學習網絡請落實這幾款小工具

程序員章魚哥 昨天

以下文章來源於我是程序員小賤，作者L的存在

我是程序員小賤

從面試出發，解剖各個知識點，讓我們向上成長！這裡除了學習，還有音樂，生活攝影，籃...

點擊上方藍色“程序員章魚哥”，關注並設為星標

重磅乾貨，第一時間送達！

來源：我是程序員小賤

在實際開發過程中，熟練使用Linux或者Windows中相關網絡工具，可以更快更準地找到故障。所以，今天就跟大家分享幾個實用的網絡利器。

1 nc

nc-->“瑞士軍刀”。不知大家在滲透過程中，拿了shell有沒有使用nc搞點事兒。它用來快速構建網絡鏈接，常用來調試客戶端程序。

参数	描述
-i	设置数据包传送的时间间隔
-l	以服务器方式运行。默认为客户端运行
-k	重复接受并处理某个端口上的所有链接
-p	以客户端运行时强制其使用指定端口
-C	将CR和LF两个字符作为结束符
-u	使用udp协议。默认tcp协议
-X	nc客户端余代理服务器通信时默认为socks5协议。
-z	扫描目标机器某个范围服务是否开启

nc參數

舉例

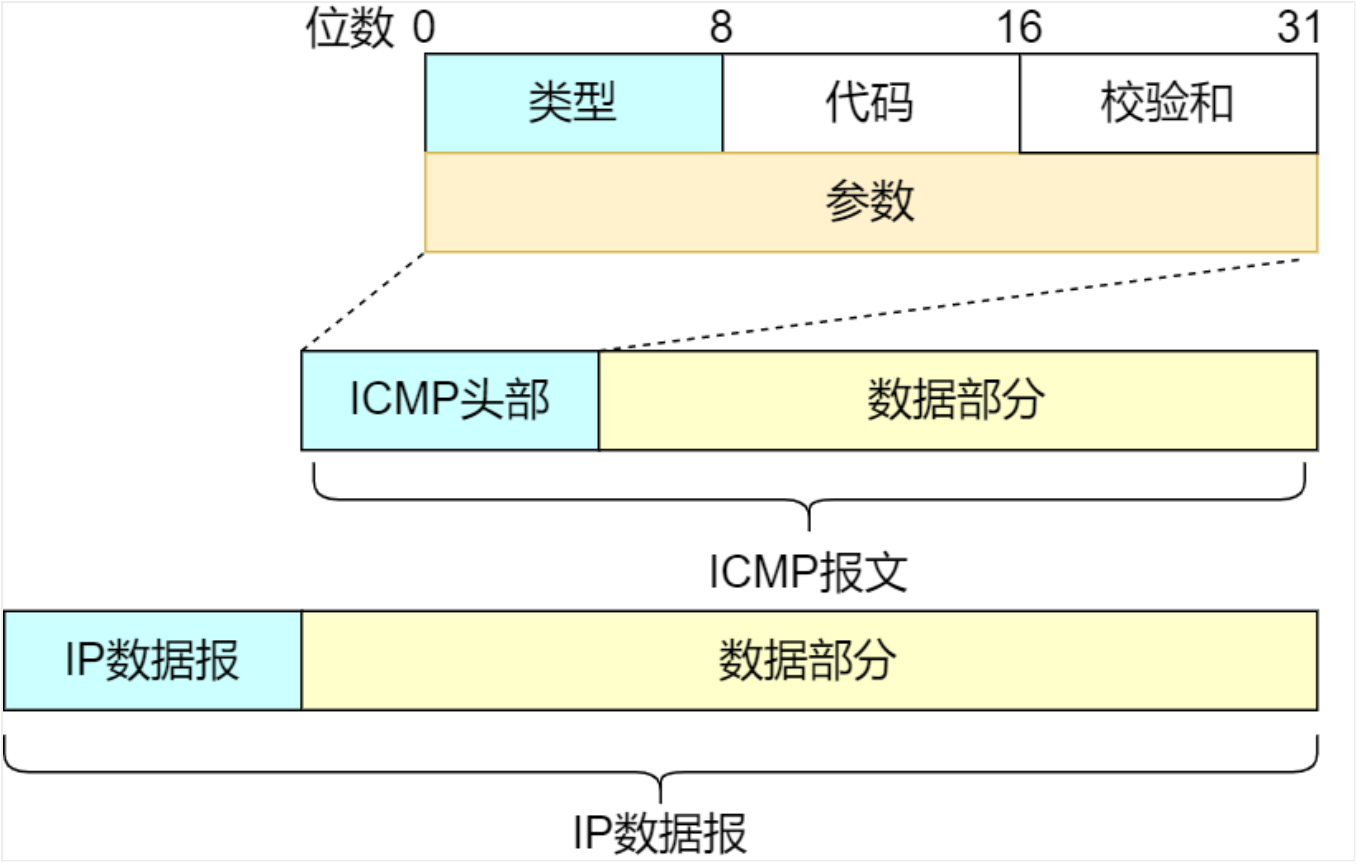
執行任務	命令
掃描機器A端口號在30-40的服務	nc -z A 30-40
連接服務器A端口號為5000	nc -CA 5000
傳送文件	MachineA:nc -v -n ip port E:\a.exe

