Linux网络安全运维

# 网络实时流量监控工具iftop

Iftop是一个免费的网卡实时流量监控工具 。iftop可以监控指定网卡的实时流量、端口连接信息、反向解析ip等，还可以精确显示本机网络流量的情况以及网络内个主机与本机相互通讯流量集合。通过iftop输出可以迅速定位主机流量异常。

## Iftop的安装

yum install libpcap libpcap-devel ncurses ncurses-devel

yum install flex byacc

tar -zxvf iftop-0.17.tar.gz

cd iftop-0.17/

./configure

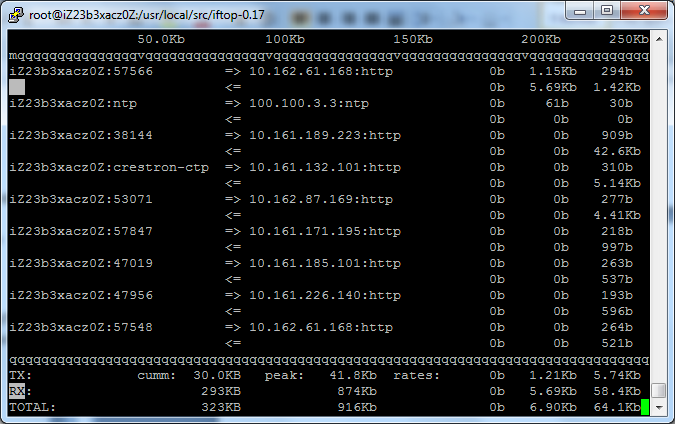
make && make install

## 使用iftop监控网卡实时流量

### Iftop使用参数：

* -i：指定需要监控的网卡
* -n：将输出的主机信息都通过IP显示，不进行DNS解析
* -B：输出以byte为单位显示网卡流量，默认是bit。
* -p：已混杂模式运行iftop，此时iftop可以用作网络探测器
* -N：只显示连接端口号，不显示端口对应服务名称。
* -P：显示主机以及端口信息。
* -F:显示特定网段的进出流量。（iftop –F 192.168.12.0/24）
* -m：设置iftop输出页面中最上面的流量刻度最大值，流量刻度分五个大段显示。

### Iftop数据界面说明



第一部分是iftop最上面一行，此行信息是流量刻度，用于显示网卡带宽流量

第二部分是分为左，中，右三列，左列和中列记录了哪些ip或者主机正在和本机网络进行连接。其中中列的=>代表发送数据，<=代表接受数据，通过箭头可以清楚的知道哪两个ip进行通讯。最右列分为三列，表示外部ip连接本机2s、10s、和40s内的平均流量。

第三部分在最下方，分为三行：TX表示发送数据，RX表示接收数据，TOTAL表示发送和接收全部流量。此三行对应三列，cum列表示从运行iftop到目前的发送、接收和总数据流量；peak表示发送，接收，以及总的流量峰值；rates表示过去2s,10s,40s内的平均流量。

### Iftop的交互操作h

n - toggle DNS host resolution

P - pause display

s - toggle show source host

h - toggle this help display

d - toggle show destination host

b - toggle bar graph display

t - cycle line display mode

B - cycle bar graph average

T - toggle cummulative line totals

j/k - scroll display

N - toggle service resolution

f - edit filter code

S - toggle show source port

l - set screen filter

D - toggle show destination port

L - lin/log scales

p - toggle port display

! - shell command

q - quit

Sorting:

1/2/3 - sort by 1st/2nd/3rd column

< - sort by source name

> - sort by dest name

o - freeze current order

# 网络流量监控与分析工具Ntop和Ntopng

Ntop是网络流量监控中的新贵，它是一种网络嗅探器，在监测网络数据传输、排除网络故障方面功能十分强大。它通过分析网络流量来判断网络上存在的各种问题，还可以监控是否有黑客正在攻击网络，如果网络突然变缓慢，通过Ntop截获的数据包，可以确定是什么类型的数据包占据了大量带宽，以及数据包的发送时间、数据包传送的延时、数据包的来源地址等，通过这些信息，运维人员可以及时做出响应，或者对网络进行调整，从而保证网络正常、稳定运行。Ntop提供了命令行界面和Web界面两种工作方式，通过Web界面，可以清晰展示网络的整体使用情况、网络中各主机的流量状态与排名、各主机占用的带宽以及各时段的流量明细、局域网内各主机的路由、端口使用情况等。

