。
  - 可转发所有高层应用，对操作系统无限制。
  - 专门设计用于防火墙，在应用层和传输层之间垫了一层，为在 TCP和UDP域中的客户机/服务器应用程序能更方便安全地使用网络防火墙所提供的服务。
    - Socks库和Sockd守护程序
    - 不支持ICMP转发

提纲

一、代理的定义

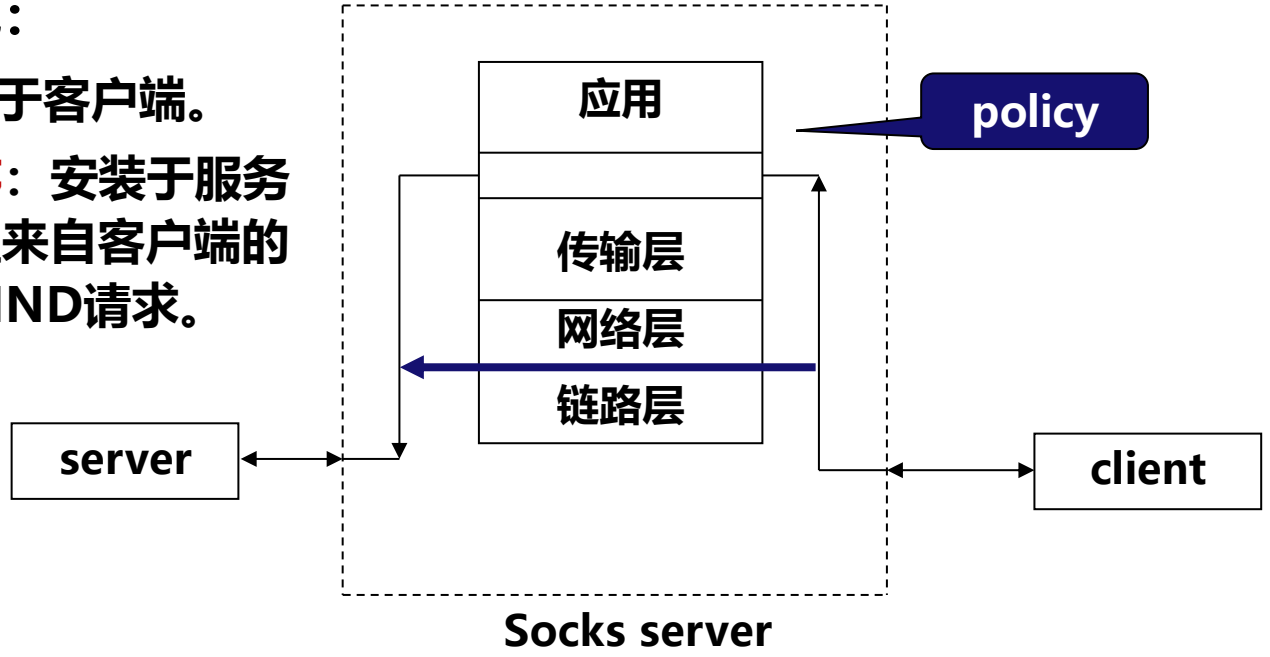
二、Socks框架

三、Socks4

四、Socks5

二 Socks框架

- Socks使用C/S模型
- Socks软件包组成：
  - Socks库：安装于客户端。
  - Sockd守护程序：安装于服务器，接收并处理来自客户端的CONNECT和BIND请求。



二 Socks框架

- Socks的优点：
  - 任何主机都可作为Socks代理服务器。
  - Socks本身未定义机密性、完整性保护方法，但由于所有通信量都要经过代理服务器转发，为统一制定安全策略并部署安全防护措施提供了便利。
  - Socks 5支持多种客户端身份认证方案，如果某些认证方案能够支持机密性和完整性保护，这些功能在应用Socks后仍然能够得到保留。

二 Socks框架

- Socks库函数和Socket库函数对应表

Socks函数	功能	Socket函数
Rconnect	与服务器建立连接	Connect
Rbind	将套接字与IP地址和端口绑定	Bind
Rlisten	在套接字上监听连接	Listen
Rgetsockname	获取套接字详细信息	Getsockname
Raccept	接受连接请求	Accept

