IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

Wireshark as a Tool for Detection of Various LAN Attacks

Haroon Iqbal[1*], Sameena Naaz[2]

[1] Department of Computer Science & Engineering, Jamia Hamdard, New Delhi, India
[2] Department of Computer Science and Engineering, Jamia Hamdard, New Delhi, India

[*]Corresponding Author: haroonshah2764@gmail.com, Tel.: 7006903047

DOI: https://doi.org/10.26438/ijcse/v7i5.833837 | Available online at: www.ijcseonline.org

Accepted: 26/May/2019, Published: 31/May/2019

Abstract— This paper describes the importance of wireshark as a sniffing tool in a computer network. Any throttle in the performance of a network can prove to be a serious concern for network administrators, often leading to huge loss of resources. Many times the cause behind service disruptions like sudden terminal shutdown, connectivity loss, performance degradation etc go undetected because of unawareness about traffic analyzing tools or not knowing exactly why a disruption has occured and is often concluded due to poor network architecture. However, sometimes the cause behind such service disruptions could be due to external attacks which attempt to bring our web server down, send false ARP reply packets or infect our network with malware to form a part of a botnet. The first step towards taking proper action in all these cases is to determine the source of the attack and here, wireshark can be used to monitor and map network traffic. This paper shows how wireshark can prove to be extremely beneficial in such scenarios and accentuates how various local area network attacks like ARP poisoning,DOS attack,MAC flooding and DNS spoofing can be detected using wireshark and also provides some mitigation techniques for these attacks.

Keywords— *Wireshark, LAN Attacks, Packet Sniffers, TCP/IP, Switch, Hub, Server*

I. INTRODUCTION

Many IDS (Intrusion Detection Systems) like AIDE (Advanced Intrusion Detection Environment) monitor network traffic for unusual and suspicious behaviour and can send alerts to the administrator. Even some IDS can take actions based on the rules when a malicious activity is detected. But these come with their own cost and might not prove cost effective for many organisations.

A solution to the above problem is to use packet capturing tools like wireshark. It allows a network administrator to scrutinize all the traffic going through the network. Each packet captured can be examined deeply and information based on its traversal through different layers of the OSI (Open System Interconnection) model can be extracted. It is also a free and open source software[1].

The remaining organisation of the paper is as follows: In part III, the solution for where and how all the network data can to be caputed has been throughly explained. In part IV, some common local area network attacks have been discussed along with their proposed mitigation techniques. Expiremental methodology has been shown in part V, where all the individual attacks have been detected using wiresharkand finally part VI concludes the demonstration.

II. ABOUT WIRESHARK

Previously known with the name of Ethereal. It was developed by Gerald Combs in 1988[1]. It is primarily used to troubleshoot and analyze computer networks. It can be run on both windows and UNIX machines. It comes preinstalled with some Linux distributions like Kali Linux. It supports a different wide range of protocols. It supports both the command-line and graphical user interface. It gives the microscopic details of what is happening on a network and is also a standard across many educational institutions, commercial and non-profit organizations.

III. DETERMINING APPROPRIATE PLACES TO CAPTURE THE NETWORK TRAFFIC

This involves finding an appropriate place to analyze the data flow and wireshark is used for carrying out this analysis. Consider a scenario where a network whose performance has degraded, needs to be troubleshot. The network consists of a number of switches, some workstations and a server. The first step to begin with is to decide where to install wireshark. Installing wireshark directly on the server seems quite convincing to analyze all the traffic that moves across the server. Well it makes sense, however there could be certain situations where physical access to the server is not possible, may be due to some security reasons. So it is not possible to install wireshark on the server itself. Below are some methods which can serve as solutions to the above problem.

1. Using a Hub

A hub is a device that broadcasts data to every node except the node on which it receives the data[2]. If wireshark is installed on a node which is directly connected to a switch port, then only that traffic can be analyzed which comes and goes through that switch port. This is because each port of a switch serves as a separate collision domain [2]. On the contrary, employing a hub connected to the same network segment of the server can solve the problem. By installing wireshark on a node connected to the hub, all the traffic on the network can now be captured. The illustration of which has been shown in Figure 1.

