Command line: Using tcpdump to find scanning activity

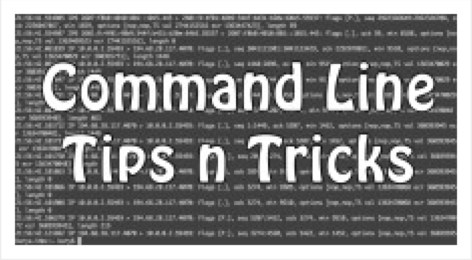
By Kary | text

May

There are fancy tools to help

09

find scanning activity, but we talk about Wireshark and packet analysis here, so let's talk about finding scanning activity if all you have is the command line. When I say "command line" I mean a shell like bash on Linux, Mac, or Cygwin on Windows.



Let's say you have a pcap of the network activity. One legit reason to use command line is if the pcap is very large and Wireshark would choke on it. I'll show you a series of commands that will identify IPs that talk to lots of different hosts and ports.

First, here's how to find a source IP initiating lots of connections:

```
$ tcpdump -nr traffic.pcap 'tcp[tcpflags]=2' | awk '{print
$3}' | cut -d. -f1,2,3,4 | sort | uniq -c | sort -nr
```

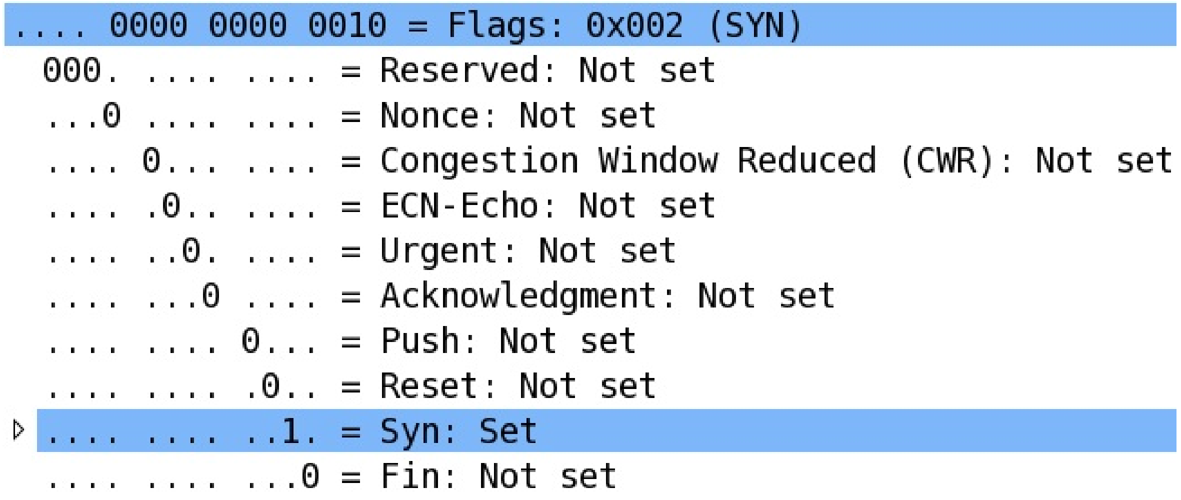
Let's break it down:

```
-nr traffic.pcap
```

means read from a file called “traffic.pcap” and do not try to convert addresses to names.

```
'tcp[tcpflags]=2'
```

is the BPF filter that only selects packets with the SYN flag set. If only the SYN bit is set, then the value of the TCP flags is 2.



```
.... 0000 0000 0010 = Flags: 0x002 (SYN)
   000. .... .... = Reserved: Not set
   ...0 .... .... = Nonce: Not set
   .... 0... .... = Congestion Window Reduced (CWR): Not set
   .... .0.. .... = ECN-Echo: Not set
   .... ..0. .... = Urgent: Not set
   .... ...0 .... = Acknowledgment: Not set
   .... .... 0... = Push: Not set
   .... .... .0.. = Reset: Not set
 ▷ .... .... ..1. = Syn: Set
   .... .... ...0 = Fin: Not set
```

The output of tcpdump on my Mac looks like:

```
21:56:50.433266 IP 10.0.0.2.42875 > 45.33.32.174.443: Flags
[S], seq 2690402238, win 1024, options [mss 1460], length 0
```

We want the source IP which is the 3rd field, so we use **awk** to grab it but you might need to adjust if your output looks different

```
awk '{print $3}'
```

That will get us the source IP and source port but we don't care about the source port, only the IP. Let's split the string 10.0.0.2.42875 but only keep the first 4 fields

```
cut -d. -f 1,2,3,4
```

You could do the last two steps in one go with **awk**, but I'd have to google it and well, no thanks.

