TCPDUMP: a simple cheatsheet

Written by Andrea Fortuna • on July 18, 2018 • in Networking

RECENT POSTS

How to upgrade BIOS on a Lenovo laptop running linux

October 8, 2019

My Weekly RoundUp #110

Having a solid grasp of tcpdump is mandatory for anyone desiring a thorough understanding of TCP/IP.

What is tcpdump?

Tcpdump is one of th best network analysis tool for information security professionals.

tcpdumpruns under the command line and allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached. Is a Free Software, originally written in 1988 by **Van Jacobson**, **Sally Floyd**, **Vern Paxson** and **Steven McCanne** who were, at the time, working in the **Lawrence Berkeley Laboratory Network Research Group**.

tcpdump is distributed under the BSD license.

Why tcpdump?

Many prefer to use higher level analysis tools such as Wireshark, but I believe that when using a tool that displays network traffic in a raw format the burden of analysis is placed directly on the human rather than the application, allowing the analyst to perform a more deeper research.

This kind of approach require a deeper understanding of the TCP/IP suite,so start using tcpdump instead of other tools whenever possible!

October 7, 2019

Watch out! A new vulnerability in WhatsApp for Android allows attackers to perform remote commands on devices

October 4, 2019

Some thoughts about Windows 10 “Timeline” forensics artifacts

October 3, 2019

CATEGORIES

Select Category

The basics

Here a few options you can use when using tcpdump.
Using this options, we will try to build some simple usecases.

Options

-i any : Listen on all interfaces just to see if you're seeing any traffic.
-i eth0 : Listen on the eth0 interface.
-D : Show the list of available interfaces
-n : Don't resolve hostnames.
-nn : Don't resolve hostnames or port names.
-q : Be less verbose (more quiet) with your output.
-t : Give human-readable timestamp output.
-tttt : Give maximally human-readable timestamp output.
-X : Show the packet's contents in both hex and ASCII.
-XX : Same as -X, but also shows the ethernet header.
-v, -vv, -vvv : Increase the amount of packet information you get back.
-c : Only get x number of packets and then stop.
-s : Define the size of the capture in bytes. Use -s0 to get everything, unless you are intentionally capturing less.
-S : Print absolute sequence numbers.
-e : Get the ethernet header as well.
-q : Show less protocol information.
-E : Decrypt IPSEC traffic by providing an encryption key.

Now, a brief excerpt about expressions, that allows you to trim out various types of traffic and find exactly what you're looking for.

There are three main types of expression: ***type, dir, and proto***.

Type options are: host, net, and port.
Direction lets you do src, dst, and combinations thereof.
Proto(col) lets you designate: tcp, udp, icmp, ah, and many more.

The Use Cases

Now, let's try using this information in real usecases:

---

tcpdump -D

Listing possible network interfaces on the system

```
$ tcpdump -D
1.eth0
2.eth1
3.eth2
```

---

tcpdump -i interface-name

Capture packets from a particular interface

```
tcpdump -i eth1
```

tcpdump -c N

Capture only N number of packets

```
tcpdump -i eth1 -c 10
```

tcpdump -w file.pcap

Capture the packets and write into a file

```
tcpdump -i eth1 -w tmp.pcap
```

tcpdump -s 0

Capture and store network frames full-length

```
tcpdump -i eth1 -w tmp.pcap -s 0
```

tcpdump -r file.pcap

Reading the packets from a saved file

```
tcpdump -tttt -r tmp.pcap
```

tcpdump -tttt

Capture packets with proper readable timestamp

```
tcpdump -i eth1 -tttt
```

tcpdump greater N

Read packets longer than N bytes

```
tcpdump -i eth1 -w tmp.pcap greater 1024
```

Specify protocol type

To receive only the packets of a specific protocol type – fddi, tr, wlan, ip, ip6, arp, rarp, decnet, tcp and udp

```
tcpdump -i eth1 arp
```

tcpdump host IP

Will show you traffic from 1.2.3.4, whether it's the source or the destination.

```
tcpdump host 1.2.3.4
```

tcpdump src/dst

Filtering by source and sestination: it's quite easy to isolate traffic based on either source or destination using `src` and `dst`.

```
tcpdump src 2.3.4.5
tcpdump dst 3.4.5.6
```

tcpdump net x.x.x.x/xx

Filter packets by network: you can combine this with the `src` or `dst` options as well.

```
tcpdump net 1.2.3.0/24
```

tcpdump port PORT_NO

Receive packets flows on a particular port

```
tcpdump -i eth1 port 22
tcpdump -i eth1 src port 1026
```

