<https://www.eet-china.com/mp/a35364.html>

<https://blog.csdn.net/hzhsan/article/details/43445787>

|  |  |
| --- | --- |
| **参数** | **含义** |
| -a | 将网络地址和广播地址转变成名字 |
| -c | 在收到指定的包的数目后，tcpdump就会停止； |
| -d | 将匹配信息包的代码以人们能够理解的汇编格式给出；以可阅读的格式输出。 |
| -dd | 将匹配信息包的代码以c语言程序段的格式给出； |
| -ddd | 将匹配信息包的代码以十进制的形式给出； |
| -e | 在输出行打印出数据链路层的头部信息； |
| -f | 将外部的Internet地址以数字的形式打印出来； |
| -l | 使标准输出变为缓冲行形式； |
| -n | 直接显示IP地址，不显示名称； |
| -nn | 端口名称显示为数字形式，不显示名称； |
| -t | 在输出的每一行不打印时间戳； |
| -v | 输出一个稍微详细的信息，例如在ip包中可以包括ttl和服务类型的信息； |
| -vv | 输出详细的报文信息； |
| -F | 从指定的文件中读取表达式,忽略其它的表达式； |
| **-i** | **指定监听的网络接口；** |
| -r | 从指定的文件中读取包(这些包一般通过-w选项产生)； |
| -w | 直接将包写入文件中，并不分析和打印出来； |
| -T | 将监听到的包直接解释为指定的类型的报文，常见的类型有rpc （远程过程调用）和snmp（简单 网络管理协议；） |

条件过滤：

1. 只抓某种协议的包

tcpdump tcp

tcpdump udp

1. 指定协议端口号

使用port参数，用于指定端口号。

命令：tcpdump tcp port 80

使用portrange参数，用于指定端口范围。

命令：tcpdump tcp portrange 1-1024

1. 指定源与目标

src 表示源。

dst 表示目标。

命令：

tcpdump src port 8080

tcpdump dst port 80

1. 指定特定主机的消息包

tcpdump host 192.168.1.113

1. 指定数据包大小

tcpdump greater 1000

tcpdump less 10

**tcpdump 与wireshark**

Wireshark(以前是ethereal)是Windows下非常简单易用的抓包工具，现在也有Linux版本。

通过Tcpdump抓取的数据包分析比较麻烦，要想很方便的分析数据包， 我们可以用Tcpdump + Wireshark 的完美组合实现：在 Linux 里抓包，然后在Windows 里分析包。

保存数据包为wireshark能识别的文件：

tcpdump tcp -i eth1 -t -s 0 -c 100 and dst port ! 22 and src net 192.168.1.0/24 -w ./target.cap

|  |  |
| --- | --- |
| **参数** | **含义** |
| tcp | ip icmp arp rarp 和 tcp、udp、icmp这些选项等都要放到第一个参数的位置，用来过滤数据报的类型 |
| -i eth1 | 只抓经过接口eth1的包 |
| -t | 不显示时间戳 |
| -s 0 | 抓取数据包时默认抓取长度为68字节。加上-S 0 后可以抓到完整的数据包 |
| -c 100 | 只抓取100个数据包 |
| dst port ! 22 | 不抓取目标端口是22的数据包 |
| src net 192.168.1.0/24 | 数据包的源网络地址为192.168.1.0/24 |
| -w ./target.cap | 保存成cap文件，方便用ethereal(即wireshark)分析 |

tcpdump -i any host 41.50.5.28 -s 0 -w /tmp/28stream.cap -vvn #抓包并保存

tcpdump的选项介绍

　　　-a 　　　将网络地址和广播地址转变成名字；

　　　-d 　　　将匹配信息包的代码以人们能够理解的汇编格式给出；

　　　-dd 　　　将匹配信息包的代码以c语言程序段的格式给出；

　　　-ddd 　　　将匹配信息包的代码以十进制的形式给出；

　　　-e 　　　在输出行打印出数据链路层的头部信息，包括源mac和目的mac，以及网络层的协议；

　　　-f 　　　将外部的Internet地址以数字的形式打印出来；

　　　-l 　　　使标准输出变为缓冲行形式；

　　　-n 　　　指定将每个监听到数据包中的域名转换成IP地址后显示，不把网络地址转换成名字；

-nn： 指定将每个监听到的数据包中的域名转换成IP、端口从应用名称转换成端口号后显示

　　　-t 　　　在输出的每一行不打印时间戳；

　　　-v 　　　输出一个稍微详细的信息，例如在ip包中可以包括ttl和服务类型的信息；

　　　-vv 　　　输出详细的报文信息；

　　　-c 　　　在收到指定的包的数目后，tcpdump就会停止；

　　　-F 　　　从指定的文件中读取表达式,忽略其它的表达式；

　　　-i 　　　指定监听的网络接口；i是interface的意思，我们需要告诉tcpdump去抓哪个网卡

-p： 将网卡设置为非混杂模式，不能与host或broadcast一起使用

　　　-r 　　　从指定的文件中读取包(这些包一般通过-w选项产生)；

　　　-w 　　　直接将包写入文件中，并不分析和打印出来；

-s snaplen snaplen表示从一个包中截取的字节数。0表示包不截断，抓完整的数据包。默认的话 tcpdump 只显示部分数据包,默认68字节。

-T 　　　将监听到的包直接解释为指定的类型的报文，常见的类型有rpc （远程过程调用）和snmp（简单网络管理协议；）

-X 告诉tcpdump命令，需要把协议头和包内容都原原本本的显示出来（tcpdump会以16进制和ASCII的形式显示），这在进行协议分析时是绝对的利器。

==========================================================================================================================

tcpdump -i any host wenyimysql.mysql.database.azure.com -s 0 -w ./stream.cap -vvn

root@hadooplel:/home/azureuser# tcpdump -i any host wenyimysql.mysql.database.azure.com -s e -w 
tcpdump: listening on any, link-type LIWX SLL (Linux cooked VI), capture size 262144 bytes 
AC116 packets captured 
13ø packets received by filter 
e packets dropped by kernel 
./stream.cap -vvn 

放到wireshark中。

A screenshot of a computer

AI-generated content may be incorrect.