在Ntop版本更新到Ntop5.x之后，官方宣布停止Ntop版本的更新，继而推出替代版本Ntopng。Ntopng在Ntop版本的基础上，去掉了一些拖沓冗长的功能，同时新增了网络流量实时监控功能，并将各个功能进行重新梳理和整合，使整个流量展示更加智能化和合理化。Ntopng使用Redis键值服务按时间序列存储统计信息，通过这种方式实现了流量状态实时展示。与Ntop类似，Ntopng也内置Web服务功能，同时，也支持命令行界面和Web界面两种工作方式，但是Ntopng降低了对CPU和内存的使用率，资源消耗更少。

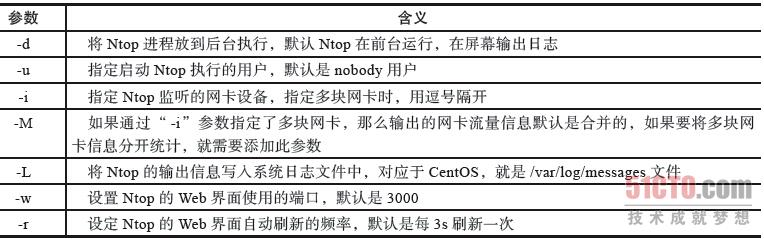
## Ntop的安装和使用

### 安装ntop

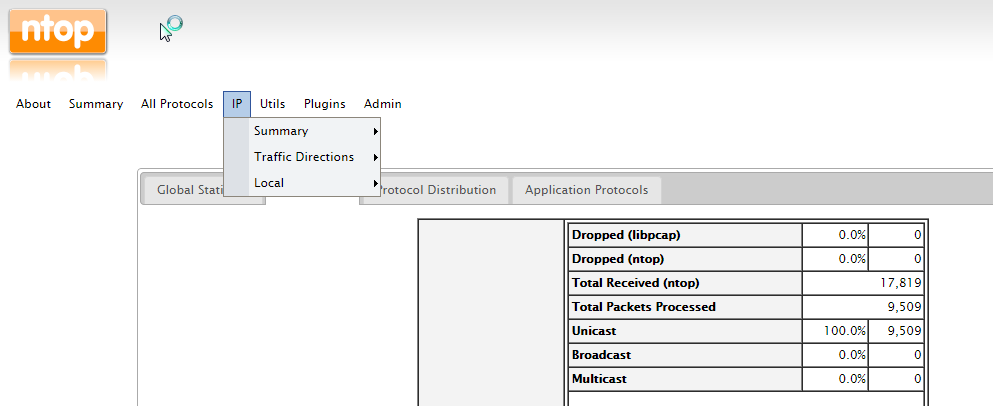
|  |
| --- |
| #安装依赖包  yum -y install libpcap libpcap-devel libtool libpng gdbm gdbm-devel glib libxml2-devel pango pango-devel gd zlib zlib-devel  yum -y install svn rrdtool rrdtool-devel python python-devel GeoIP GeoIP-devel  # 编译安装Ntop  tar -zxvf ntop-5.0.1.tar.gz  cd ntop-5.0.1/  ./autogen.sh --with-tcpwrap  make && make install  #配置Ntop，安装完后默认的路径在/usr/local/var/ntop  chown -R nobody /usr/local/var/ntop/  #输入两次密码，设置admin密码  ntop -A  #授权tcp\_wrappers  vim /etc/hosts.allow  ntop: 122.225.227.163  vim /etc/hosts.deny  ntop:ALL |

### Ntop命令的使用

ntop -i eth1 -L -d



### Web界面下Ntop的使用方法和技巧



## Ntopng安装和使用

### Ntopng安装

|  |
| --- |
| #安装依赖包  yum -y install libpcap-devel glib2-devel GeoIP-devel libxml2-devel libxml2-devel autoconf automake sqlite-devel libcurl libcurl-dev  #安装ntopng  tar -zxvf ntopng-2.0.tar.gz  cd ntopng-2.0/  ./autogen.sh  ./configure  /usr/bin/gmake geoip  make && make install  #安装redis  #配置Ntopng  -G=/var/tmp/ntopng.pid --local-networks "10.0.0.0/8"    //指定监控的网段 --interface eth0      //指定监控的接口 --interface eth1 --user nobody      //指定以什么账户运行 --http-port 3000    //指定Web服务的端口，不指定默认是3000端口  #启动redis服务和Ntopng  service redis start  /usr/local/bin/ntopng /etc/ntopng/ntopng.conf &  #登录  http://ip:3000  Username： admin  Password ： admin |

