The facilities of detection by using a tool of Wireshark

Sarah R. Hashim[1], Rusul A. Enad[1], Alyaa M. Al-Khafagi[1], Noor Kamil Abdalhameed[2]
[1]Department of Pharmacy, Al-Zahraa University for Women, Karbala, Iraq
[2]Department of Medical Devices Engineering, Al-Turath University, Baghdad, Iraq

Article Info

Article history:

Received Dec 8, 2022
Revised Feb 22, 2023
Accepted Feb 28, 2023

Keywords:

Data sniffing
Internet protocol address
Intrusion detection
Traffic
WireShark tool

ABSTRACT

Wireshark is easy for using as a packet inspection tool, in additional the feature of packets colorizing is easy for a various type of traffic. This paper exemplifies how Wireshark is used in networks as a tool. To clarify the effectiveness of malicious packet identification in any network, an experiment was conducted. Using the Wireshark program, testing was carried out in real time through experimentation and analysis. Inferences were drawn that clearly show Wireshark's capabilities as a tool in a powerful system for discovering the breach. The functionality of Wireshark is to analyze the network protocol and its open-source features for enabling the addition of likely tasks in the detecting devices were emphasized. Wireshark's skills for handling and interpreting packet data have been highlighted and the access control list (ACL) filtering has been the main application of Wireshark.

Corresponding Author:

Sarah R. Hashim
Department of Pharmacy, Al-Zahraa University for Women
Karbala, Iraq
Email: sarah.rafil@alzahraa.edu.iq

1. INTRODUCTION

Recently, mobile gadgets have seen a lot of use. Therefore, research in the fields of computer and mobility is crucial. Users of such devices have security as one of their top concerns [1]. A fundamental worry for network security, is the improbable and undesired admission of malevolent users and or harmful data packets [2]. The fundamental building blocks of every communication system are data packets. Thus, network security also entails data packet security. The most fundamental building element of communication, a data packet streamlines the flow of its countless duplicates to convey data from one device to another [3]. A data segment, which also contains other data like as the protocol being used, the target hardware address, and contains a data packet. In a nutshell, by examining its contents, it is possible to determine the identity of packet pending from any shady source. Packet sniffing is the study of identifying and only examining a data segment's and its packet's contents. Packet logging is the process of compiling this data into a log. A packet analyzer is a piece of computer hardware or software that may intercept and record data traveling over a digital network or a section of one [4]. The sniffer intercepts each packet as data streams pass through the network, decodes it, and then examines its content in accordance with the necessary specifications [5].

Keeping an eye on network resources in order to spot unusual activity and abuse is the aim of packet sniffing. This idea has had a sharp increase in acceptance and integration into the infrastructure for overall information security [6], [7]. With the advent of computer security risk, the detection as a concept was created. [7]-[9] included crucial data that might be useful for detecting abuse and comprehending user behavior. Host-based intrusion detection was introduced as a result of his work. Another intrusion detection system has been disclosed by the authors of [10]. This project built an intrusion detection system (IDS) that compared audit data to predefined patterns to analyze it. The concept of network intrusion detection has been defined in [11], [12].

In the early 1990s, commercial development of intrusion detection technologies started. With its host-based Stalker family of IDS tools, Haystack labs was the first commercial IDS tool provider. In spite of this, commercial intrusion detection systems took a while to develop and didn't really take off until the second half of the decade.

Wireshark is used in this research to examine the operation of a packet analyzer as well as packet sniffing and logging techniques, Figure 1 showed the Wireshark architecture. A popular open source network protocol analyzer is called Wireshark [13], [14]. An IDS is any packet sniffer or logger with the additional capability of identifying hostile network activity [15]-[19]. Additionally, an IDS typically maintains a database of known attack signatures and may identify when a signature and recent or current behavior are closely matched by comparing patterns of activity and traffic [20]-[22].

The IDS can then send out alarms or alerts at that point. A pattern that fits a known malware is called a signature. In this study, a test problem was created, and on the basis of the experiment's findings, reasonable inferences regarding Wireshark's potential as an IDS were made.

