Linux系统如何抓包

PLAT 马永鸿 2020-12-09 发表

问题描述

在linux系统中，自带网络数据包截获分析工具。支持针对网络层、协议、主机、网络或端口的过滤。并提供and、or、not等逻辑语句帮助去除无用的信息。

解决方法

命令：tcpdump - dump traffic on a network

1 不指定任何参数

tcpdump

2 监听特定网卡

监听第一块网卡上经过的数据包。主机上可能有不止一块网卡，所以经常需要指定网卡。

查看一下本机的网卡，可以看到，接入管理网络的ip地址为192.168.113.238，对应的网卡为eth0

```
[root@mayonghong ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.113.238  netmask 255.255.255.0  broadcast 192.16
8.113.255
        inet6 fe80::f99d:84dd:398f:fca4  prefixlen 64  scopeid 0x20<l
ink>
        inet6 fe80::fe17:6c19:4ba4:6c04  prefixlen 64  scopeid 0x20<l
ink>
        ether 0c:da:41:1d:21:fe  txqueuelen 1000  (Ethernet)
        RX packets 164468  bytes 16640543 (15.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9159  bytes 6975599 (6.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 765186  bytes 363838148 (346.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 765186  bytes 363838148 (346.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.122.1  netmask 255.255.255.0  broadcast 192.168.
122.255
        ether 52:54:00:d7:0d:03  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

tcpdump -i eth0

3. 监听特定主机

例子：监听本机跟主机192.168.113.251之间往来的通信包。

备注：出、入的包都会被监听。

tcpdump host 192.168.113.251

```
[root@mayonghong ~]# tcpdump host 192.168.113.251
tcpdump: verbose output suppressed, use -v or -vv for full protocol d
ecode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 b
ytes
14:20:51.367571 ARP, Request who-has 192.168.113.251 tell mayonghong,
 length 28
14:20:51.368965 ARP, Reply 192.168.113.251 is-at 2c:23:3a:b7:93:03 (o
ui Unknown), length 46
14:20:54.760890 ARP, Request who-has 192.168.113.254 tell 192.168.113
.251, length 46
14:21:08.440469 ARP, Request who-has 192.168.113.251 tell 192.168.113
.207, length 46
14:21:33.110827 ARP, Request who-has 192.168.113.251 tell 192.168.113
.126, length 46
14:21:39.641121 ARP, Request who-has 192.168.113.254 tell 192.168.113
.251, length 46
14:21:48.336183 ARP, Request who-has 192.168.113.251 tell 192.168.113
.102, length 28
14:21:51.293637 IP mayonghong > 192.168.113.251: ICMP echo request, i
d 45874, seq 7680, length 40
14:21:51.294879 IP 192.168.113.251 > mayonghong: ICMP echo reply, id
45874, seq 7680, length 40
14:21:56.295588 ARP, Request who-has 192.168.113.251 tell mayonghong,
 length 28
14:21:56.296968 ARP, Reply 192.168.113.251 is-at 2c:23:3a:b7:93:03 (o
ui Unknown), length 46
```

4. 特定来源、目标地址的通信

特定来源

tcpdump src host hostname

特定目标地址

tcpdump dst host hostname

如果不指定src跟dst，那么来源 或者目标 是hostname的通信都会被监听

下图为监听源为192.168.113.251的报文，可以看到有SNMP的消息。

