Comparative Study of two Most Popular Packet Sniffing Tools- Tcpdump and Wireshark

Piyush Goyal[1] and Anurag Goyal[2]
Amity School of Engineering and Technology, Amity University, NOIDA
[1]piyush.goyal2021@gmail.com, [2]apnv2021@gmail.com

Abstract- With the ever expanding sphere of Internet and its applications, the scope of Networking, data transfer and data security too have tremendously increased. This has led to sophisticated tools that are though useful in cyber mitigation but are also widely used by cyber criminals to eavesdrop or gain illegal access. This Statement stands true for Network monitoring and Packet Sniffing tools. Though, they were designed to assist the network administrators in better assessing the servers, traffic and diagnosing the issues but they have become the favorite tool of hackers to scan a particular network and sniff on unprotected data. White Hat hackers use these tools to prevent such attacks by criminals as they identify and filter out malicious packets and their source. In this paper, we have thoroughly compared two of the most widely used open source packet sniffing and network monitoring tools- Wireshark and Tcpdump.

Keywords- *Packet Sniffer, Promiscuous mode, Wireshark, Tcpdump, Monitor mode, Network sensing.*

1. Introduction-

Packet Sniffers or analyzers are a computer program that can intercept the incoming and outgoing traffic in a network. They can even decode the raw data of packet passing through as a data stream, thus exposing details of what other devices connected to internet on a Local network are doing. These sniffers can also store packet details for a later use.

There are broadly 2 modes in packet sniffing- Promiscuous mode and Monitor mode. In Promiscuous mode, all the network data that is destined or not destined to a controller device (the device running packet sniffer) is captured by the Network Interface Controller (NIC) or Wireless NIC in case of Wi-Fi. In non-promiscuous mode, only the data destined for a particular controller through MAC addresses is sent to CPU, the rest packets are dropped. In Monitor mode, a user need not associate himself with a particular access point. Computer having Wireless Network Interface Controller will enable a user to collect all the packets that are not even in the local network in which user may be connected. This helps in capturing the handshake protocol packets as well. [1]

2. Key Features of Tcpdump and Wireshark

Although, there are hundreds of packet sniffers present in the market according to different industry and personal requirements, but of all these, Tcpdump and Wireshark are most widely used and liked tools.

2.1 Tcpdump

One of the finest and lightest network traffic capturing tool freely available is Tcpdump. It is quite useful for the network learners and enthusiasts who want a thorough understanding of TCP/IP as it dumps the packets in raw format without much analysis.[2] This task of analyzing the raw data by human mind leads to clarity in the networking and packet concepts. For the sake of this purpose, Tcpdump provides many options where details of Packets captured can be viewed in several formats like '-t' option gives the human readable timestamp output, '-X' displays Hex and ASCII of packet contents, '-v' '-vv' and '-vvv' increases packet information to be displayed.

2.2 Wireshark

The most popular network sniffing tool is Wireshark. It is accredited to its simple and graphical interface and its powerful capturing and filtering options. This tool is also freely available just like Tcpdump.

Wireshark has the ability to scan any sort of networks- Ethernets, Wi-Fi, monitor mode or even Bluetooth. This is the primary reason why Wireshark is globally used in industry giants as not only a sniffer but as an Intrusion Detection System(IDS). This tool can swiftly detect and report the Denial of Service(DOS) attacks, attacks on Firewalls or any breach of security through networks.

978-1-5090-5001-7/17/$31.00 ©2017
DOI 10.1109/CICN.2017.19

Wireshark can go up to the depths of bits of packets. It also separates the different network protocol packets by highlighting protocols with different color schemes like green for HTTP, blue for DNS etc.