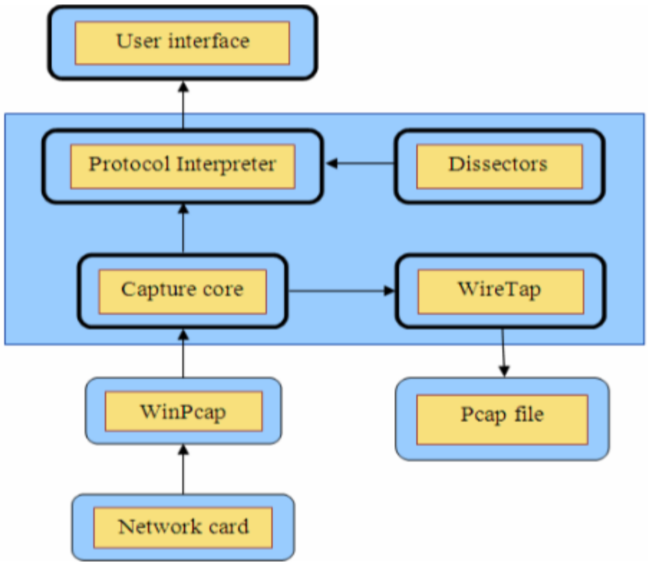
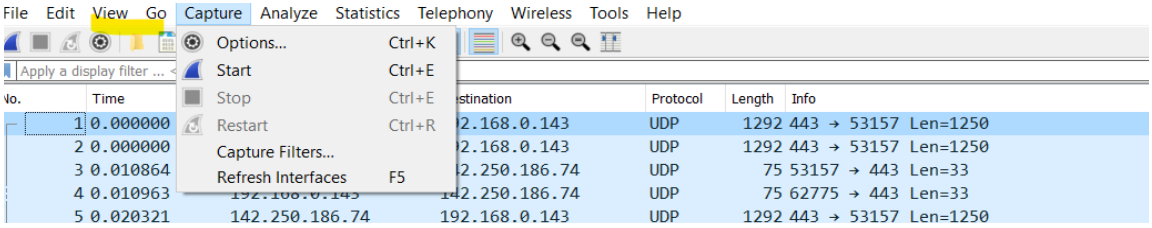


Figure 1. Wireshark architecture

2. WIRESHARK AND SNIFFING TOOLS

The world's most popular network protocol analyzer is Wireshark [23], [24]. It has a wide range of supported operating systems, including Windows, OS X, Linux, and UNIX, and provides a powerful feature set. Network experts, security experts, developers, and teachers all over the world commonly utilize it. It is available as open source for free and is released in accordance with the rules of the GNU General Public License version 2. It is an illustration of a disruptive technology developed and supported by a global group of protocol experts. Wireshark used to go by the name ethereal. The software package Wireshark is a free packet sniffer. The features of Wireshark allow for the collection, viewing, and analysis of data packets [25]. The extensive wireless protocol analysis functionality offered by Wireshark enables administrators to troubleshoot wireless networks. Administrators can use Wireshark to gather traffic "from the air" and decode it into a format that makes it simpler to spot the issues causing sluggish performance, unpredictable connectivity, and other common troubles.

It's not too difficult to set up traditional network sniffing on an ethernet network. A new packet of traffic has been recorded begins on a Wireshark-running analysis workstation in a shared environment. There are numerous wired and wireless methods, encompassing a wide range of topologies and protocols, for connecting a node to a network. Users of Wireshark have the option to record all packets passing through a specific interface at a specific time and over the entire network. The capture tool is one of the main tools. Any of the nodes' available interfaces can be made capture-able by using the interface as seen in Figure 2. The options tab offers a more complex approach for each individual interface. The possibilities of browsing through packets in the capture list are provided by the go menu items.

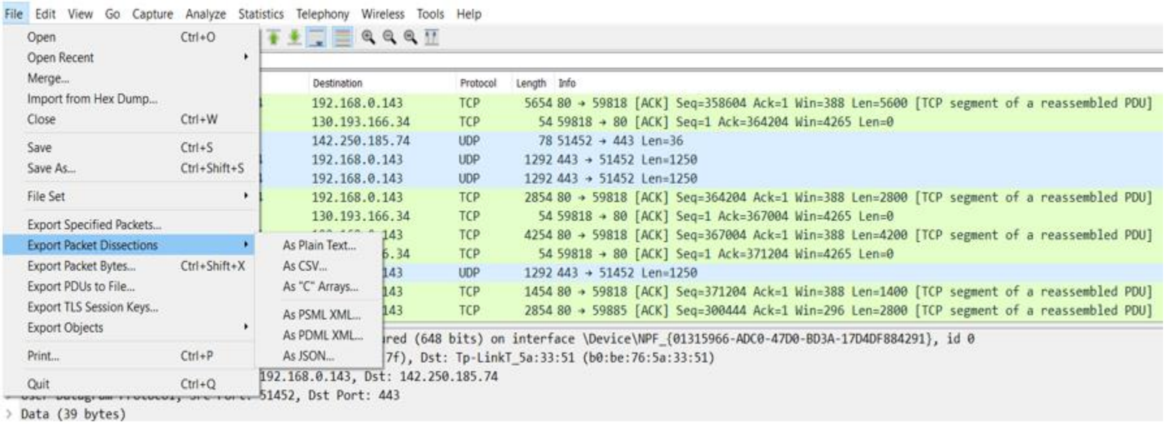


```
No.   Time        ...            ...stination     Protocol  Length Info
    1 0.000000                   ...2.168.0.143   UDP       1292 443 → 53157 Len=1250
    2 0.000000                   ...2.168.0.143   UDP       1292 443 → 53157 Len=1250
    3 0.010864                   ...2.250.186.74  UDP         75 53157 → 443 Len=33
    4 0.010963                   ...2.250.186.74  UDP         75 62775 → 443 Len=33
    5 0.020321  142.250.186.74   192.168.0.143    UDP       1292 443 → 53157 Len=1250
```