2 ping

> 用來實現對網絡連通性探測。我們知道網絡上機器有唯一確定的IP地址，給對方發送數據包，根據返回的信息初步判斷目標機器是否存在或者目標機器操作系統是啥。那麼，經常使用的ping，底層原理是什麼，是TCP/UDP?

ping

在具體實現中，其實使用了ICMP協議（網際控制報文協議），它是一種基於IP協議的控制協議。那麼，ICMP協議（網際控制報文協議），其報文什麼樣子呢?



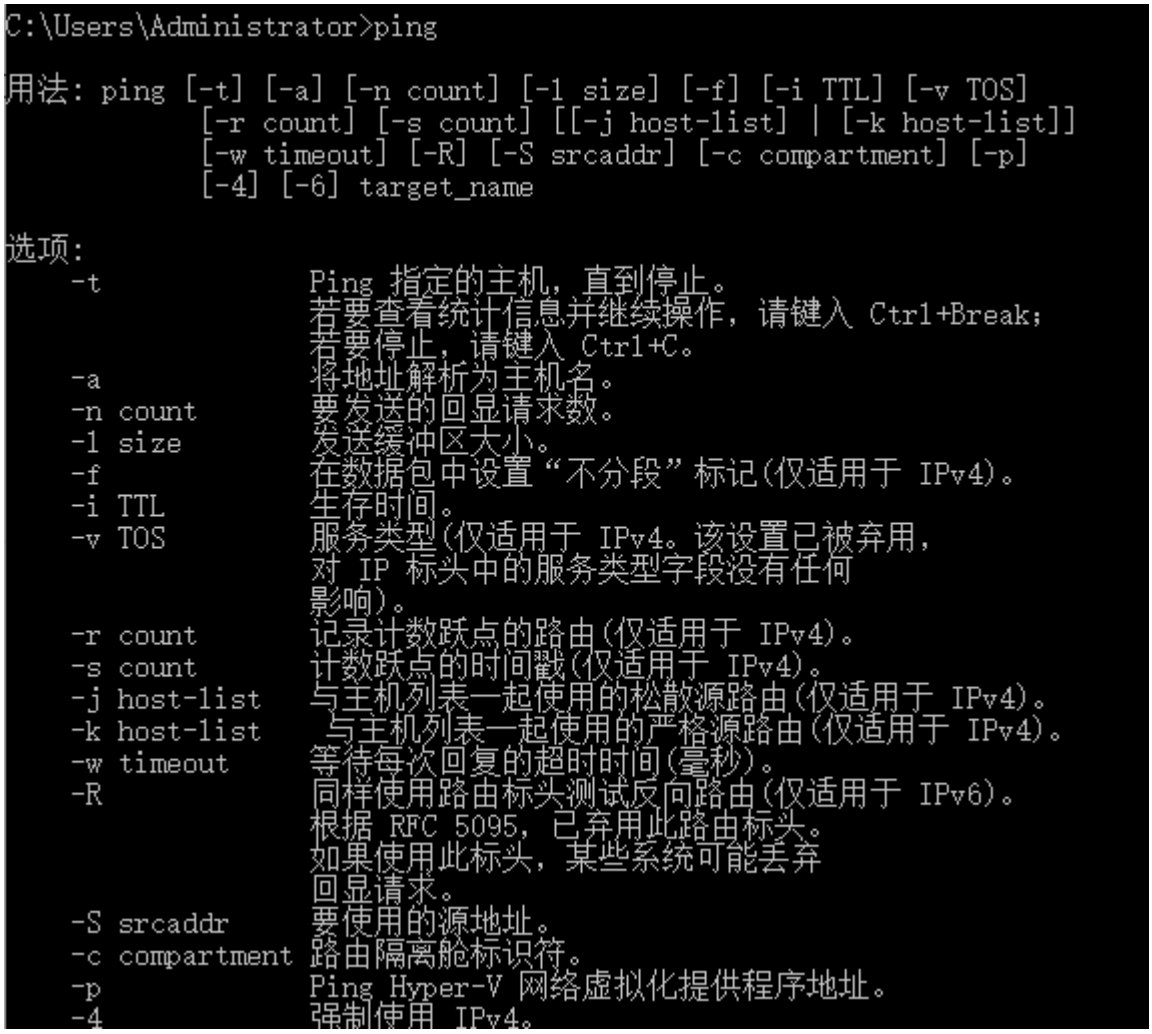
下面分別闡述下字段含義:

- 類型: 表示ICMP的類型，如果為0表示請求類型，為8表示應答。
- 代碼: 用來查找產生錯誤的原因。
- 校驗和: 檢查錯誤的數據。
- 標識符: 使用標識符確認到底是誰發送的控制協議。
- 序列號: 唯一確定的一個報文。

ping命令組裝成上述的IP報文進行發送，報文目的地為ping目的地址，原地址為發送ping主機地址，然後按照ICMP的規則填寫數據。

隨後IP報文通過ARP協議，請求廣播到局域網絡上的所有主機，並接收返回消息，以此確定目標的物理地址。

查看詳細參數



ping參數

常用參數

[-l] ： 定義所發送數據包的大小，默認為32字節。

[-n] ： 定義所發數據包的次數，默認為3次。

[-t] ： 表示不間斷向目標IP發送數據包。

TTL

> TTL 是IP 協議包中的一個值，它告訴網絡路由器，包在網絡中的時間是否太長而應被丟棄。

- TTL設置時間越長，那麼緩存時間也就越長，更新也就越不容易生效。增大TTL可以節約域名解析時間從而加快網站的訪問

- 減小TTL值，減少更換空間時的不可訪問時間

返回值

- Request timed out
  可能出現的情況：
  -對方已經關機或者根本沒有這個地址。
  -可能不在同一個網段，即使通過路由也無法找到對方從而出現超時。
  -對方存在但是設置了防火牆過濾。