3. Tcpdump vs. Wireshark

3.1 Options

- Tcpdump is capable of selectively dumping the packets based on matching strings, numbers or even a C program fragment with –dd or –ddd options [3] whereas Wireshark will dump all the packets and then apply the matching filter.
- Wireshark can dump the packets in multiple files with a single command –b. [4] It can also create a ring buffer as per the choice of user. This option is unavailable in Tcpdump.
- Wireshark can collect and print different types of statistics and display a window that is updated in semi-real time through –z command [4] whereas no such feature of statistics exists in Tcpdump.
- Tcpdump can decrypt IPsec ESP packets if compiled with cryptography. This is done by –E command. [3] This feature is absent in Wireshark.
- Immediate mode is present in Tcpdump but absent from Wireshark. This mode enables packets to directly reach the terminal window rather than buffer. This is default in Tcpdump.
- Verbosity (amount of packet information) can be managed on command line in Tcpdump with –v, -vv or –vvv commands whereas it is done by Wireshark in graphical Interface.

3.2 Memory

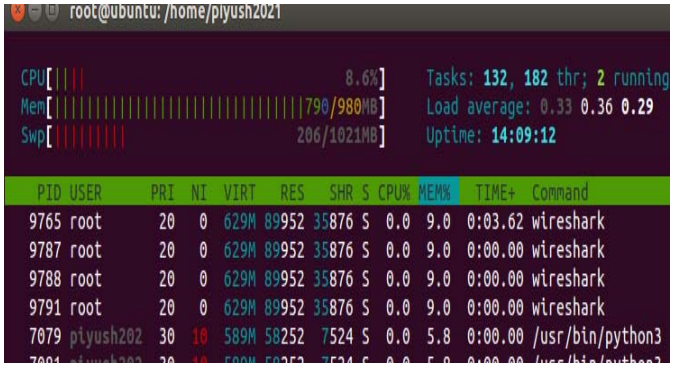

Fig 3.2.1 Memory used by Wireshark

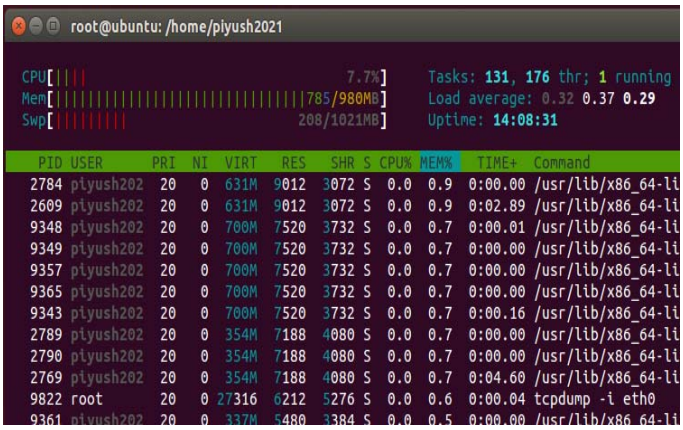

Fig 3.2.2 Memory used by Tcpdump

After analyzing both the tools several times by running in the Htop utility available in Ubuntu, the above cases are received that should be regarded as average of several cases performed on both tools while they were scanning in Ethernet mode (eth0).

VIRT (Virtual Size) - The total program size in memory.

The VIRT of Tcpdump is 27-29 Mb approximately while that of Wireshark is more than 20 times, that is 620-640 Mb.

RES (Resident Size)- The actual physical memory consumed by process.

The RES of Tcpdump is 5.8-6.3 Mb while that of Wireshark is 87-91 Mb.

SHR (Shared Memory)- The sharable size of VIRT.

SHR of Tcpdump is 5.1-5.3 Mb while that of Wireshark is 35-36 Mb.

MEM% (Memory Percentage)- Percentage of memory used by process

Tcpdump has Mem% of 0.6% while Wireshark has 8.8-9.6% share in memory.

From above analysis, it is crystal clear that every sort of memory being used by Wireshark is several times that of what is being used by Tcpdump. Since, memory is a crucial parameter in assessing a tool, Tcpdump wins this league.