Figure 2. The capture tool

3. LOGGING TOOLS

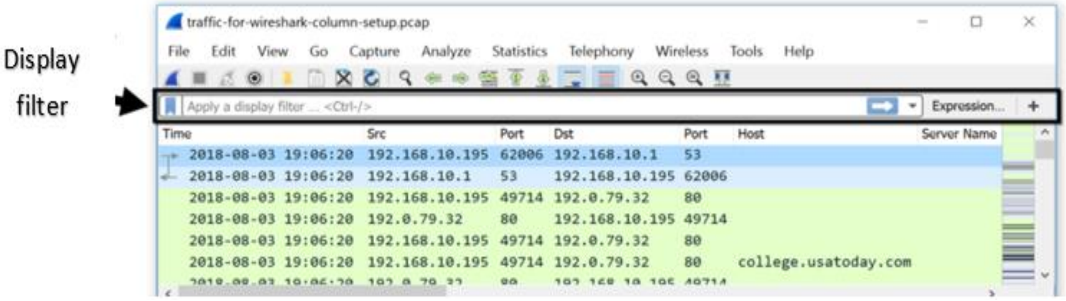
In terms of log maintenance, Wireshark surpasses other IDS or intrusion prevention systems (IPS) devices with incredible versatility. Depending of the network and the devices capacity, log file can be collected hourly or weekly. Consequently, it is simple to collect files via a fast dispensation node and transmit them to a moderate database. Depending on the analyzer tool used, as shown in Figure 3, an additional interesting story is the capability for exporting the capture into a variability of other and easier-to-understand formats, such as plain text, and CSV.



```
Destination      Protocol  Length Info
192.168.0.143    TCP       5654 80 → 59818 [ACK] Seq=358604 Ack=1 Win=388 Len=5600 [TCP segment of a reassembled PDU]
130.193.166.34   TCP         54 59818 → 80 [ACK] Seq=1 Ack=364204 Win=4265 Len=0
142.250.185.74   UDP         78 51452 → 443 Len=36
192.168.0.143    UDP       1292 443 → 51452 Len=1250
192.168.0.143    UDP       1292 443 → 51452 Len=1250
192.168.0.143    TCP       2854 80 → 59818 [ACK] Seq=364204 Ack=1 Win=388 Len=2800 [TCP segment of a reassembled PDU]
130.193.166.34   TCP         54 59818 → 80 [ACK] Seq=1 Ack=367004 Win=4265 Len=0
...143           TCP       4254 80 → 59818 [ACK] Seq=367004 Ack=1 Win=388 Len=4200 [TCP segment of a reassembled PDU]
...6.34          TCP         54 59818 → 80 [ACK] Seq=1 Ack=371204 Win=4265 Len=0
...143           UDP       1292 443 → 51452 Len=1250
...143           TCP       1454 80 → 59818 [ACK] Seq=371204 Ack=1 Win=388 Len=1400 [TCP segment of a reassembled PDU]
...143           TCP       2854 80 → 59885 [ACK] Seq=300444 Ack=1 Win=296 Len=2800 [TCP segment of a reassembled PDU]

...red (648 bits) on interface \Device\NPF_{01315966-ADC0-47D0-BD3A-17D4DF884291}, id 0
...7f), Dst: Tp-LinkT_5a:33:51 (b0:be:76:5a:33:51)
...192.168.0.143, Dst: 142.250.185.74
...51452, Dst Port: 443
> Data (39 bytes)
```