---

tcpdump less/greater

Filter traffic based on Packet Size: you can use less, greater, or their associated symbols that you would expect from mathematics.

```
tcpdump -i eth1 less 32
tcpdump -i eth1 greater 64
tcpdump -i eth1 <= 128
```

---

tcpdump dst IPADDRESS and port PORT-NO

Capture packets for particular destination IP and Port

```
tcpdump -i eth1 dst 10.181.140.216 and port 22
```

tcpdump -vvv

Display more packet information

```
E.g. tcpdump -i eth1 -vvv
```

tcpdump -e

Display link level header of every packet: -e

```
tcpdump -i eth1 -e -t
listening on eth2, link-type EN10MB (Ethernet), capture size 65535 bytes
52:54:00:e1:1c:10 (oui Unknown) > 01:80:c2:00:00:00 (oui Unknown), 802.3, length 60
52:54:00:e1:1c:10 (oui Unknown) > 01:80:c2:00:00:00 (oui Unknown), 802.3, length 60
```

tcpdump -t

Don't print a timestamp on each dump lin: without using **-t** option we can see the below output timestamp is dumped.

```
tcpdump -i eth2

listening on eth2, link-type EN10MB (Ethernet), capture size 65535 bytes
08:44:51.295229 STP 802.1d, Config, Flags [none], bridge-id 8000.52:54:00:e1:1c:10.
08:44:53.296795 STP 802.1d, Config, Flags [none], bridge-id 8000.52:54:00:e1:1c:10.
```

and with **-t** option:

```
tcpdump -i eth2 -t

listening on eth2, link-type EN10MB (Ethernet), capture size 65535 bytes
STP 802.1d, Config, Flags [none], bridge-id 8000.52:54:00:e1:1c:10.8003, length 43
STP 802.1d, Config, Flags [none], bridge-id 8000.52:54:00:e1:1c:10.8003, length 43
```

---

tcpdump -n

Display packets with IP address instead of DNS names: -nBasically tcpdump converts the plain address to DNS names. Using n option we can make tcpdump to display ip address.

```
tcpdump -i eth1 -n
```

---

tcpdump -A

Display Captured Packets in **ASCII**

```
tcpdump -i eth1 -A
```

---

tcpdump -XX

Display Captured Packets in **HEX** and **ASCII**

```
tcpdump -i eth1 -XX
```

---

tcpdump -nnvXSs 0 -c1 icmp

Hex output: useful when you want to see the content of the packets in question, and it's often best used when you're isolating a few candidates for closer scrutiny.

Some everyday examples

tcpdump can output content in **ASCII**, so you can use it to search for cleartext content using other command-line tools like `grep`.

The `-l` switch lets you see the traffic as you're capturing it, and helps when sending to commands like `grep`.

Find HTTP User Agents

```
tcpdump -vvAls0 | grep 'User-Agent:'
```

Cleartext GET Requests

```
tcpdump -vvAls0 | grep 'GET'
```

Find HTTP Host Headers

```
tcpdump -vvAls0 | grep 'Host:'
```

Find HTTP Cookies

```
tcpdump -vvAls0 | grep 'Set-Cookie|Host:|Cookie:'
```

Find SSH Connections

This one works regardless of what port the connection comes in on, because it's getting the banner response.

```
tcpdump 'tcp[(tcp[12]>>2):4] = 0x5353482D'
```

Find DNS Traffic

```
tcpdump -vvAs0 port 53
```

Find FTP Traffic

```
tcpdump -vvAs0 port ftp or ftp-data
```

Find NTP Traffic

```
tcpdump -vvAs0 port 123
```

Find Cleartext Passwords

```
tcpdump port http or port ftp or port smtp or port imap or port pop3 or port telne
```

References and further readings

- **Manpage of TCPDUMP**
- **Lawrence Berkeley Laboratory Network Research Group**
- **TCPDUMP HomePage**

Related posts

Share this:

Like this:

Loading...

TAGS

COMMAND LINE CYBERSECURITY NETWORKING SECURITY TCPDUMP

PRINT

Andrea Fortuna

Finding malware on memory dumps using Volatility and Yara rules

2 COMMENTS

CSPS Protocol

January 29, 2019 at 9:45 am

Nice blog info. thanks for the lovely posting.

REPLY

SysAdmin

June 10, 2019 at 3:25 pm

If you need to get rid of strings like

STP 802.1d, Config, Flags [none], bridge-id 8000.00:30:19:2c:93:41.8041, length 42

in the tcpdump's output try adding "not ether host 01:00:0c:cc:cc:cd" filter (exact value can be found via -e flag).

REPLY

COMMENTS

Enter your comment here...

This site uses Akismet to reduce spam. Learn how your comment data is processed.