### Ntopng的使用

# 网络探测和安全审核工具nmap

## nmap和zenmap简介

nmap是一个开源的网络发现工具，通过它能够找出网络上的主机，并测试主机上哪些端口处于监听状态，接着通过端口确认主机运行的应用程序与版本信息，还能侦测操作系统的类型和版本。

Nmap是NetWork Mapper的缩写，是一个流行的安全工具。它的特点是：非常灵活，支持主流操作系统，使用简单，自由软件。Zenmap是nmap的GUI版本。，有nmap官方提供，随着nmap包一起发布。Zenmap使用Python语言编写，主要的目的是为nmap提供更加简单的操作。Nmap的下载地址：https://nmap.org/

Nmap功能非常强大，主要有如下四个基本功能：

* 主机发现
* 端口扫描
* 应用程序及版本监测
* 操作系统及版本监测

网络嗅探一般从主机发现开始，发现主机在线后，接着需要进行端口扫描来确定运行的应用和版本信息，最总确认操作系统的版本和漏洞信息。

## nmap的安装

tar jxvf nmap-6.49BETA5.tar.bz2

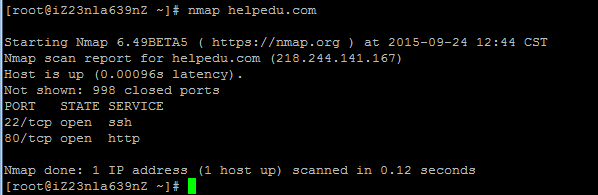
cd nmap-6.49BETA5/

./configure

make && make install

## nmap典型用法

### nmap 目标主机ip或者域名(可以多个)



### nmap –T4 –A –v目标主机ip或者域名

-A：启动全面扫描，包括主机发现，端口扫描，应用程序与版本、操作系统

-T(0~5)：指定扫描过程中使用的时序模板，等级越高，扫描速度越快(容易被防火墙或者入侵检测设备的发现)。

-v：显示扫描细节

主要包括：主机是否在线；扫描1000个最有可能开放的端口；对端口上运行的应用程序进行统计分析；对操作系统和版本进行分析；对目标主机的路由进行追踪。

## nmap主机发现扫描

主机发现主要判断目标主机是否在线，其扫描原理类似于ping，通过发送探测数据包目标主机，nmap支持多种不同的主机探测方法，例如TCP SYN/ACK包、发送SCTP包、发送ICMP echo、timestamp/netmask请求报文。

语法：nmap 选项或者参数 目标主机

常用选项：

-sn：只进行主机扫描，不进行端口扫描（常用）

-Pn：跳过主机扫描，默认所有主机都在线。进行端口扫描。（常用）

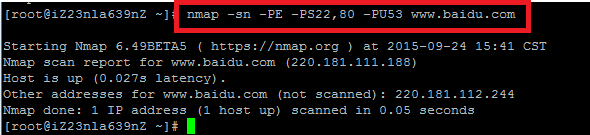
-sL：仅仅列出指定的目标主机ip,不进行主机发现扫描。

-PS/PA/PU/PY [portlist]：指定使用TCP SYN,TCP ACK,UDP,SCTP方式进行主机发现，例如nmap –PS80,21

-PE/PP/PM：使用ICMP echo、timestamp、netmask请求报文方式发现主机。

-PO：使用ip协议簇探测目标主机是否在线。

-n/-R：指定是否使用DNS解析协议。



## nmap端口扫描

端口扫描是nmap最核心的功能，通过端口扫描可以发现目标主机上TCP,UDP端口的开放情况。默认情况下，nmap会扫描1000个最有可能开放的端口，并将侦测到端口分为六类：