Next we need to sort the source IPs and feed them into **uniq** which will count the unique source IPs. Then we reverse sort the output

```
sort | uniq -c | sort -nr
```

and get something like

```
190232 10.0.0.2
253    10.0.0.10
14     10.0.1.45
5      10.0.5.12
```

So 10.0.0.2 has sent 190232 SYN packets i.e. connection initiations.

Let's figure out who this IP is talking to

```
$ tcpdump -nr traffic.pcap 'tcp[tcpflags]=2 and src host
10.0.0.2' | awk '{print $5}' | cut -d. -f1,2,3,4 | sort |
uniq -c | wc -l
```

Notice I changed the filter to include source IP and changed the `awk` to grab the 5th field which is the destination IP. I also changed the end to just count the number of unique IPs this host is trying to talk to

```
    10983
```

So 10.0.0.2 has tried to connect to 10983 unique destination IPs.

What if there is only a small number of destination IPs or even just one? Maybe some IP e.g. 10.1.2.3 is port scanning your server 192.168.0.10. So for a particular source and destination IP pair, lets see how many ports are hit

```
$tcpdump -nr traffic.pcap 'tcp[tcpflags]=2 and src host 10.1.2.3 and dst host 192.168.0.10' | awk '{print $5}' | cut -d. -f5 | sort | uniq -c | wc -l
```

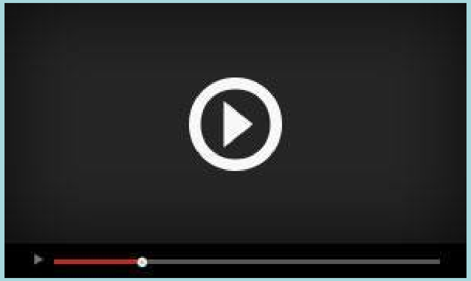
```
     1004
```

This time I used `cut` to grab the destination port only. So we can see that source IP 10.1.2.3 has sent a SYN packet to 1004 different ports on destination IP 192.168.0.10. Port scan, sho nuff.

You can do this with `tshark` and not need the `awk` and `cut`, but `tcpdump` is faster and again, if the pcap is very large `tshark` could choke on it.

I'm not a shell master, so please suggest more efficient versions in the comments. And if you're a Windows Powershell wizard, maybe you can suggest a way to accomplish this same thing.

Share this post! Spread the packet gospel!



Does Wireshark make you feel overwhelmed?

FREE 4 part Wireshark video minicourse:

- ✓ Learn how to focus on the real problem
- ✓ Learn why you shouldn't capture on the client
- ✓ Learn how to setup Wireshark to find root cause fast

Name

Email

Get Course

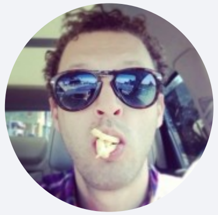
Related

How Can the Packet Size Be Greater than the MTU?
August 15, 2014
In "text"

Understanding the tcptrace Time-Sequence Graph in Wireshark
June 3, 2014
In "text"

Troubleshooting a One-Way Performance Issue
August 16, 2015
In "case study"

Follow



About the Author

I like being the hero. Being able to drop a bucket of root cause analysis on a burning network problem has made me a hero (to some people) and it feels real good, y'all. Get good at packet analysis and be the hero too. I also like french fries.

Leave a Comment:

Name *

E-Mail *

Website

□ Save my name, email, and website in this browser for the next time I comment.

Comments:

POST COMMENT

« Previous Post

Solving Tomcat Throughput Issues on Windows

Next Post »

How to Do TCP Sequence Number Analysis

RECENT POSTS

The Network vs the Application: Who's to Blame?

Troubleshooting Slow FTP Uploads

Troubleshooting a One-Way Performance Issue

Troubleshooting MTU Problems With Wireshark

When is a Fast Retransmission Not a Fast Retransmission?

RECENT COMMENTS

- Solomon on Troubleshooting MTU Problems With Wireshark
- david.woo on How Can the Packet Size Be Greater than the MTU?
- ed on The Network vs the Application: Who’s to Blame?
- Kary on How to Troubleshoot Throughput and TCP Windows
- something cool on How to Hack a Cisco Router ACL

RECENT COMMENTS

- Solomon on Troubleshooting MTU Problems With Wireshark
- david.woo on How Can the Packet Size Be Greater than the MTU?
- ed on The Network vs the Application: Who’s to Blame?
- Kary on How to Troubleshoot Throughput and TCP Windows

something cool on How to Hack a Cisco Router ACL

RECENT POSTS

- The Network vs the Application: Who’s to Blame?
- Troubleshooting Slow FTP Uploads
- Troubleshooting a One-Way Performance Issue
- Troubleshooting MTU Problems With Wireshark
- When is a Fast Retransmission Not a Fast Retransmission?

FOLLOW PACKETBOMB

ATTRIBUTION

Shark photo via Sylke Rohrlach