二 Socks框架

- **Socks客户端命令：**
  - **CONNECT：通告代理服务器与远程主机建立连接，调用函数：Rconnect**
  - **BIND：通告服务器接收来自某个远程主机的连接请求，调用函数：Rbind，Rlisten和Raccept.**

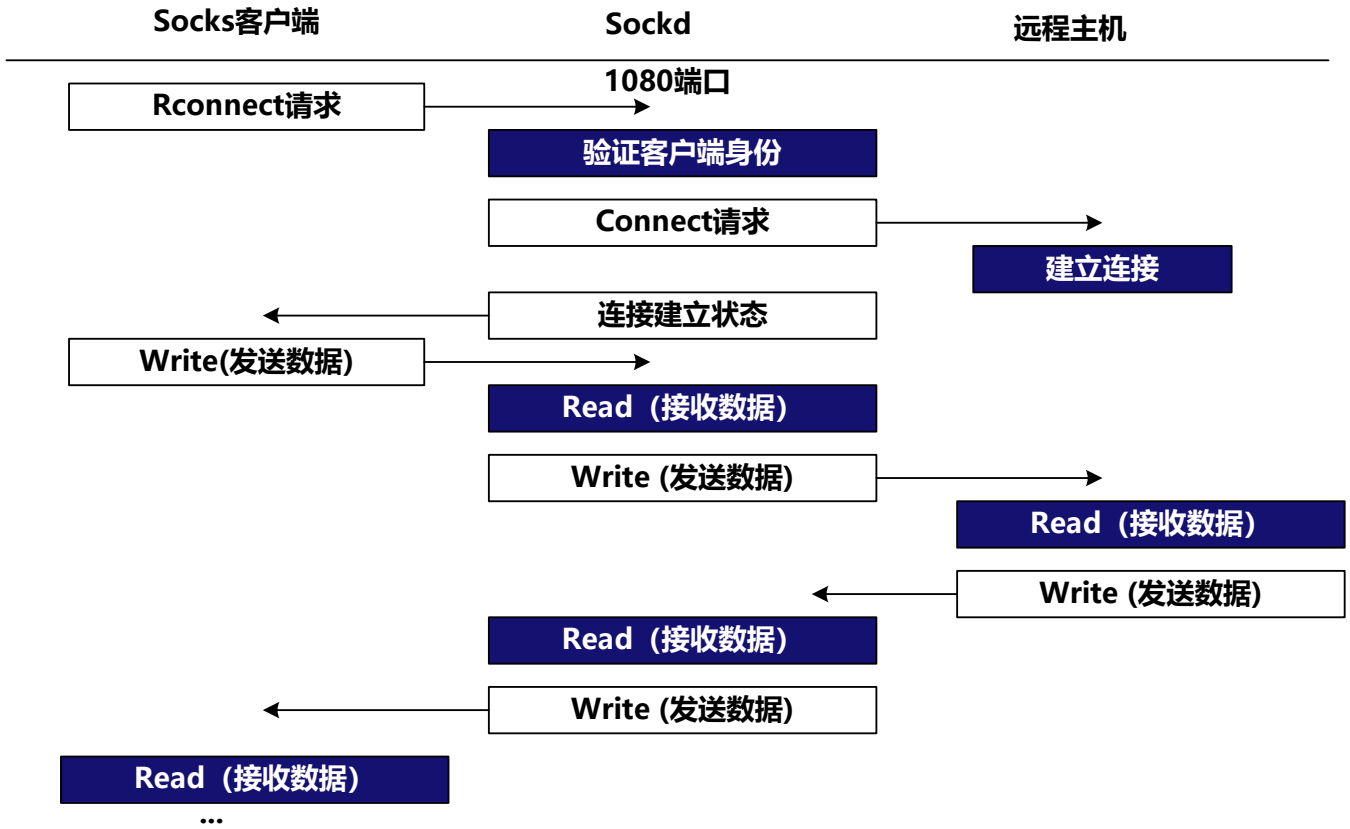
二 CONNECT 命令处理过程

- **不使用Socks时客户端和远程主机的连接建立过程：**
  - 客户端调用Socket的Connect函数
  - 后台与远程主机完成TCP的三次握手过程：
    - 如果连接成功，Connect函数返回TRUE
    - 如果连接失败，Connect函数返回FALSE
  - 如连接成功，则客户端调用Write和Read函数发送和接受数据