* Open：端口是开放的
* Closed：端口是关闭的
* Filtered：端口被防火墙过滤了，无法进一步确定状态。
* Unfiltered：端口没有被屏蔽，但是出于开放状态还需要进一步确认。
* Open|filtered：端口可能开放也可能屏蔽
* Close|filtered：端口可能关闭也可能屏蔽

nmap [选项或者参数] 目标主机

参数：

-sS/sT/sA/sW/sM：指定使用TCP SYN/Connect()/ACK/Windows/Maimon scans来对目标主机

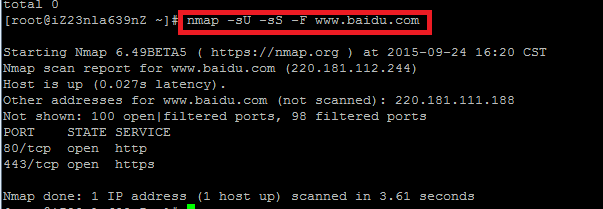
-sU：指定使用UDP扫描方式扫描目标主机的UDP端口状况

-sN/sF/sX：指定使用TCP Null、FIN、Xmas scans秘密扫描方式来协助侦测目标主机的TCP端口状态

-p<port ranges>：仅仅扫描指定的一个或者一批端口。例如-p T:80-88,U:53,S:9其中T代表TCP协议，U表示UDP协议，S表示SCTP协议。

-F：快速扫描模式，仅仅扫描开放率最高的100个端口

--top-ports<number>：进扫描开放率最高的number个端口。



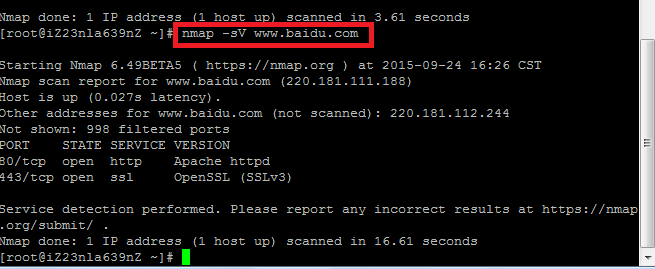
## nmap版本侦测

nmap的版本侦测主要用来确认主机开放端口对应的应用程序和版本信息，支持TCP/UDP协议，支持多种平台的服务侦测。

用法：nmap [选项] 目标主机

参数：

* -sV：设置nmap进行版本侦测
* --version-intensity<level>：设置版本侦测的强度值，取值范围0~9，默认7.数值越高，探测出的服务版本就越精确，时间也更长。
* --version-light：设置用轻量级侦测方式，相当于强度2
* --version-all：尝试使用所有的probes进行侦测，相当于侦测强度9。.
* --version-trace：显示版本侦测的详细过程。



## nmap操作系统的侦测

用法：nmap 选项 目标主机

参数：

-O：设置nmap进行操作系统侦测

--osscan-guess：猜测目标主机的操作系统类型，nmap给出的可能性的比例，用户根据提供的比例总和判断操作系统的类型。

# Tcpdump抓包命令的使用

tcpdump命令是一款sniffer工具，它可以打印所有经过网络接口的数据包的头信息，也可以使用-w选项将数据包保存到文件中，方便以后分析。

## tcpdump命令的使用

tcpdump (选项)

|  |
| --- |
| -a：尝试将网络和广播地址转换成名称；  -c<数据包数目>：收到指定的数据包数目后，就停止进行倾倒操作；  -d：把编译过的数据包编码转换成可阅读的格式，并倾倒到标准输出；  -dd：把编译过的数据包编码转换成C语言的格式，并倾倒到标准输出；  -ddd：把编译过的数据包编码转换成十进制数字的格式，并倾倒到标准输出；  -e：在每列倾倒资料上显示连接层级的文件头；  -f：用数字显示网际网络地址；  -F<表达文件>：指定内含表达方式的文件；  -i<网络界面>：使用指定的网络截面送出数据包；  -l：使用标准输出列的缓冲区；  -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。  -N：不列出域名；  -O：不将数据包编码最佳化；  -p：不让网络界面进入混杂模式；  -q ：快速输出，仅列出少数的传输协议信息；  -r<数据包文件>：从指定的文件读取数据包数据，达到流量回放的目的；  -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。  -S：用绝对而非相对数值列出TCP关联数；  -t：在每列倾倒资料上不显示时间戳记；  -tt： 在每列倾倒资料上显示未经格式化的时间戳记；  -T<数据包类型>：强制将表达方式所指定的数据包转译成设置的数据包类型；  -v：详细显示指令执行过程；  -vv：更详细显示指令执行过程；  -x：用十六进制字码列出数据包资料；    -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。  -w<数据包文件>：把数据包数据写入指定的文件，和-r的作用相对。 |