3.3 Power Usage

Usage	Wakeups/s	GFX Wak/s	Category	Description
1.0 ms/s	15.8		Process	[khubd]
1.5 ms/s	14	0.1	Process	wireshark
1.3 ms/s	9.6	0.1	Process	/usr/lib/vmware-tools/s
604.0 us/s	7.1		Timer	hrtimer_wakeup
226.5 us/s	4.2		Timer	tick_sched_timer
167.8 us/s	3.9		Process	/usr/bin/dumpcap -n -i
3.1 ms/s	1.7		Interrupt	[18] uhci_hcd:usb2
0.9 ms/s	2.3	0.5	Process	compiz
1.6 ms/s	1.7		Process	[kworker/0:0]
0.0 us/s	1.9		kWork	pm_runtime_work
37.2 us/s	1		Process	tcpdump -i eth0

Fig 3.3.1 Wakeups/s & Events/s of Wireshark and Tcpdump

Power usage is another important factor in comparing the efficiency of tools. The tools that use minimum energy and give maximum output are said to be more efficient.

The utility tool Powertop was used to analyze the battery and processor usage by these two tools and the results were nearly same all the time.

Wireshark is using hundreds of times of more energy than Tcpdump. This becomes evident as the above results were taken after several iterations and a long duration of time. Also Powertop –calibrate command was run prior to get the most reliable results. While Tcpdump is using 37-38 microseconds per second of processor, the Wireshark is using about 1.5 microsecond per second of processor.

Also, Wireshark is having 14 wakeups/second while Tcpdump has only 1. Graphics wakeups/second of Wireshark is 0.1 which is Not applicable in Tcpdump due to lack of Graphical interface.

Usage	Events/s	Category	Description
0.1%	15.8	Process	[khubd]
0.2%	14	Process	wireshark
0.1%	9.6	Process	/usr/lib/vmware-tools/sbin64/vmtoolsd -n vmusr
0.1%	7.1	Timer	hrtimer_wakeup
0.0%	4.2	Timer	tick_sched_timer
0.0%	3.9	Process	/usr/bin/dumpcap -n -i eth0 -y EN10MB -Z none
0.3%	1.7	Interrupt	[18] uhci_hcd:usb2
0.1%	2.3	Process	compiz
0.2%	1.7	Process	[kworker/0:0]
0.0%	1.9	kWork	pm_runtime_work
0.0%	1	Process	tcpdump -i eth0

Fig 3.3.2 Battery usage by Wireshark and Tcpdump

The battery being used by Tcpdump is negligible, that is 0% while battery used by Wireshark is 0.2%.

Due to huge amounts of packets in monitor mode, the usage becomes manifold in both the cases but this usage has increased almost equally in both the tools.

Usage	Events/s	Category	Description
312.4 ms/s	11.8	Process	/usr/bin/X -core
169.7 ms/s	5.9	Process	compiz
131.8 ms/s	12.8	Process	gnome-terminal
252.2 µs/s	23.6	kWork	ieee80211_iface_w
2.3 ms/s	22.7	Process	[rcu_sched]
33.4 ms/s	3.0	Process	wireshark
31.3 ms/s	0.00	Process	nautilus -n
22.2 ms/s	0.00	Process	/usr/bin/ibus-dae
2.6 ms/s	4.9	Timer	hrtimer_wakeup
14.7 ms/s	0.00	Interrupt	PS/2 Touchpad / K
2.5 ms/s	3.9	Process	/usr/lib/vmware-t
7.8 ms/s	1.0	Timer	tick_sched_timer
0.0 µs/s	3.9	kWork	ath9k_led_work
7.2 ms/s	0.00	Interrupt	[17] ehci_hcd:usb
7.2 ms/s	0.00	Interrupt	[6] tasklet(softi
6.9 ms/s	0.00	Process	tcpdump -i mon0
655.4 µs/s	2.0	kWork	flush_to_ldisc

Fig 3.3.3 Statistics in monitor mode

From above analysis, it is clear that Tcpdump beats Wireshark in Efficiency related to power and processor usage.

3.4 Speed of Capture

The speed of capture of packets as found out by using timers and running both the tools simultaneously in the same local machine resulted in almost the same output.

In monitor mode, due to very high capture rate, the number of packets being dropped by Tcpdump are large. But by increasing the snaplen, we can make this zero. Then comparing the two tools fetch the results that Wireshark is 2-3% faster in capturing packets than Tcpdump.