```
[root@mayonghong ~]# tcpdump src host 192.168.113.251
tcpdump: verbose output suppressed, use -v or -vv for full protocol d
ecode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 b
ytes
14:27:15.975524 IP 192.168.113.251 > mayonghong: ICMP echo reply, id
45874, seq 10240, length 40
14:27:20.986096 ARP, Reply 192.168.113.251 is-at 2c:23:3a:b7:93:03 (o
ui Unknown), length 46
14:27:21.811520 ARP, Request who-has 192.168.113.254 tell 192.168.113
.251, length 46
14:27:35.433274 IP 192.168.113.251.snmp > mayonghong.54513:  GetRespo
nse(54)  system.sysUpTime.0=330672995 E:25506.2.6.1.1.1.1.8.192=36
14:28:21.650096 IP 192.168.113.251 > mayonghong: ICMP echo reply, id
45874, seq 10752, length 40
14:28:26.664936 ARP, Reply 192.168.113.251 is-at 2c:23:3a:b7:93:03 (o
ui Unknown), length 46
14:28:45.391506 IP 192.168.113.251 > mayonghong: ICMP echo reply, id
58418, seq 0, length 40
14:28:48.839360 IP 192.168.113.251.snmp > mayonghong.54513:  GetRespo
nse(208)  system.sysDescr.0=48_50_20_43_6f_6d_77_61_72_65_20_50_6c_61
_74_66_6f_72_6d_20_53_6f_66_74_77_61_72_65_2c_20_53_6f_66_74_77_61_72
_65_20_56_65_72_73_69_6f_6e_20_37_2e_31_2e_30_34_35_2c_20_52_65_6c_65
_61_73_65_20_33_31_30_39_50_30_35_0d_0a_48_50_20_35_31_33_30_2d_32_34
_47_2d_50_6f_45_2b_2d_34_53_46_50_2b_20_28_33_37_30_57_29_20_45_49_20
_53_77_69_74_63_68_0d_0a_43_6f_70_79_72_69_67_68_74_20_28_63_29_20_32
_30_31_30_2d_32_30_31_35_20_48_65_77_6c_65_74_74_2d_50_61_63_6b_61_72
_64_20_44_65_76_65_6c_6f_70_6d_65_6e_74_20_43_6f_6d_70_61_6e_79_2c_20
_4c_2e_50_2e
```

5. 特定端口

tcpdump port 3000

6. 监听TCP/UDP

服务器上不同服务分别用了TCP、UDP作为传输层，假如只想监听TCP的数据包

tcpdump tcp

7. 来源主机+端口+TCP

监听来自主机192.168.113.251在端口22上的TCP数据包

tcpdump tcp port 22 and src host 192.168.113.251

8. 监听特定主机之间的通信

tcpdump ip host 192.168.113.251 and 192.168.113.252

192.168.113.251除了和192.168.113.252之外的主机之间的通信

tcpdump ip host 192.168.113.251 and ! 192.168.113.252

9. 稍微详细点的例子

tcpdump tcp -i eth0 -t -s 0 -c 100 and dst port ! 22 and src net 192.168.113.0/24 -w ./target.cap

(1)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

(2)-i eth0 : 只抓经过接口eth0的包

(3)-t : 不显示时间戳

(4)-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

(5)-c 100 : 只抓取100个数据包

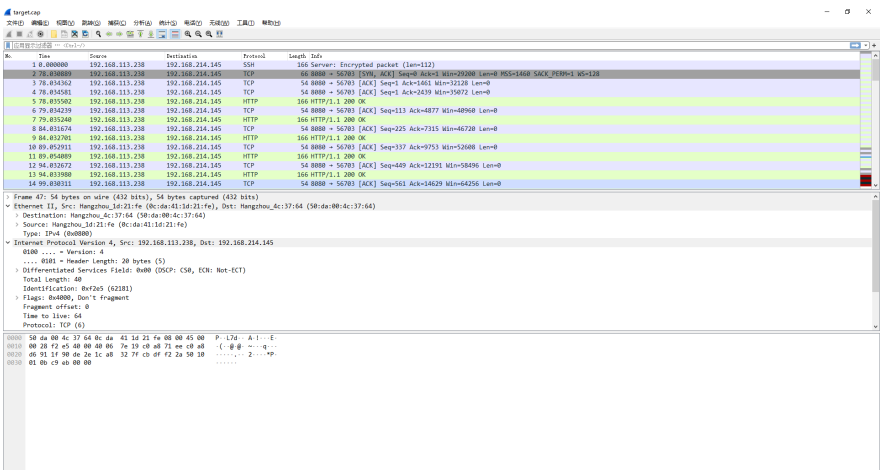
(6)dst port ! 22 : 不抓取目标端口是22的数据包

(7)src net 192.168.113.0/24 : 数据包的源网络地址为192.168.113.0/24

(8)-w ./target.cap : 保存成cap文件，方便用ethereal(即wireshark)分析

```
[root@mayonghong ~]# tcpdump tcp -i eth0 -t -s 0 -c 100 and dst port
! 22 and src net 192.168.113.0/24 -w ./target.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size
 262144 bytes
^C60 packets captured
71 packets received by filter
0 packets dropped by kernel
```

将其下载到电脑中，可用wireshark软件进行分析。



为了分析在iMC平台上服务器与被管理设备的SNMP交互过程，命令如下：

tcpdump -i eth0 -s 0 -c 100 host 192.168.113.251 -w ./myh1.cap

```
[root@mayonghong ~]# tcpdump -i eth0 -s 0 -c 100 host 192.168.113.251
 -w ./myh1.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size
 262144 bytes


100 packets captured
101 packets received by filter
0 packets dropped by kernel
```

用wireshark软件打开进行分析