Figure 3. Tools for analyzer

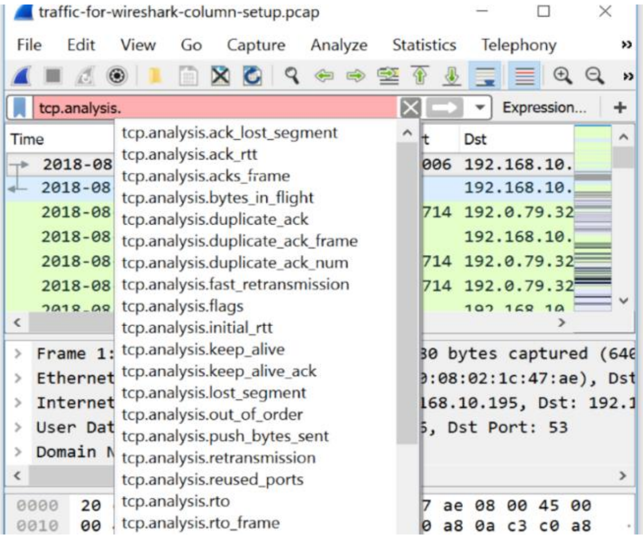
One is used when Wireshark captures packets, while the other is used when it displays packets. Show filters allow the administrator to focal point on the packets that attention them though obscuring the ones that aren't of interest right now. Packets maybe selected according to the protocol, the existence of a field, its value, and a comparison of fields. Right above the column display part of Wireshark is a strip that filters the show. Here you can enter expressions for filtering the frames, internet protocol (IP) packets, or transmission control protocol (TCP) that Wireshark displays from a packet capture (PCAP) (Figure 4). You can also select this to give you far more in-depth definitions.



```
Time                 Src             Port   Dst             Port   Host                 Server Name
2018-08-03 19:06:20  192.168.10.195  62006  192.168.10.1    53
2018-08-03 19:06:20  192.168.10.1    53     192.168.10.195  62006
2018-08-03 19:06:20  192.168.10.195  49714  192.0.79.32     80
2018-08-03 19:06:20  192.0.79.32     80     192.168.10.195  49714
2018-08-03 19:06:20  192.168.10.195  49714  192.0.79.32     80
2018-08-03 19:06:20  192.168.10.195  49714  192.0.79.32     80     college.usatoday.com
2018-08-03 19:06:20  192.0.79.32     80     192.168.10.195  49714
```

Figure 4. Place of the display filter

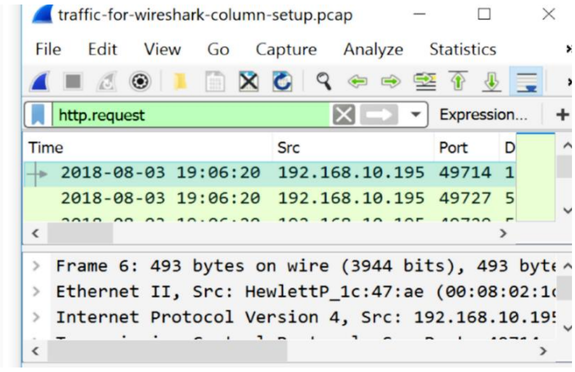
In response to the text you have entered in the display filter, Wireshark provides a roll of offers. As illustrated in picture 5, the demonstration is not yet established while the display filter strip is yet red. If the display filter bar goes green, the expression was recognized and ought to function as intended. The expression has been approved if the display filter bar becomes yellow color, but it probably won't function as planned, as seen in Figure 5. In additional as shown in Figure 6 for more focusing in case of using http protocol. As shown in Figure 7, we can see the work of the Wireshark correctly, and the screen shows the type of the protocol, its length, source, distance, and information.



```
tcp.analysis.

tcp.analysis.ack_lost_segment
tcp.analysis.ack_rtt
tcp.analysis.acks_frame
tcp.analysis.bytes_in_flight
tcp.analysis.duplicate_ack
tcp.analysis.duplicate_ack_frame
tcp.analysis.duplicate_ack_num
tcp.analysis.fast_retransmission
tcp.analysis.flags
tcp.analysis.initial_rtt
tcp.analysis.keep_alive
tcp.analysis.keep_alive_ack
tcp.analysis.lost_segment
tcp.analysis.out_of_order
tcp.analysis.push_bytes_sent
tcp.analysis.retransmission
tcp.analysis.reused_ports
tcp.analysis.rto
tcp.analysis.rto_frame
```

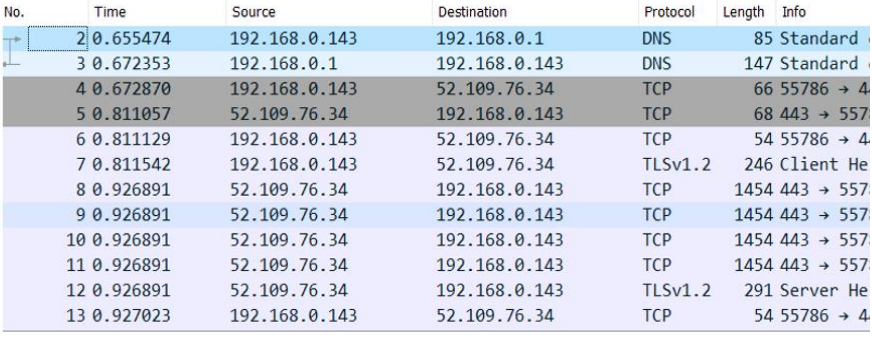
```
2018-08        006 192.168.10.
2018-08            192.168.10.
2018-08        714 192.0.79.32
2018-08            192.168.10.
2018-08        714 192.0.79.32
2018-08        714 192.0.79.32

> Frame 1:     30 bytes captured (646
> Ethernet     :08:02:1c:47:ae), Dst
> Internet     168.10.195, Dst: 192.1
> User Dat     5, Dst Port: 53
> Domain N

0000  20       7 ae 08 00 45 00
0010  00       0 a8 0a c3 c0 a8
```