Though in monitor mode, this does not sound like much of a difference but while scanning the network thoroughly and finding out the malicious packet, even dropping or not sensing of one packet makes a huge difference. This is sensitive issue that Wireshark is able to capture more packets in the same time.

In Ethernet mode, the packet capture of Wireshark was equal to Tcpdump if the network is having less traffic, that is less than 1000 packets in 60 seconds. If the number of packets increases, Wireshark captures more with 0.5-1% gain.

This analysis shows that Wireshark beats Tcpdump in the speed of packet capturing.

3.5 Post Capture

An important part of study revolves around what has to be done with the packets and how the tools help us in analyzing the packets. This is the main purpose of using these tools and the tool that saves our time in analyzing the packet content is preferred over the other.

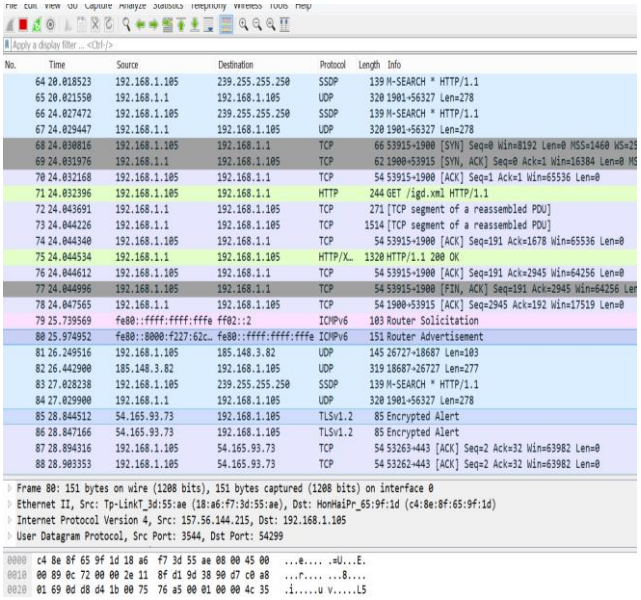

Fig 3.5.1 Wireshark Sample Capture

```
root@ubuntu:/home/piyush2021# tcpdump -s 1500 -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 1500 bytes
13:40:20.723603 IP6 fe80::5853:ac1f:1c11:2bd9.dhcpv6-client > ff02::1:2.dhcpv6-s
erver: dhcp6 solicit
13:40:20.881009 IP 192.168.159.144.49522 > 192.168.159.2.domain: 37240+ PTR? 2.0
.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa. (90)
13:40:20.887211 ARP, Request who-has 192.168.159.144 tell 192.168.159.2, length
46
13:40:20.887247 ARP, Reply 192.168.159.144 is-at 00:0c:29:d9:30:37 (oui Unknown)
, length 28
13:40:20.887350 IP 192.168.159.2.domain > 192.168.159.144.49522: 37240 NXDomain
0/1/0 (154)
13:40:20.902282 IP 192.168.159.144.11155 > 192.168.159.2.domain: 1523+ PTR? 9.d.
b.2.1.1.c.1.f.1.c.a.3.5.8.5.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. (90)
13:40:20.904834 IP 192.168.159.2.domain > 192.168.159.144.11155: 1523 NXDomain 0
/1/0 (139)
13:40:20.905690 IP 192.168.159.144.54115 > 192.168.159.2.domain: 43651+ PTR? 2.1
59.168.192.in-addr.arpa. (44)
13:40:20.908631 IP 192.168.159.2.domain > 192.168.159.144.54115: 43651 NXDomain
0/1/0 (94)
13:40:20.909090 IP 192.168.159.144.50649 > 192.168.159.2.domain: 54822+ PTR? 144
.159.168.192.in-addr.arpa. (46)
13:40:20.911879 IP 192.168.159.2.domain > 192.168.159.144.50649: 54822 NXDomain
0/1/0 (96)
13:40:21.723181 IP6 fe80::5853:ac1f:1c11:2bd9.dhcpv6-client > ff02::1:2.dhcpv6-s
erver: dhcp6 solicit
13:40:23.724153 IP6 fe80::5853:ac1f:1c11:2bd9.dhcpv6-client > ff02::1:2.dhcpv6-s
erver: dhcp6 solicit
13:40:25.916197 ARP, Request who-has 192.168.159.2 tell 192.168.159.144, length
28
13:40:25.916488 ARP, Reply 192.168.159.2 is-at 00:50:56:f6:eb:4e (oui Unknown),
length 46
13:40:27.725027 IP6 fe80::5853:ac1f:1c11:2bd9.dhcpv6-client > ff02::1:2.dhcpv6-s
erver: dhcp6 solicit
13:40:35.725848 IP6 fe80::5853:ac1f:1c11:2bd9.dhcpv6-client > ff02::1:2.dhcpv6-s
erver: dhcp6 solicit
13:40:51.726904 IP6 fe80::5853:ac1f:1c11:2bd9.dhcpv6-client > ff02::1:2.dhcpv6-s
erver: dhcp6 solicit
```