## TCP协议中FLAG的含义

TCP FLAG 标记基于标记的TCP包匹配经常被用于过滤试图打开新连接的TCP数据包。 TCP标记和他们的意义如下：

|  |
| --- |
| F : FIN - 结束; 结束会话  S : SYN - 同步; 表示开始会话请求  R : RST - 复位;中断一个连接  P : PUSH - 推送; 数据包立即发送  A : ACK - 应答  U : URG - 紧急  E : ECE - 显式拥塞提醒回应  W : CWR - 拥塞窗口减少 |

服务器A和客户机B

### 三次握手Three-way Handshake(建立连接)

第一次握手：(B) --> [SYN] --> (A)

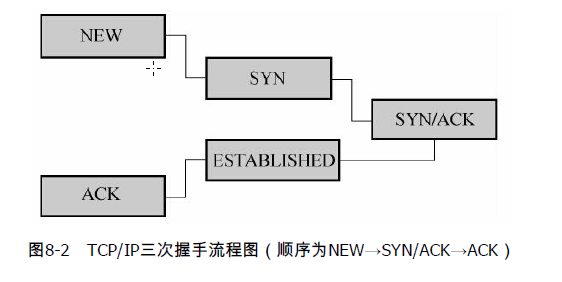
过程解析：B首先向A发一个SYN (Synchronize) 标记的包，告诉A请求建立连接。一个 SYN包就是仅SYN标记设为1的TCP包，只有当A受到B发来的SYN包，才可建立连接。因此，如果你的防火墙丢弃所有的发往外网接口的SYN包，那么你将不 能让外部任何主机主动建立连接。

第二次握手：(B) <-- [SYN/ACK] <--(A)

过程解析：A接收到后会发一个对SYN包的确认包(SYN/ACK)回去，表示对第一个SYN包的确认，并继续握手操作.。（注意SYN/ACK包是仅SYN 和 ACK 标记为1的包）

第三次握手：(B) --> [ACK] --> (A)

解析过程：B收到SYN/ACK 包,B发一个确认包(ACK)，通知A连接已建立。至此，三次握手完成，一个TCP连接完成。（注意：ACK包就是仅ACK 标记设为1的TCP包. 需要注意的是当三此握手完成、连接建立以后，TCP连接的每个包都会设置ACK位）



### 四次握手Four-way Handshake(连接终止)

第一次握手： (B) --> ACK/FIN --> (A) 客户端请求关闭服务端的数据传输

第二次握手：(B) <-- ACK <-- (A) 服务端响应确认报文

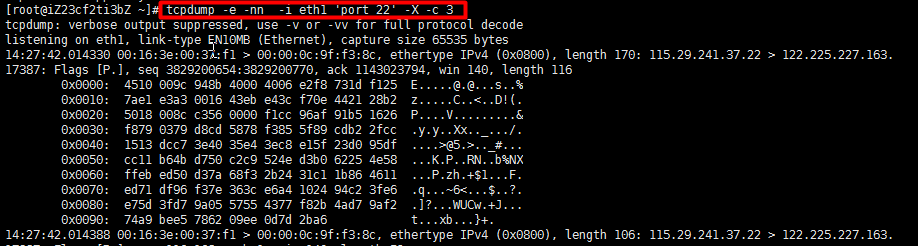
第三次握手： (B) <-- ACK/FIN <-- (A) 服务端请求关闭连接

第四次握手：(B) --> ACK --> (A) 客户端响应

由于TCP连接是双向连接, 因此关闭连接需要在两个方向上做。ACK/FIN 包(ACK 和FIN 标记设为1)通常被认为是FIN(终结)包.然而, 由于连接还没有关闭, FIN包总是打上ACK标记. 没有ACK标记而仅有FIN标记的包不是合法的包，并且通常被认为是恶意的。

## tcpdump的输出结果解析

tcpdump -e -nn -i eth1 'port 22' -X -c 3



第一行:“tcpdump: verbose output suppressed, use -v or -vv for fullprotocol decode”提示使用选项-v和-vv，可以看到更全的输出内容。

第二行“listening on eth1, link-type EN10MB (Ethernet), capture size 65535 bytes”我们监听的是通过eth1这个NIC设备的网络包，且它的链路层是基于以太网的，要抓的包大小限制是65535字节。包大小限制值可以通过-s选项来设置。

第三行” 14:27:42.014330 00:16:3e:00:37:f1 > 00:00:0c:9f:f3:8c, ethertype IPv4 (0x0800), length 170: 115.29.241.37.22 > 122.225.227.163.17387: Flags [P.], seq 3829200654:3829200770, ack 1143023794, win 140, length 116”