- Destination host Unreachable
  可能出現的情況：
  -與對方不在同一個網段且沒有設置默認路由
  -網線出毛病

- Bad IP address
  可能出現的情況：
  - ＩP地址不存在
  -沒有正確連接DNS服務器從而無法解析

3 ifconfig/ip addr

> 查看服務器網卡、IP等信息。

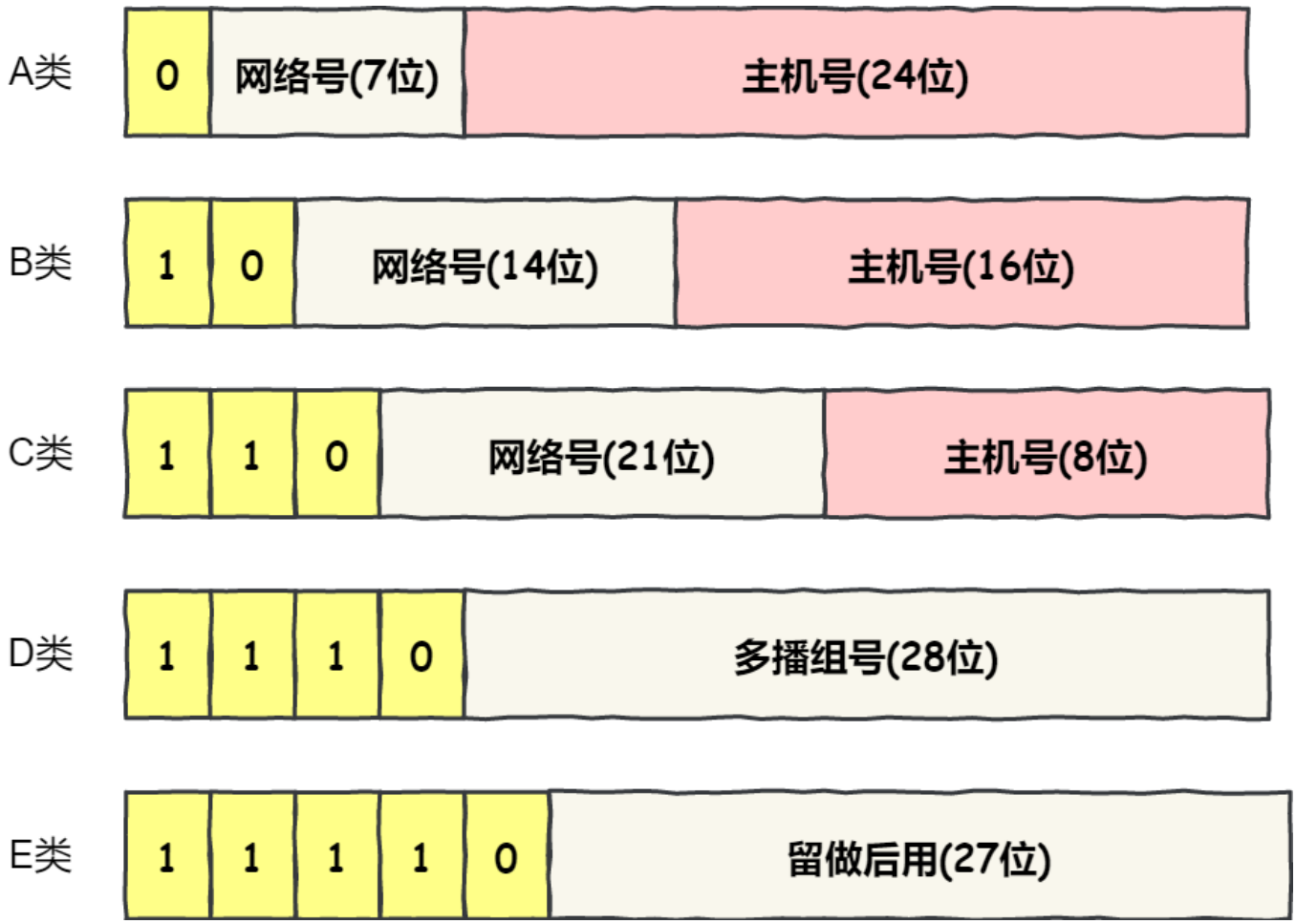
```
[root@slave102 shell]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: em1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether e0:db:55:1f:[illegible]:80 brd ff:ff:ff:ff:ff:ff
    inet [illegible] brd [illegible] scope global noprefixroute em1
       valid_lft forever preferred_lft forever
    inet6 fe80::e2db:55ff:fe1f:9980/64 scope link noprefixroute
```

ipaddr

上圖中被馬賽克的位置假設為10.172.100.3，這樣就是一個IP地址，IP地址按照小數點分割為四部分，每部分佔8字節，所以IP地址為32位。

那麼，這樣的IP地址一共有多少呢？一共有2的32次方個，估算約為42.9億個，除去一些特用的IP和一些不能用的IP，剩下可用的不到40億。為解決網絡地址資源不足的問題，從而出現IPV6，128位的IP地址。

剛開始覺得32位很夠用了，還將其分為5類，如下圖所示：



網絡地址分類

我們再看看各類地址的主機數量是多少?