二 CONNECT 命令处理过程

- **使用Socks时客户端和远程主机建立连接过程**
  - 客户端调用Rconnet函数，指明了远程主机的IP地址和端口。
    - 与Socks服务器的1080号端口建立连接。
    - 发送CONNECT请求消息
  - 服务器的Sockd验证客户端的身份，如验证通过后使用Connect与远程主机建立连接。
  - 服务器Sockd向客户端返回连接状态
  - 若连接成功，服务器作为数据的中转站。

二 CONNECT 命令处理过程



二 BIND命令处理过程

- **不使用Socks代理时，客户端接收来自远程主机的连接请求的过程：**
  - 创建套接字
  - 调用Bind函数，将套接字与本地端口号绑定
  - 调用Listen函数，监听连接请求
  - 调用Accept函数，接受连接请求，启动新进程（或线程）处理连接请求
  - 调用Write和Read函数发送和接收数据

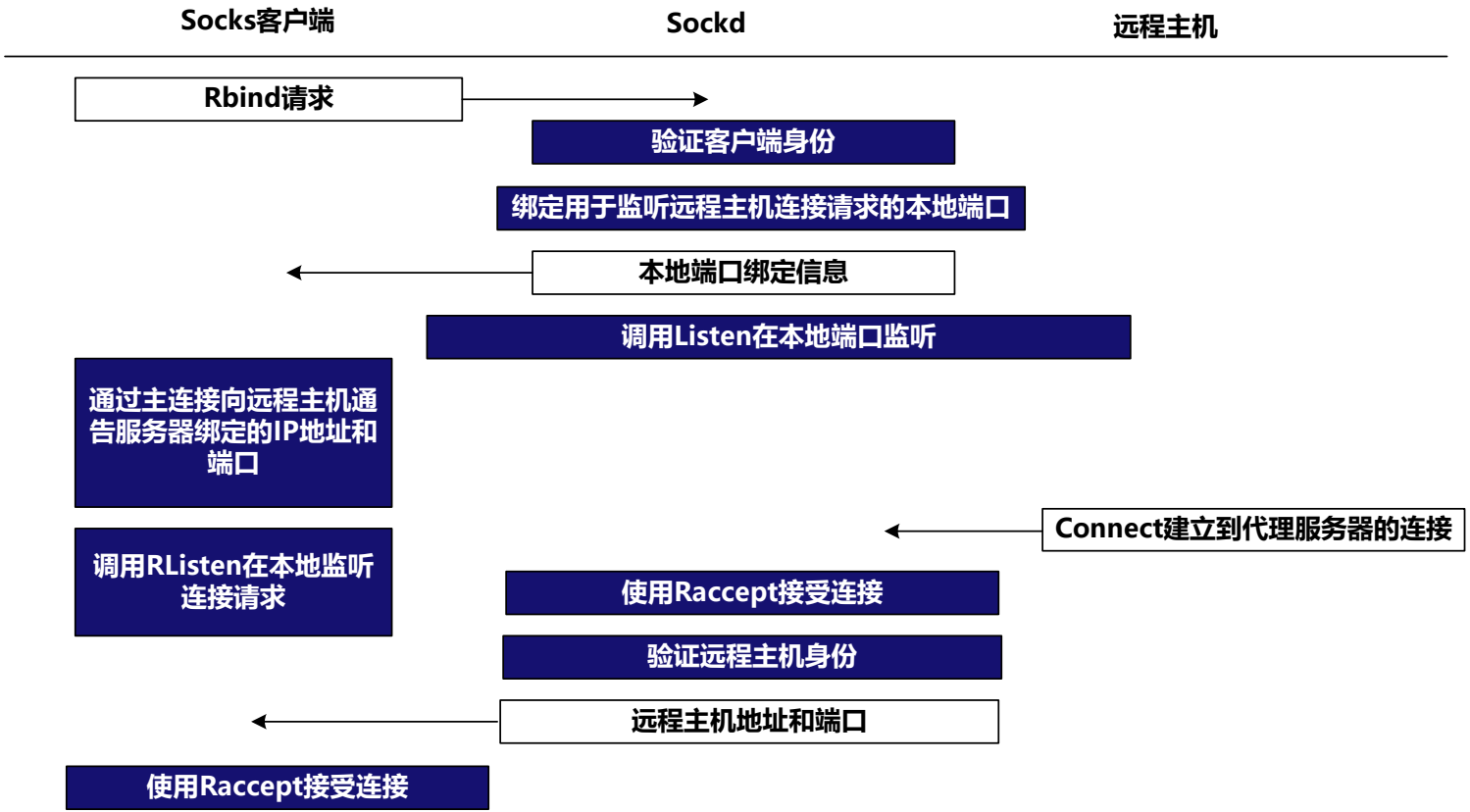
二 BIND命令处理过程

- 使用Socks代理时，客户端通告代理服务器接收远程主机的连接请求过程：
  - 利用CONNECT命令建立了到远程主机的“主连接”
  - Socks客户端调用Rbind函数向代理服务器发送请求：
  - 通过三次握手与代理服务器建立连接
  - 向代理服务器发送BIND请求消息，其中包含了远程主机IP地址

二 BIND命令处理过程

- **代理服务器验证客户端身份，若验证通过**
  - 代理服务器在本地创建套接字，绑定一个【IP地址和端口】，用以接收远程主机的连接请求，把IP地址和端口的绑定信息返回给客户端，并在该套接字上监听远程主机的连接请求。
  - 客户端通过主连接向远程主机通告代理服务器绑定的IP地址和端口，然后调用RListen在本地监听连接请求。
  - 远程主机向客户端通告的IP地址和端口号发出连接请求。
  - 代理服务器将远程主机发来的信息中的源地址与客户端BIND请求中包含的地址进行比较，如果一致，则接受请求并通告客户端；否则拒绝连接，并向客户端返回错误应答。
  - 客户端收到代理服务器转发的连接请求后，调用Raccept接受连接，开始数据投递过程。

二 BIND命令处理过程



提纲

一、代理的定义

二、Socks框架

三、Socks4

四、Socks5