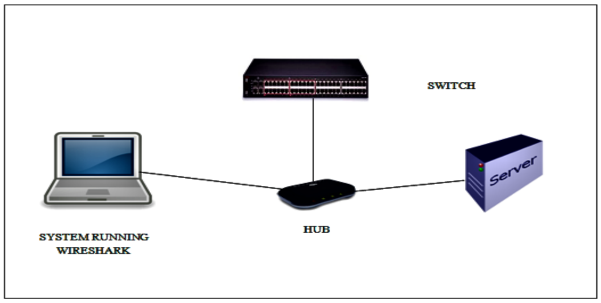


Fig 1: A hub connectivity setup to capture traffic

2. SPAN (Switched Port Analyzer)

Packets seen on one or more switch ports can be copied and directed towards a particular switch port. A node with wireshark installed can be connected to this port in order to analyze the traffic. This method is very convenient if the switch can be accessed directly and is also called port mirroring. One thing to keep in mind is that the port chosen for mirroring needs to be fast enough so that there is not any traffic loss. This scenario has been shown in Figure 2.

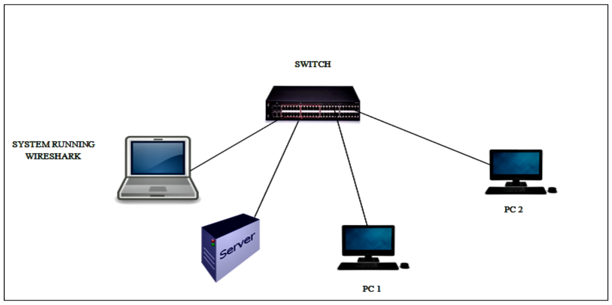
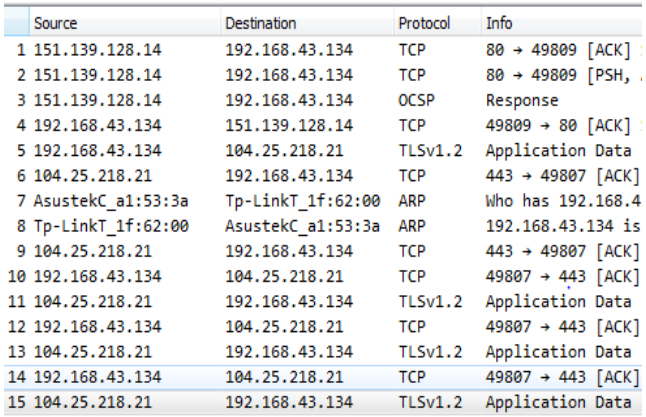


Fig 2: Port mirroring connection setup

3. Remote Packet Capture

This method enables a user to execute a packet capture program on a system. It can sniff all the packets on the target machine and can forward them to another machine. Listening ports, authorized client lists and other options need to be configured properly on the target device. The host machine towards which all the traffic from target device will be directed must be installed with wireshark in order to examine the packets thoroughly. The screenshot showing how remote packet capture looks like in wireshark has been shown in Figure 3.



	Source	Destination	Protocol	Info
1	151.139.128.14	192.168.43.134	TCP	80 → 49809 [ACK]
2	151.139.128.14	192.168.43.134	TCP	80 → 49809 [PSH,
3	151.139.128.14	192.168.43.134	OCSP	Response
4	192.168.43.134	151.139.128.14	TCP	49809 → 80 [ACK]
5	192.168.43.134	104.25.218.21	TLSv1.2	Application Data
6	104.25.218.21	192.168.43.134	TCP	443 → 49807 [ACK]
7	AsustekC_a1:53:3a	Tp-LinkT_1f:62:00	ARP	Who has 192.168.4
8	Tp-LinkT_1f:62:00	AsustekC_a1:53:3a	ARP	192.168.43.134 is
9	104.25.218.21	192.168.43.134	TCP	443 → 49807 [ACK]
10	192.168.43.134	104.25.218.21	TCP	49807 → 443 [ACK]
11	104.25.218.21	192.168.43.134	TLSv1.2	Application Data
12	192.168.43.134	104.25.218.21	TCP	49807 → 443 [ACK]
13	104.25.218.21	192.168.43.134	TLSv1.2	Application Data
14	192.168.43.134	104.25.218.21	TCP	49807 → 443 [ACK]
15	104.25.218.21	192.168.43.134	TLSv1.2	Application Data