类别	IP地址范围	最大主机数	私有IP地址范围
A	0.0.0.0-127.255.255.255	16777214	10.0.0.0-10.255.255.255
B	128.0.0.0-191.255.255.255	65534	172.16.0.0-172.31.255.255
C	192.0.0.0-223.255.255.255	254	192.168.0.0-192.168.255.255

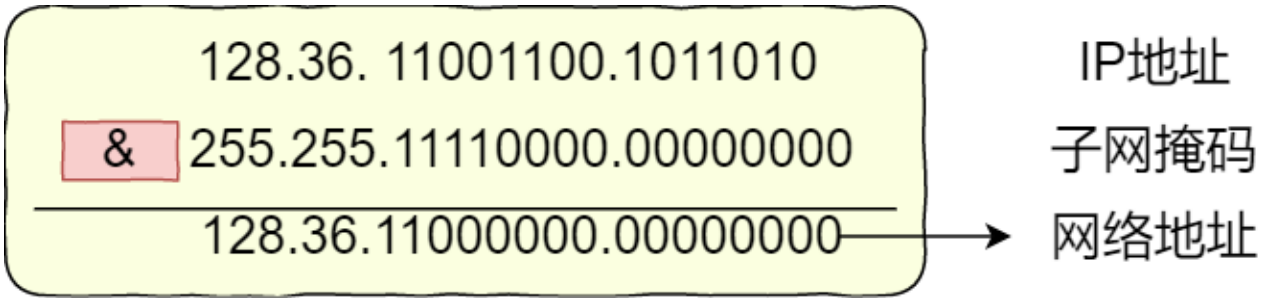
上圖中可知道C類地址太少了吧，但是B類地址又太多，怎麼中和一下嘞?

無類型域間選路

> CIDR地址中包含標準的32位IP地址和有關網絡前綴位數的信息。比如10.172.100.3/24，IP地址斜杠後面數字24，代表前24位是網絡號，後面8位為主機號。

如何得到網絡號?

> 使用IP地址和子網掩碼進行AND計算得到網絡號。



4 tcpdump

> 和它類似的工具在windows中是wireshark，其採用底層庫winpcap/libpcap實現，採用了bpf過濾機制。下面我們看看提供的不同參數的含義。

参数名	含义
-n	使用IP地址表示主机。使用数字表示端口
-i	指定要监听的端口。如果为"-i any"表示住区所有网卡数据包
-v	输出诸如ip数据包中的TTL更加详细的信息
-t	不打印时间戳
-e	显示以太网帧头部信息
-c	仅仅抓取指定数量的数据包
-x	按照十六进制显示数据包内容
-X	不仅仅输出-x结果还输出十六进制对应的ASCII字符
-s	设置抓包时的抓包长度
-w	将输出结果定向到某个文件，一般为pcap后缀
-r	从文件读取数据包并显示

tcpdump

知道了相關參數，下面看幾個案例。

執行任務	命令
捕獲特定網口數據包	tcpdump -i eth0
捕獲特定個數(1000)的包	tcpdump -c 1000 -i eth0
將捕獲的包保存到文件	tcpdump -w a.pcap -i eth0
讀取pcap格式的包	tcpdump -r a.pcap
增加捕獲包的時間戳	tcpdump -n -ttt -i eth0
指定捕獲包的協議類型	tcpdump -i eth0 arp

執行任務	命令
捕獲指定端口	tcpdump -i eth0 post 22
捕獲特定目標ip+port的包	tcpdump -i eth0 dst address and port 22
捕獲DNS請求和響應	tcpdump -i eth0 -s0 port 53
匹配Http請求頭	tcpdump -s 0 -v -n -l | egrep -i "POST /|GET /|Host:"