Figure 5. Display filter in Wireshark gives the suggestion according to what you need



```
http.request

Time                  Src             Port   D
2018-08-03 19:06:20   192.168.10.195  49714  1
2018-08-03 19:06:20   192.168.10.195  49727  5

> Frame 6: 493 bytes on wire (3944 bits), 493 byte
> Ethernet II, Src: HewlettP_1c:47:ae (00:08:02:1c
> Internet Protocol Version 4, Src: 192.168.10.19
```

Figure 6. Display filter accepts a term of expression



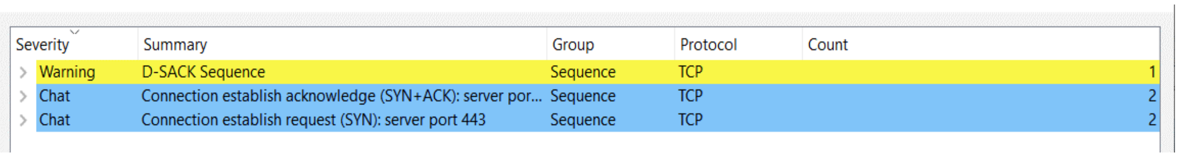
No.	Time	Source	Destination	Protocol	Length	Info
2	0.655474	192.168.0.143	192.168.0.1	DNS	85	Standard
3	0.672353	192.168.0.1	192.168.0.143	DNS	147	Standard
4	0.672870	192.168.0.143	52.109.76.34	TCP	66	55786 → 4
5	0.811057	52.109.76.34	192.168.0.143	TCP	68	443 → 557
6	0.811129	192.168.0.143	52.109.76.34	TCP	54	55786 → 4
7	0.811542	192.168.0.143	52.109.76.34	TLSv1.2	246	Client He
8	0.926891	52.109.76.34	192.168.0.143	TCP	1454	443 → 557
9	0.926891	52.109.76.34	192.168.0.143	TCP	1454	443 → 557
10	0.926891	52.109.76.34	192.168.0.143	TCP	1454	443 → 557
11	0.926891	52.109.76.34	192.168.0.143	TCP	1454	443 → 557
12	0.926891	52.109.76.34	192.168.0.143	TLSv1.2	291	Server He
13	0.927023	192.168.0.143	52.109.76.34	TCP	54	55786 → 4

Figure 7. Sample Screenshot of Wireshark in action

4. POST SNIFFING ANALYSIS

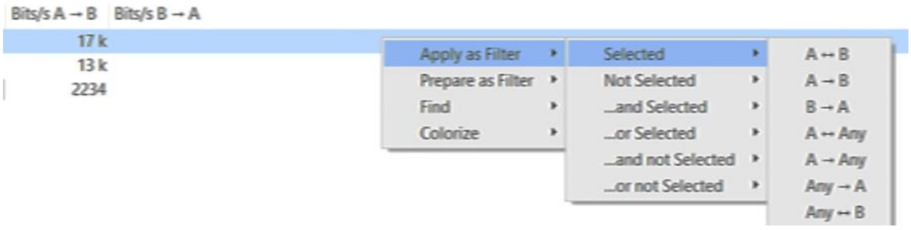
This section examines the display filters, which are the second category of filter. The first one, "filtering while capturing," has already been addressed. Applications for display filters include error detection, packet sniffing, and pattern recognition. It's important to note that, unlike a typical IDS/IPS, Wireshark does not automatically produce alarms and notifications. As an alternative, actions that were taken through a capture can be seen and examined in a while, either by hand or with the aid of other apps. The expert information table (Figure 8) can be used as a tool to support the aforementioned arguments because it clearly denotes checksum mistakes, redundancy checks, and lost segment accounting. In case of taking right click for the TCP adress many filters will apper as shown in Figure 9.

It can attend in on nodes talking as they move packets in two different ways in the specified direction in the captured file. The statistical IO graph is the second important instrument (Figure 10). These graphs can display the overall network traffic flow or just the traffic for a certain set of protocols. Wireshark is one of the easiest to use sniffing software as well as it also has the option of exhibiting distinct post-filtered capture on the graph in different colors to allow simple recognition. Either the system clock or the first packet can be used to set the time. When we merge multiple capture files taken at various times, we may effectively use the system clock time. The timestamp is another statistic tool that merits mentioning in this context; it allows users to time stamp each packet as they see fit. The purpose of the experimentation below is to determine whether the node (server), has ever received an unauthorized packet from an external node, also known as an experimental node, which stands in for a single or a set of bad nodes. We currently use four nodes and each of which represents a potential one or group of nodes in the scenario of a real time as shown in Figure 11.



