等级：博客 4
积分：1312
勋章：
周排名：6万+
总排名：5万+
展开

原创 Linux命令：traceroute命令（路由跟踪）

2017-07-14 17:17:26 南宫小仙僧 阅读数 53444 更多

traceroute是用来检测发出数据包的主机到目标主机之间所经过的网关数量的工具。traceroute的原理是试图以最小的TTL（存活时间）发出探测包来跟踪数据包到达目标主机所经过的网关，然后监听一个来自网关ICMP的应答。发送数据包的大小默认为38个字节。

原理：程序利用增加存活时间（TTL）来实现其功能。每当数据包(3个数据包包括源地址，目的地址和包发出的时间标签)经过一个路由器，其存活时间就会减1。当其存活时间是0时，主机便取消数据包，并传送一个ICMP（Internet控制报文协议。它是TCP/IP协议族的一个子协议，用于在IP主机、路由器之间传递控制消息。控制消息是指网络通不通、主机是否可达、路由是否可用等网络本身的消息。这些控制消息虽然并不传输用户数据，但是对于用户数据的传递起着重要的作用。） TTL数据包给原数据包的发出者。

traceroute程序完整过程：首先它发送一份TTL字段为1的IP数据包给目的主机，处理这个数据包的第一个路由器将TTL值减1，然后丢弃该数据报，并给源主机发送一个ICMP报文（“超时”信息，这个报文包含了路由器的IP地址，这样就得到了第一个路由器的地址），然后traceroute发送一个TTL为2的数据报来得到第二个路由器的IP地址，继续这个过程，直至这个数据报到达目的主机。

1.命令格式:

traceroute [参数] [主机]

2.命令功能:
traceroute指令让你**追踪网络数据包的路由途径**，预设数据包大小是40Bytes，用户可另行设置。
具体参数格式：traceroute [-dFlnrvx][-f<存活数值>][-g<网关>...][-i<网络界面>][-m<存活数值>][-p<通信端口>][-s<来源地址>][-t<服务类型>][-w<超时秒数>][主机名称或IP地址][数据包大小]

3.命令参数:
-d 使用Socket层级的排错功能。
-f 设置第一个检测数据包的存活数值TTL的大小。
-F 设置勿离断位。
-g 设置来源路由网关，最多可设置8个。
-i 使用指定的网络界面送出数据包。
-I 使用ICMP回应取代UDP资料信息。
-m 设置检测数据包的最大存活数值TTL的大小。
-n 直接使用IP地址而非主机名称。
-p 设置UDP传输协议的通信端口。
-r 忽略普通的Routing Table，直接将数据包送到远端主机上。
-s 设置本地主机送出数据包的IP地址。
-t 设置检测数据包的TOS数值。
-v 详细显示指令的执行过程。
-w 设置等待远端主机回报的时间。
-x 开启或关闭数据包的正确性检验。

4 实例:
实例4.1： traceroute www.baidu.com
结果：

```
oai@oai-virtual-machine: ~
traceroute to www.a.shifen.com (183.232.231.173), 64 hops max
  1   192.168.20.2  0.112ms  0.059ms  0.097ms
  2   *  *  *
  3   *  *  *
  4   *  *  *
  5   *  *  *
  6   *  *  *
  7   *  *  *
  8   *  *  *
  9   *  *  *
 10   *  *  *
 11   *  *  *
 12   *  *  *
 13   *  *  *
 14   *  *  *
 15   *  *  *
 16   *  *  *
 17   *  *  *
 18   *  *  *
 19   *  *  *
 20   *  *  *
 21   *  *  *
 22   *  *  *
 23   *  *  *
 24   *  *  *
 25   *  *  *
 26   *  *  *
 27   *  *  *
 28   *  *  *
 29   *  *  *
 30   *  *  *
```