5 lsof

> 列出當前系統打開的文件描述符工具，可以得知感興趣的描述符是被哪些進程使用。

同樣，我們看看相關參數：

参数	描述
-i	显示sokcet文件描述符
-c	显示指定的命令打开的所有文件描述符
-t	仅显示打开了目标文件描述符的进程pid

lsof

舉例

執行任務	命令
列出所有的網絡鏈接	lsof -i
列出所有udp的網絡鏈接	lsof -i udp
列出誰在使用某個端口	lsof -i :3306

執行任務	命令
列出誰在使用特定的tcp端口	lsof -i tcp:80
根據文件描述範圍列出文件信息	lsof -d 2-3

6 netstat

> netstat是一個網絡信息統計工具。它可以得到網卡接口上全部鏈接、路由表信息、網卡接口信息等。通常在網絡編程中我們用它來顯示TCP連接以及狀態信息。

参数	描述
-n	使用IP地址表示主机
-a	显示结果中包含监听的socket
-t	仅显示TCP连接
-r	显示路由信息
-i	显示网卡接口数据流量
-c	每隔1s输出一次
-o	显示socket定时器的信息
-p	显示socket所属的进程的PID和名字

下面列舉幾個常用例子。

執行任務	命令
列出所有連接	netstat -a
只列出TCP或者UDP	netstat -at/netstat -au
列出監聽中的連接	netstat -tnl

執行任務	命令
獲取進程名、進程號以及用戶ID	nestat -nlpt
打印統計信息	netstat -s
netstat持續輸出	netstat -ct
打印active狀態的連接	netstat -atnp | grep ESTA
查看服務是否運行(npt)	netstat -aple| grep ntp

7 dpkt

> dpkt定義包packet類，它定義了網絡報文類型的基礎類。其中IP、ICMP等繼承於dpkt class，每一個子類有一個_hdr_結構，此結構定義了不同報文的頭部，方便取出相應的控製字段。示例如下：

```
#!/usr/bin/python
#coding=utf-8
import dpkt
import socket
import optparse

def printPcap(pcap):
    # 遍历[timestamp, packet]记录的数组
    for (ts, buf) in pcap:
        try:
            # 获取以太网部分数据
            eth = dpkt.ethernet.Ethernet(buf)
            # 获取IP层数据
            ip = eth.data
            # 把存储在inet_ntoa中的IP地址转换成一个字符串
            src = socket.inet_ntoa(ip.src)
            dst = socket.inet_ntoa(ip.dst)
            print '[+] 源地址: ' + src + ' --> 目标地址: ' + dst
        except:
            pass

def main():
    parser = optparse.OptionParser('[*] Usage : ./pcapTest.py -f <file>')#测试包
```

```
    parser.add_option('-f',dest='fileName',type='string',help='specify target filename'
    (options,args) = parser.parse_args()
    fileName = options.fileName# 取得包名

    if fileName == None:
        print parser.usage
        exit(0)
    else:
        #f = open('geotest.pcap')
        f = open(fileName)
        pcap = dpkt.pcap.Reader(f)
        printPcap(pcap)

if __name__ == '__main__':
    main()
```

8 scapy

> 注意哈，這個是嗅探包，不是爬蟲框架scrapy哈。看看官網怎麼說的，如“強大的交互式包操作工具”、“支持大量協議的包解析和包構造”、“輕鬆取代hping，85%的nmap，arpspoof，tcpdump等等”。不過歸根到底，它說的強大功能，都是基於Scapy是一個強大的網絡數據包操作工具才能實現得了的。這裡只是大概介紹，具體用法官網非常詳細，有助於學習網絡協議。

```
>>> IP()
<IP |>
>>> a=IP(dst="172.16.1.40")
>>> a
<IP dst=172.16.1.40 |>
>>> a.dst
'172.16.1.40'
>>> a.ttl
64
```

往期精彩內容:

悄悄使用！3M/S，還要什麼VIP?

聽聲辨位，一個讓我感到毛骨悚然的GitHub 項目！

GitHub熱榜，打馬賽克也不安全了！AI消除馬賽克so easy!

...

喜歡此內容的人還喜歡

別瞎學了，這幾門語言要被淘汰了！

碼農掃地僧

一款基於Spring Boot 的現代化社區（論壇/問答/社交網絡/博客）

Coding這件小事

推荐一款高颜值的Spring Boot 快速開發框架

碼農掃地僧