Severity	Summary	Group	Protocol	Count
Warning	D-SACK Sequence	Sequence	TCP	1
Chat	Connection establish acknowledge (SYN+ACK): server por...	Sequence	TCP	2
Chat	Connection establish request (SYN): server port 443	Sequence	TCP	2

Figure 8. Table of information



Bits/s A → B	Bits/s B → A
17 k	
13 k	
2234	

Figure 9. The tools of discussion

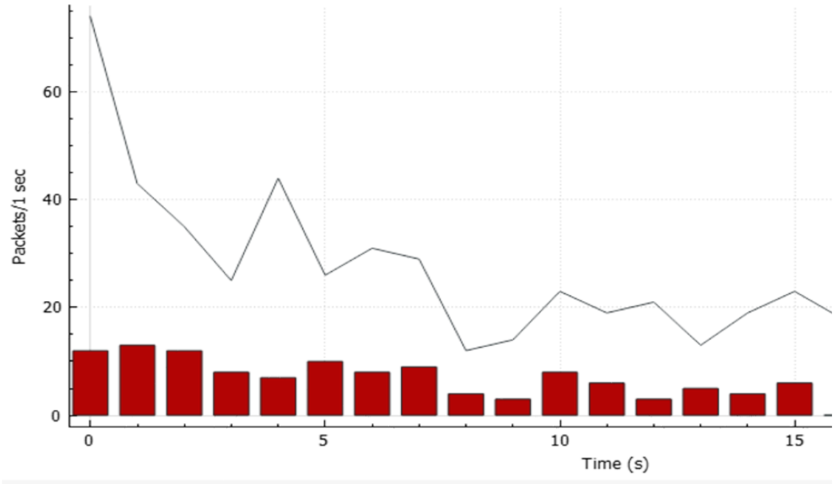


Figure 10. The IO graph tool

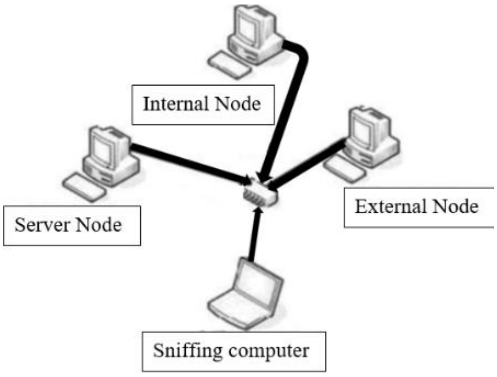
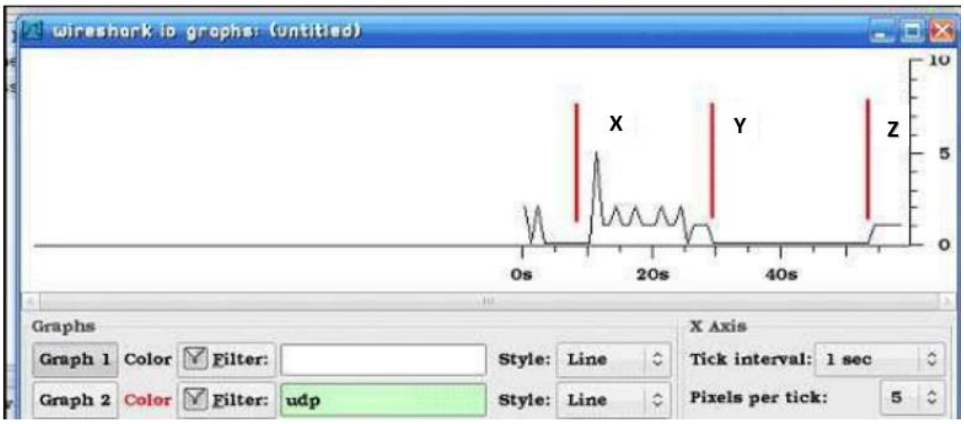


Figure 11. Experimental setup

The region z represented the control traffic flow at the beginning of network and demonstrations no sharp peaks as shown in Figure 12. On the other hand, in the region y we can get the movement in the network between internal and external node and additional to this from the internal node to the server. In additional in the region x, the bad activity begins at this time and go together with UDP action in the packet capture pane and the sharp peaks in the I/O.



Graphs

Graph 1 Color Filter: Style: Line

Graph 2 Color Filter: udp Style: Line

X Axis

Tick interval: 1 sec

Pixels per tick: 5

Figure 12. Graph showing 3 regions

5. CONCLUSION

The aforementioned experiment proves that IDS/IPS devices are necessary in any conventional network. Wireshark's skills for handling and interpreting packet data have also been highlighted. In this experiment, ACL filtering has been the main application of Wireshark. The Wireshark tool offers a wide range of additional filtering options, as well as filtering based on the protocols being used and also according to packet size, and sub-strings. Therefore, Wireshark can transform into global detection software with the right use of filtering commands and other utilities.