Fig 3: Wireshark remote data capture

IV. Common Local Area Network Attacks

Since LAN communications do not remain limited within a confined space, they need to communicate with the outside world. This makes it vulnerable to various forms of attacks. Many of these attacks can be launched from outside the LAN while others can be carried out within the network itself. So there is a need for network administrators to detect and identify the source of these attacks and take action accordingly. Some of these attacks have been discussed below.

1. ARP Poisoning Attack

Address Resolution Protocol poisoning also called ARP spoofing is an attack that can be used to infect communications across the network. ARP resolves IP addresses into MAC addresses. In ARP spoofing, an attacker sends out fake ARP replies to its target machines in an attempt to associate its own MAC address to their IP addresses. With this done, false information gets filled in the ARP caches of the target machines. As a result if a machine sends a message to the other device, the message gets directly to the attacker who can view it, modify it and even forward it to its real destination without being noticed[3].

Mitigation Technique: We can manage to have static ARP entries in our network. Every machine on our network needs to be manually fed with an ARP entry. Then only the predefined list of IP-MAC associations is allowed to communicate in the network and all other ARP replies will be ignored. Although this involves a lot of overhead but is a great way of prevention against ARP spoofing[4]. Similarly Virtual Private Networks also prevent against ARP spoofing attacks as all the communication is encrypted via a tunnel which keeps hackers at bay.

2. MAC Flooding Attack

The aim behind this attack is to fill the content addressable memory (CAM) with numerous invalid source MAC addresses in order to saturate the CAM table so there is no space left for valid entries. CAM refers to the internal memory. CAM refers to the internal memory of switch where port numbers are assigned to corresponding MAC addresses. Whenever a packet arrives at a port, the switch adds in its CAM table the port and MAC address of the machine who sent the packet. If the destination MAC address is not known, then the switch simply copies and broadcasts the packet to every other port on the switch. All the nodes reject the packet except the one for whom the packet is destined. The switch then adds this new MAC-port association in its CAM table. So in future if a packet is destined to the same node, it will be directly sent to the port rather than being broadcasted. In MAC flooding attack the attacker sends traffic on one of the switch ports and each time a packet is sent to the port it contains a new MAC address. In this way the CAM table gets filled with invalid MAC addresses and when it gets saturated, it begins to remove the valid entries. In this way, the entire table becomes infected with malicious entries[5].

Mitigation Technique*:* Port security is a mechanism which can greatly minimize port flooding attacks. Implementing port security puts a limit on the number of MAC addresses a port can learn. If we limit it to a number say 3, it means that the MAC address of the port can be changed only three times. If an attacker then tries to flood the port with many false MAC addresses, the port will simply go into shutdown state. As a result the attacker can no longer fill the CAM table with invalid entries which can highly minimize such attacks.

3. DOS Attacks

Denial of Service (DOS) attack is a type of cyber attack in which the attacker seeks to temporarily or permanently disrupt the services provided by a server so as to make the resources unavailable to its intended users. It is done by sending superfluous requests to the server in an attempt to saturate the server of its request serving capacity. In a DDOS attack, the target machine is flooded with traffic that does not originate from a single source but from many different sources. So it becomes practically impossible to stop the attack simply by blocking a particular source[6]. Although these attacks do not cause any theft or breach of confidential information but the target gets devoid of a great deal of time and money especially when these attacks are carried out on large web servers which provide services to government, banking, commerce or trade organizations.

Mitigation Technique: Contacting the internet service provider (ISP) seems to be the most appropriate action if you find this attack happening on your network. ISPs can determine if the traffic can be rerouted and considering services which can dissipate the huge DDOS traffic among a network of servers is a good idea[7].