说明:
记录按序列号从1开始，每个纪录就是一跳，每跳表示一个网关，我们看到每行有三个时间，单位是 ms，其实就是-q的默认参数。探测数据包向每个网关发送三个数据包后，网关响应后返回的时间；如果您用 traceroute -q 4 www.58.com，表示向每个网关发送4个数据包。见下图：

```
oai@oai-virtual-machine:~$ traceroute -q 4 www.baidu.com
traceroute to www.a.shifen.com (183.232.231.173), 64 hops max
  1   192.168.20.2  0.003ms  0.001ms  0.001ms  0.001ms
  2   *  *  *  *
```

有时我们traceroute 一台主机时，会看到有一些行是以星号表示的。出现这样的情况，可能是防火墙封掉了ICMP的返回信息，所以我们得不到什么相关的数据包返回数据。
有时我们在某一网关处延时比较长，有可能是某台网关比较阻塞，也可能是物理设备本身的原因。当然如果某台DNS出现问题时，不能解析主机名、域名时，也会 有延时长的现象；您可以加-n 参数来避免DNS解析，以IP格式输出数据。
如果在局域网中的不同网段之间，我们可以通过traceroute 来排查问题所在，是主机的问题还是网关的问题。如果我们通过远程来访问某台服务器遇到问题时，我们用到traceroute 追踪数据包所经过的网关，提交IDC服务商，也有助于解决问题；但目前看来在国内解决这样的问题是比较困难的，就是我们发现问题所在，IDC服务商也不可能帮助我们解决。

实例4.2：跳数设置
命令：traceroute -m 10 www.baidu.com
结果：

```
oai@oai-virtual-machine:~$ traceroute -m 10 www.baidu.com
traceroute to www.a.shifen.com (183.232.231.172), 10 hops max
  1   192.168.20.2  0.270ms  0.080ms  0.081ms
  2   *  *  *
```

```
 3   *  *  *
 4   *  *  *
 5   *  *  *
 6   *  *  *
 7   *  *  *
 8   *  *  *
 9   *  *  *
10   *  *  *
oai@oai-virtual-machine:~$
```

说明:通过结果可以看到存活数值=10，当存活数值=0时，主机便取消数据包

实例3：探测包使用的基本UDP端口设置6888

命令： traceroute -p 6888 www.baidu.com

结果：

```
oai@oai-virtual-machine:~$ traceroute -p 6888 www.baidu.com
traceroute to www.a.shifen.com (183.232.231.173), 64 hops max
  1   192.168.20.2  0.002ms  0.002ms  0.001ms
  2   *  *  *
  3   *  *  *
  4   *  *  *
  5   *  *  *
  6   *  *  *
  7   *  *  *
  8   *  *  *
  9   *  *  *
 10   *  *  *
 11   *  *  *
 12   *
```

实例4：把对外发探测包的等待响应时间设置为3秒

命令：traceroute -w 3 www.baidu.com

结果：

```
oai@oai-virtual-machine:~$ traceroute -w 3  www.baidu.com
traceroute to www.a.shifen.com (183.232.231.173), 64 hops max
  1   192.168.20.2  0.065ms  0.052ms  0.058ms
  2   *  *  *
  3   *  *  *
  4   *  *  *
  5   *  *  *
  6   *  *  *
  7   *  *  *
  8   *  *  *
```

本文参考资料：

1.http://www.cnblogs.com/peida/archive/2013/03/07/2947326.html

知识扩展：

目的主机接受在接收到TTL值为1的IP数据包是不会丢失的吧，这样也不会产生一个超时的ICMP数据报文了，那么程序如何判断是否已经到达目的主机了呢?

在Linux下，traceroute程序发送一个UDP数据报给目的主机，但是它选择一个不可能的值作为UDP端口号(大于30000)，使目的主机的任何一个应用程序都不可能使用该端口，因此该数据报到达目的主机时，目的主机会产生一个“端口不可达”错误的ICMP报文，这样traceroute程序要做的就是区分接收到的ICMP报文是超时还是端口不可达，从而来区分是路由器还是目的主机

知识扩展参考资料：http://www.cnblogs.com/cotyb/p/5341439.html

文章最后发布于: 2017-07-14 17:17:26