Socks 4

- Socks4定义CONNECT请求消息、BIND请求消息以及状态应答消息
  - 请求消息格式

VN	CD	目标端口	目标IP	用户IP	NULL

  - 应答消息格式

VN	CD	目标端口	目标IP

四 CONNECT请求及状态应答消息

- 在收到这个请求后，Socks4服务器会根据源IP地址、源端口、目标IP地址、目标端口号及USERID信息确定是否与远程主机建立连接。
  - 若同意建立该连接，则向目标发出连接请求。
  - 无论结果如何，Socks服务器都会返回状态应答消息，其中VN应取值“0”；CD字段则包含了结果状态码；“目标端口"和“目标IP”这两个字段被忽略。

四 CONNECT请求及状态应答消息

- Socks4支持基于身份协议（RFC1413）的客户端认证方法，即Ident。该协议使用113号TCP端口，守护程序实现通常命名为identd。
  - 服务器收到这个回应后，会将其与CONNECT请求中包含的用户ID进行比较。若比较相同，则身份验证通过；否则验证失败。

四 BIND请求及状态应答消息

- 服务器收到请求后会返回**状态应答消息**。
  - 当目标IP取0时，表示的就是socks服务器自身的地址。
  - 客户端在收到这个应答后，应将这两个信息通过主连接通告给远程主机。
- 当服务器收到远程主机的连接请求时，会匹配请求报文的**源IP地址**与BIND请求中包含的**目标IP**字段。
  - 若二者不同，服务器向客户端返回状态码为91的应答，并断开与远程主机的连接；
  - 否则返回状态码为90的应答，并准备进行随后的数据转发过程。

提纲

一、代理的定义

二、Socks框架

三、Socks4

四、Socks5

Socks 5

- **Socks 5沿袭了Socks 4的体系结构以及命令，并作了以下扩展**
  - **扩展了客户端身份认证功能，支持多种身份验证方法：用户名/口令认证方法、GSSAPI认证。**
  - **扩展了寻址方法：除了IPv4地址外，还支持域名及IPv6地址**
  - **增加了对UDP的支持**

四 身份认证扩展

- 服务器从客户端提供的认证方法列表中选定一个方法，并返回给客户端。
- 这个应答消息包含1B的“版本”字段以及1B的“认证方法"字段。
  - 若“认证方法"字段为255，说明认证方法协商失败，通信双方将立刻终止连接。