REFERENCES

[1] P. R. Upadhyayula, K. Amarendra, and S. B. Kudupudi, “Intrusion detection of imbalanced network traffic,” in *7th International Conference on Communication and Electronics Systems, ICCES 2022 - Proceedings*, Jun. 2022, pp. 840–844, doi: 10.1109/ICCES54183.2022.9835996.

[2] J. S. Raj and A. S. Stephy, “Trust based routing algorithm in internet of things (IoT),” *IRO Journal on Sustainable Wireless Systems*, vol. 01, no. 01, pp. 42–61, Mar. 2019, doi: 10.36548/jsws.2019.1.004.

[3] A. J. A. Al-Gburi *et al.*, “A miniaturised UWB FSS with stop-band characteristics for EM shielding applications,” *Przeglad Elektrotechniczny*, vol. 97, no. 8, pp. 142–145, Aug. 2021, doi: 10.15199/48.2021.08.25.

[4] M. Stolze, R. Pawlitzek, and A. Wespi, “Visual Problem-solving support for new event triage in centralized network security monitoring: challenges, tools and benefits,” *Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI)*, vol. P-39, pp. 67–76, 2003, doi: 10.1109/IMFS54183.2022.9835686.

[5] M. K. Abdulhameed *et al.*, “Novel design of triple-bands EBG,” *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 17, no. 4, p. 1683, Aug. 2019, doi: 10.12928/telkomnika.v17i4.12616.

[6] H. A. Marhoon, R. Alubady, and M. K. Abdulhameed, “Direct line routing protocol to reduce delay for chain based technique in wireless sensor network,” *Karbala International Journal of Modern Science*, vol. 6, no. 2, pp. 190–195, Jun. 2020, doi: 10.33640/2405-609X.1585.

[7] A. M. Dinar, A. S. M. Zain, F. Salehuddin, M. K. Abdulhameed, M. K. Mohsen, and M. L. Attiah, “Impact of gouy-chapman-stern model on conventional ISFET sensitivity and stability,” *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 17, no. 6, pp. 2842–2850, Dec. 2019, doi: 10.12928/TELKOMNIKA.v17i6.12838.

[8] M. K. Abdulhameed, M. S. Kod, and A. J. A. Al-Gburi, “Enhancement of elevation angle for an array leaky-wave antenna,” *Przeglad Elektrotechniczny*, vol. 97, no. 8, pp. 109–113, Aug. 2021, doi: 10.15199/48.2021.08.19.

[9] S. Gaw and E. W. Felten, “Password management strategies for online accounts,” in *Proceedings of the second symposium on Usable privacy and security - SOUPS ’06*, 2006, vol. 149, p. 44, doi: 10.1145/1143120.1143127.

[10] R. Baecker, K. Booth, S. Jovicic, J. McGrenere, and G. Moore, “Reducing the gap between what users know and what they need to know,” in *Proceedings on the 2000 conference on Universal Usability - CUU ’00*, 2000, pp. 17–23, doi: 10.1145/355460.355467.

[11] M. Özalp, C. Karakuzu, and A. Zengin, “Distributed intrusion detection systems: a survey,” *Academic Perspective Procedia*, vol. 2, no. 3, pp. 400–407, Nov. 2019, doi: 10.33793/acperpro.02.03.18.

[12] A. Mosavi, S. Shamshirband, E. Salwana, K. Chau, and J. H. M. Tah, “Prediction of multi-inputs bubble column reactor using a novel hybrid model of computational fluid dynamics and machine learning,” *Engineering Applications of Computational Fluid Mechanics*, vol. 13, no. 1, pp. 482–492, Jan. 2019, doi: 10.1080/19942060.2019.1613448.

[13] V. Palanisamy and R. Thirunavukarasu, “Implications of big data analytics in developing healthcare frameworks – A review,” *Journal of King Saud University - Computer and Information Sciences*, vol. 31, no. 4, pp. 415–425, Oct. 2019, doi: 10.1016/j.jksuci.2017.12.007.

[14] J. Sadowski, “When data is capital: Datafication, accumulation, and extraction,” *Big Data & Society*, vol. 6, no. 1, p. 205395171882054, Jan. 2019, doi: 10.1177/2053951718820549.

[15] J. R. Saura, B. R. Herraez, and A. Reyes-Menendez, “Comparing a traditional approach for financial brand communication analysis with a big data analytics technique,” *IEEE Access*, vol. 7, pp. 37100–37108, 2019, doi: 10.1109/ACCESS.2019.2905301.