Fig 3.5.2 Tcpdump Sample Capture

As visible from the pictures above, due to its powerful graphical interface, Wireshark is able to separate different types of packets with color codes. The codes are decided by the type of packets, their relation and protocols. For example, router solicitation and advertisement is color coded in pink, HTTP in green, TCP handshaking in grey and so on.

On the other hand, Tcpdump does not differentiate different types of packets and leaves it to the user to classify each packet.

The amount of information can be adjusted with a click in Wireshark whereas, it must be predefined with –v or –vv or –vvv in Tcpdump.

Once capturing has been started, it is difficult to group and apply filters in Tcpdump, whereas it can be done in real time in Wireshark very easily.

If the command to save packets in a file is not specified prior to running capture in Tcpdump, the packets will remain in terminal window only, whereas Wireshark provides the facility to save the packets later on. It even gives a warning if we try quitting without saving the packets.

4. Conclusion

From the above findings, it can be concluded that Tcpdump outclasses Wireshark in Terms of Battery Usage, memory usage and Processor usage but Wireshark is more capable in analyzing the packets captured and the speed of capture.

5. Practicality of this Research

These analyses are helpful in selecting Packet Sniffers as per the requirements. If the Sniffers are needed in a company where they are to be run all the time as an IDS, [5] then Wireshark is preferred as it can detect malicious packets automatically and report immediately.
If thorough understanding is needed of packets and packet sensing, then Tcpdump should be chosen as it presents data in raw format.
If the requirement in an organization is of packet sniffer that doesn't burden the systems in terms of memory and battery, then Tcpdump is the best option.
If an Organization requires simple and graphical interface for non-technical persons, then Wireshark must be chosen.

Like this, these analyses can help in several ways for selecting the appropriate Packet Sniffing tool.

6. References

1. Asrodia, Pallavi, and Hemlata Patel. "Analysis of various packet sniffing tools for network monitoring and analysis." *International Journal of Electrical, Electronics and Computer Engineering* 1.1 (2012): 55-58.

2. Asrodia, Pallavi, and Hemlata Patel. "Network traffic analysis using packet sniffer." *International Journal of Engineering Research and Applications* 2.3 (2012): 854-856.

3. *www.Tcpdump.org*

4. *www.wireshark.org*

5. Qadeer, Mohammed Abdul, et al. "Network traffic analysis and intrusion detection using packet sniffer." *Communication Software and Networks, 2010. ICCSN'10. Second International Conference on*. IEEE, 2010.

6. [Cisco5606] Cisco Systems, " Simple Network Management Protocol", Internetworking Technologies Handbook, Chpt 56, 1992--2006 http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm

7. Tierney, Brian L, "Self-Configuring Network Monitor A High Performance Network Engineering Proposal: Network Measurement and Analysis", For the period June 1, 2001 - May 31, 2004 http://dsd.lbl.gov/Net-Mon/SCNM-proposal.pdf