四 用户名/口令认证方法

- 如果客户端和服务器协定使用**用户名/口令**认证方法后，
  - 客户端向服务器发送认证请求报文，报文中包括了用户名和口令。
  - 服务器验证用户名和口令，并根据验证结果返回应答消息。

四 请求/应答过程及寻址方法扩展

- 请求消息

VN	命令	保留	地址类型	目标地址…	目标端口

- 应答消息

VN	应答	保留	地址类型	绑定地址…	绑定端口

CMD: 请求消息类型，取值如下

X’ 01: connection

X’ 02: bind

X’ 03: UDP associate

ATYP: 地址类型，取值如下

X’ 01: IP v4

X’ 03: Domain name

X’ 04: IP v6

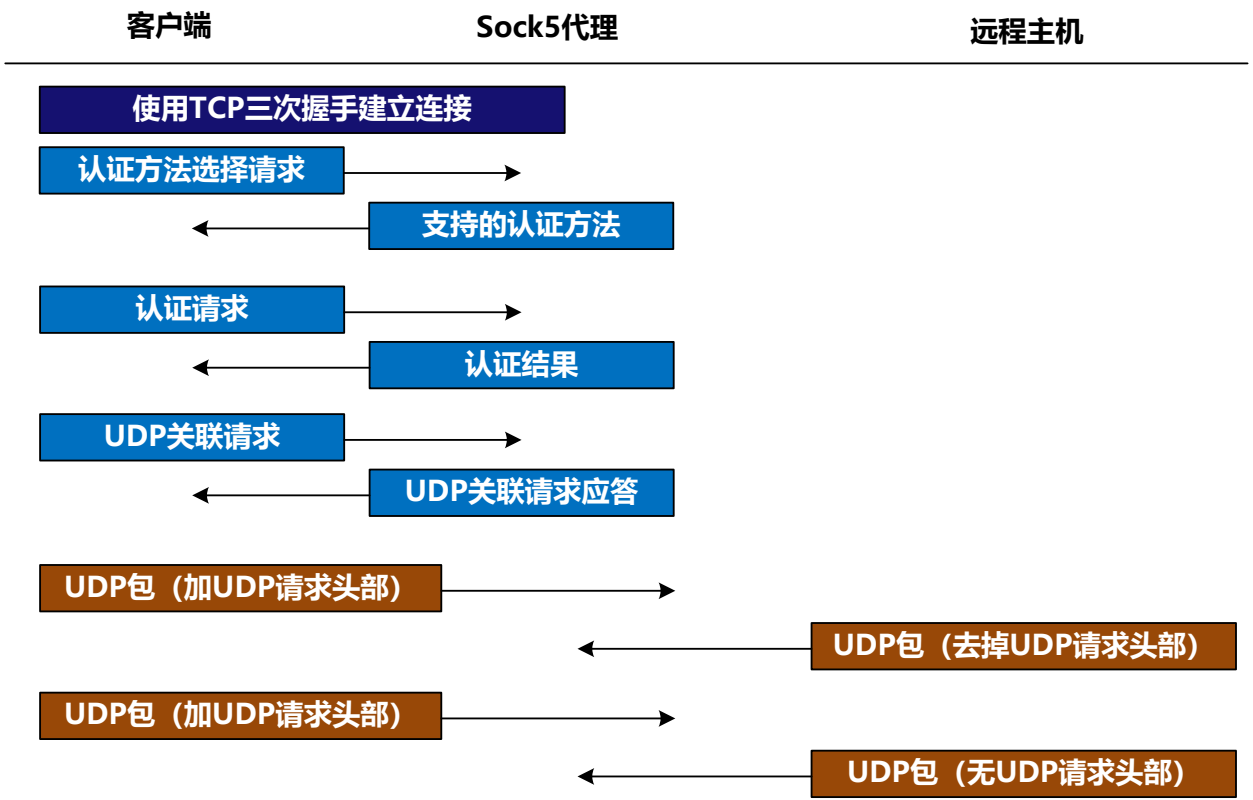
四 UDP支持

- 使用Socks 5转发UDP会话方法：
  - 请求命令为“UDP associate”（设置为字段3），“目标地址”和“目标端口”设置为客户端发送及接收UDP报文所使用的地址和端口号。
  - 服务器返回的应答消息中将“绑定地址”和“绑定端口”设置为用于转发UDP会话的地址和端口号。
- 在UDP数据报报头及应用数据之间增加一个UDP请求首部：