[16] M. K. Abdulhameed, M. S. M. Isa, I. M. Ibrahim, M. K. Mohsen, S. R. Hashim, and M. L. Attiah, “Improvement of microstrip antenna performance on thick and high permittivity substrate with electromagnetic band gap,” *Journal of Advanced Research in Dynamical and Control Systems*, vol. 10, no. 4 Special Issue, pp. 661–669, 2018.

[17] S. Schulz, M. Becker, M. R. Groseclose, S. Schadt, and C. Hopf, “Advanced MALDI mass spectrometry imaging in pharmaceutical research and drug development,” *Current Opinion in Biotechnology*, vol. 55, pp. 51–59, Feb. 2019, doi: 10.1016/j.copbio.2018.08.003.

[18] C. Shang and F. You, “Data analytics and machine learning for smart process manufacturing: recent advances and perspectives in the big data era,” *Engineering*, vol. 5, no. 6, pp. 1010–1016, Dec. 2019, doi: 10.1016/j.eng.2019.01.019.

[19] G. S. Ananth, N. Shylashree, S. Tunga, and B. N. Latha, “A novel design for hardware interface board with reduced resource utilization,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 24, no. 3, p. 1414, Dec. 2021, doi: 10.11591/ijeecs.v24.i3.pp1414-1420.

[20] L. T. Heberlein, G. V. Dias, K. N. Levitt, B. Mukherjee, J. Wood, and D. Wolber, “A network security monitor,” in *Proceedings. 1990 IEEE Computer Society Symposium on Research in Security and Privacy*, 1990, pp. 296–304, doi: 10.1109/RISP.1990.63859.

[21] M. K. Abdulhameed, M. S. Mohamad Isa, Z. Zakaria, M. K. Mohsin, and M. L. Attiah, “Mushroom-like EBG to improve patch antenna performance for C-band satellite application,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 5, pp. 3875–3881, Oct. 2018, doi: 10.11591/ijece.v8i5.pp3875-3881.

[22] M. K. Abdulhameed *et al.*, “Enhanced performance of compact 2 × 2 antenna array with electromagnetic band-gap,” *Microwave and Optical Technology Letters*, vol. 62, no. 2, pp. 875–886, Feb. 2020, doi: 10.1002/mop.32092.

[23] G. Morales-Romero *et al.*, “Asynchronous learning: evaluation of virtual classroom metrics according to the perception of university students,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 28, no. 2, pp. 1058–1066, Nov. 2022, doi: 10.11591/ijeecs.v28.i2.pp1058-1066.

[24] M. K. Abdulhameed, S. R. Hashim, N. K. Abdalhameed, and A. J. A. Al-Gburi, “Increasing radiation power in half width microstrip leaky wave antenna by using slots technique,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, p. 392, Feb. 2022, doi: 10.11591/ijece.v12i1.pp392-398.

[25] M. K. Mohsen, M. S. M. Isa, A. A. M. Isa, M. K. Abdulhameed, M. L. Attiah, and A. M. Dinar, “Enhancement of boresight radiation for leaky wave antenna array,” *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 17, no. 5, pp. 2179–2185, Oct. 2019, doi: 10.12928/TELKOMNIKA.v17i5.12631.

BIOGRAPHIES OF AUTHORS

Sarah R. Hashim received the B.Sc. degree in computer science from the University of Karbala in Iraq (2013), the M.Sc. degree. in Network and information technology from Universiti Teknikal Malaysia Melaka (UTeM). Her research interests are on network ip protocol. She is currently engaged as an assistant lecturer at Alzahraa University for Women. She can be contacted at email: sarah.rafil@alzahraa.edu.iq.

Rusul A. Enad she received a B.Sc. in computer science from the University of Karbala in Iraq. She received an M.Sc. in Software from Iran's Imam Reza International University. Her research interests are on detecting DDoS attacks on IoT devices. She is currently engaged as an assistant teacher at Alzahraa University for Women. She can be contacted at email: rusul.ali@alzahraa.edu.iq.

Alyaa M. Al-Khafagi is a M.Sc. degree in an Information technology, specifically in deep learning with image processing from the University of Babylon, Babylon, Iraq. Her research areas are artificial intelligent, deep learning, neural networks, CNN, robotics, image processing, predictions and recognitions, medical image analysis, and biometrics. She is currently taking a position of a responsible for the Ibn Sina division for elearning and assistant teacher at Al-Zahraa University for Women, Karbala, Iraq. She can be contacted at email: alyaa.mahdi@alzahraa.edu.iq.

Noor Kamil Abdalhameed received the B.Sc. degree in electrical engineering from the University of Babylon in Iraq (2007), the M.Sc. degree. in electrical and computer engineering from Florid Institution of Technology (FIT), USA. Her research interests are on network ip protocol. She is currently engaged as an assistant lecturer at Al-Turath University. She can be contacted at email: noor.kamel@turath.edu.iq.