Also routers and firewalls must be configured to reject bogus packets. So it is important to keep the routers and firewalls updated with latest security patches.

4. DHCP Spoofing Attack

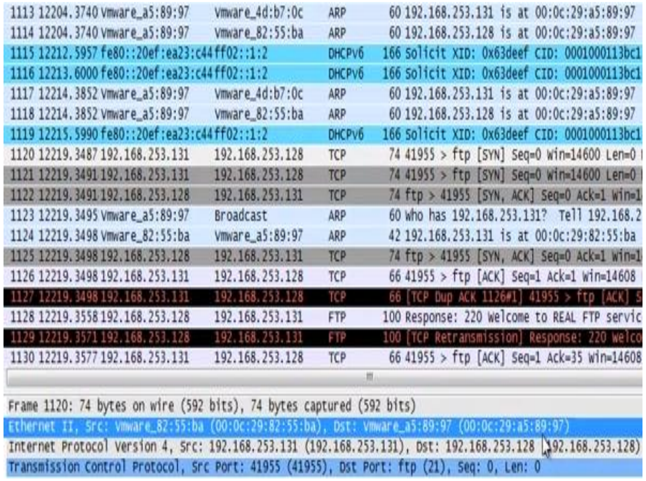
Dynamic Host Configuration Protocol (DHCP) is responsible for giving IP address, default gateway, subnet mask and even DNS information to its DHCP clients. DHCP does not use authentication mechanism that would allow it to verify that the packets are coming from a genuine DHCP server[8]. This vulnerability of DHCP can be exploited and used to carry out an attack called DHCP spoofing. When a client makes a DHCP request, it goes to every device on the network but only the DHCP server knows the actual meaning of the request who then sends a unicast reply to the client giving it all its IP information. In DHCP spoofing a rogue DHCP server replies to the DHCP client before the actual server. As a result the client gets wrong DHCP information which includes the default gateway address[9]. With this done the attacker now redirects all the traffic of the infected device through his machine without the client knowing that its traffic is being sniffed. The attacker can then forward the traffic to its actual destination so as to make client communication possible.

Mitigation Technique: DHCP snooping is a technique that can be used to prevent DHCP spoofing attack. This mechanism configures switch ports into two different categories viz. the trusted and the untrusted category. Only those ports which are configured as trusted can receive DHCP responses. If an attacker attempts to send DHCP response to an untrusted port, the port will simply be disabled. Further if a port has not been set as either trusted or untrusted, it is considered as untrusted by default and can’t except any DHCP responses[10].

V. EXPERIMENTAL METHODOLOGY

1. ARP Spoofing Attack Detection

Consider a scenario in which we have two systems A and B who want to communicate with each other. The IP and MAC addresses of system A are 192.168.253.128 and 00.0C:29:4D:B7:0C respectively. IP and MAC addresses of B are 192.168.253.133 and 00:0C:29:B2:55:BA respectively. Suppose an attacker with IP and MAC address of 192.168.253.132 and 00:0C:29:A5:89:97 launches an ARP poisoning attack to sniff the communication taking place between A and B. When the communication is observed with wireshark, we notice that a single MAC address is associated with two different IP addresses. We also see the a lot of ARP reply packets being generated from the attacker machine providing its own MAC address to both A and B. Screenshot shown in Fig 4 shows how the attack looks when captured in wireshark.