14:27:42.014330：分别对应包被抓到的时，分，秒，微妙。

00:16:3e:00:37:f1 > 00:00:0c:9f:f3:8c 表示MAC地址00:16:3e:00:37:f1发送到MAC地址为00:00:0c:9f:f3:8c的主机，

ethertype IPv4 (0x0800)：表示Ethernet帧的协议类型为ipv4(即代码为0x0800)。

” length 170: 115.29.241.37.22 > 122.225.227.163.17387: Flags [P.]：

length 170表示以太帧长度为170。115.29.241.37.22表示这个包的源IP为115.29.241.37，源端口为22，’>’表示数据包的传输方向, 122.225.227.163.17387,表示这个数据包的目的端ip为122.225.227.163,目标端口为17387。Flags是[P]，表明是服务器向客户端传送数据。

seq 3829200654:3829200770表示序列号

” win 140, length 116”win 140 表示窗口大小为140字节，长度116。

下面几行分别是IP,TCP首部

## tcpdump过滤语句

可以给tcpdump传送“过滤表达式”来起到网络包过滤的作用，而且可以支持传入单个或多个过滤表达式。过滤表达式大体可以分成三种过滤条件：“类型”、“方向”和“协议”。还有三种逻辑运算，取非运算是 'not ' '! ', 与运算是'and','&&';或运算是'or' ,'||' ，可以使用逻辑运算符将三种过滤条件搭配组合成我们的过滤表达式。

### 类型过滤

类型的关键字，主要包括host(指定ip地址)，net(指定网络地址)，port(指定端口)。如果没有指定类型，缺省的类型是host。

例如：

* host 210.45.114.211，指定主机 210.45.114.211，
* net 210.11.0.0 指明210.11.0.0是一个网络地址，
* port 21 指明端口号是21。

### 方向过滤

主要包括src, dst ,dst or src, dst and src。如果没有指明方向关键字，则缺省是srcor dst关键字。

举例：

* src 210.45.114.211 ,指明ip包中源地址是210.45.114.211
* dst net 210.11.0.0 指明目的网络地址是210.11.0.0

### 协议过滤

主要包括 ether,ip,ip6,arp,rarp,tcp,udp等类型。如果没有指定任何协议，则tcpdump将会监听所有协议的信息包。

### 其他过滤类型

gateway, broadcast,less,greater

## 常见使用

使用1：想查目标机器端口是21或80的网络包

|  |
| --- |
| tcpdump -i eth0 -c 10 'dst port 21 or dst port 80' |

使用2：截获主机115.29.241.37和主机115.29.210.85或115.29.211.14的通信，使用命令(注意括号的使用)

|  |
| --- |
| tcpdump -i eth1 -c 3 'host 115.29.241.37 and (115.29.210.85 or 115.29.211.14)' |

使用3：想获取使用ftp端口和ftp数据端口的网络包

linux系统下 /etc/services这个文件里面，就存储着所有知名服务和传输层端口的对应关系。如果你直接把/etc/services里的ftp对应的端口值从21改为了3333，那么tcpdump就会去抓端口含有3333的网络包了。

|  |
| --- |
| tcpdump 'port ftp or ftp-data' |

使用4：想要获取主机172.16.0.11除了和主机210.45.123.249之外所有主机通信的ip包

|  |
| --- |
| tcpdump ip ‘host 172.16.0.11 and ! 210.45.123.249’ |

使用5：172.16.0.11的80端口和110和25以外的其他端口的包

|  |
| --- |
| tcpdump -i eth0 ‘host 172.16.0.11 and! port 80 and ! port 25 and ! port 110’ |

使用6：proto [ expr : size]的用法

proto即protocol的缩写：它表示这里要指定的是某种协议名称

expr：用来指定数据报字节单位的偏移量。该偏移量相对于指定的协议层，默认的起始位置是0；而

size：表示从偏移量的位置开始提取多少个字节，可以设置为1、2、4,默认为1字节。如果只设置了expr，而没有设置size，则默认提取1个字节。比如ip[2:2]，就表示提取出第3、4个字节；而ip[0]则表示提取ip协议头的第一个字节。在我们提取了特定内容之后，我们就需要设置我们的过滤条件了，我们可用的“比较操作符”包括：>，<，>=，<=，=，!=，总共有6个。

|  |
| --- |
| tcpdump -i eth0 'host 115.29.241.37 and host baidu.com and tcp[tcpflags]&tcp-syn!=0' -c 3 -nn |