保留()	分片	地址类型	目标地址…	目标端口	数据…

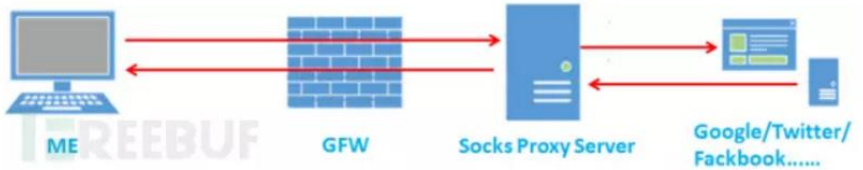
四 UDP支持

- UDP转发流程
  - socks客户端和服务器同时充当UDP转发应用的客户端和转发代理。



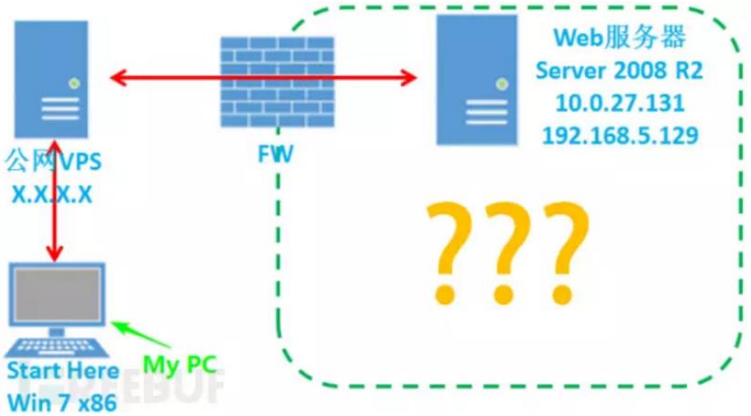
四 Socks代理：内网漫游

- 早期我们可能会采用HTTP Proxy代理，在浏览器上设置下代理服务器的IP、端口、认证账户和密码。但有些软件的网络通信数据并不是HTTP协议，就需要寻找其它方法。
- SOCKS代理是个不错的解决方案，不管应用层是什么协议，只要是传输层是TCP协议就可以代理。



四 Socks代理：内网漫游

- 左侧是个人电脑和一台具有公网IP的VPS，My PC通过NAT连接互联网。
- 右侧模拟的是一个小型网络。假设我们获得了一台Web服务器的控制权限，该服务器配有两块网卡，10.0.27.131连通互联网，192.168.5.129可与内部网络连通。



四 相关工具

- Earthworm
  - 工具网址：http://rootkiter.com/EarthWorm
  - Earthworm不仅具备SOCKSv5的功能，还具备反弹式SOCKS代理和lcx端口转发、映射的全部功能（listen、tran、slave）。
  - 同时提供了多种可执行文件，以适用不同的操作系统，Linux、Windows、MacOS、Arm-Linux。命令行操作，无GUI，适合渗透测试使用。

四 相关工具

- xsocks
  - 工具网址：https://github.com/5loyd/xsocks
  - xsocks是一款能在Windows和Linux系统上运行的反弹式SOCKS代理服务端。命令行操作，无GUI，适合渗透测试使用。当然使用前需要从github上下载代码编译，Windows用VS2010及以上版本即可编译。

四 相关工具

- ShadowSOCKS（影梭）
  - ShadowSOCKS是一个开源 SOCKS5 代理项目。
  - ShadowSOCKS有python、nodejs等诸多版本，还有移动版。
  - 在Windows平台下，推荐使用libQtShadowSOCKS，用C++编写并使用了Qt 5框架。
  - 无GUI界面，命令行版，支持x86和x64系统，相当好用。

四 相关工具

- SocksCap64
  - 工具网址：http://www.sockscap64.com
  - 一般浏览器是支持代理设置的，但如果是渗透测试，肯定要使用很多不同的工具，但这些工具并不能保证都支持代理设置。因此我们需要使用应用程序外壳代理软件来使不能设置代理的软件和程序的网络数据流量通过代理。
    - 完美支持SOCKS4/5/Http/Shadowsocks代理协议。
    - 完美支持TCP& UDP网络协议。
    - 支持远程SOCKS代理解析域名。

办公地点：理科大楼B1209

联系方式：17621203829

邮箱：liuhongler@foxmail.com