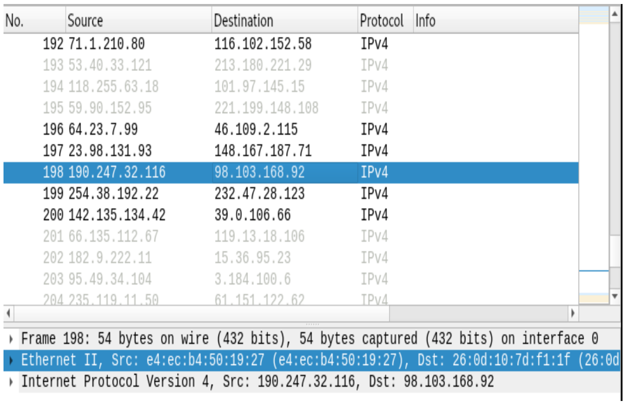
```
1113 12204.3740 Vmware_a5:89:97      Vmware_4d:b7:0c   ARP      60 192.168.253.131 is at 00:0c:29:a5:89:97
1114 12204.3740 Vmware_a5:89:97      Vmware_82:55:ba   ARP      60 192.168.253.128 is at 00:0c:29:a5:89:97
1115 12212.5957 fe80::20ef:ea23:c44 ff02::1:2         DHCPv6  166 Solicit XID: 0x63deef CID: 0001000113bc1
1116 12213.6000 fe80::20ef:ea23:c44 ff02::1:2         DHCPv6  166 Solicit XID: 0x63deef CID: 0001000113bc1
1117 12214.3852 Vmware_a5:89:97      Vmware_4d:b7:0c   ARP      60 192.168.253.131 is at 00:0c:29:a5:89:97
1118 12214.3852 Vmware_a5:89:97      Vmware_82:55:ba   ARP      60 192.168.253.128 is at 00:0c:29:a5:89:97
1119 12215.5990 fe80::20ef:ea23:c44 ff02::1:2         DHCPv6  166 Solicit XID: 0x63deef CID: 0001000113bc1
1120 12219.3487 192.168.253.131      192.168.253.128   TCP      74 41955 > ftp [SYN] Seq=0 Win=14600 Len=0
1121 12219.3491 192.168.253.131      192.168.253.128   TCP      74 41955 > ftp [SYN] Seq=0 Win=14600 Len=0
1122 12219.3491 192.168.253.128      192.168.253.131   TCP      74 ftp > 41955 [SYN, ACK] Seq=0 Ack=1 Win=1
1123 12219.3495 Vmware_a5:89:97      Broadcast         ARP      60 Who has 192.168.253.131?  Tell 192.168.2
1124 12219.3498 Vmware_82:55:ba      Vmware_a5:89:97   ARP      42 192.168.253.131 is at 00:0c:29:82:55:ba
1125 12219.3498 192.168.253.128      192.168.253.131   TCP      74 ftp > 41955 [SYN, ACK] Seq=0 Ack=1 Win=1
1126 12219.3498 192.168.253.131      192.168.253.128   TCP      66 41955 > ftp [ACK] Seq=1 Ack=1 Win=14608
1127 12219.3498 192.168.253.131      192.168.253.128   TCP      66 [TCP Dup ACK 1126#1] 41955 > ftp [ACK] S
1128 12219.3558 192.168.253.128      192.168.253.131   FTP     100 Response: 220 Welcome to REAL FTP servic
1129 12219.3571 192.168.253.128      192.168.253.131   FTP     100 [TCP Retransmission] Response: 220 Welco
1130 12219.3577 192.168.253.131      192.168.253.128   TCP      66 41955 > ftp [ACK] Seq=1 Ack=35 Win=14608

Frame 1120: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
Ethernet II, Src: Vmware_82:55:ba (00:0c:29:82:55:ba), Dst: Vmware_a5:89:97 (00:0c:29:a5:89:97)
Internet Protocol Version 4, Src: 192.168.253.131 (192.168.253.131), Dst: 192.168.253.128 (192.168.253.128)
Transmission Control Protocol, Src Port: 41955 (41955), Dst Port: ftp (21), Seq: 0, Len: 0
```

Fig 4: ARP spoofing as captured in Wireshark

2. MAC Flooding Attack Detection

Wireshark is allowed to capture the traffic and analyzed properly. If observed carefully we can see that there are hundreds of different physical addresses. We notice huge number of different physical addresses that easily outnumber the total devices in our network. From here we can recognize that this is rather a MAC flooding attack. Screenshot shown in Fig 5 shows how the attack looks like when captured using wireshark. Every source and destination IP address shown in the figure has its own MAC address which can be seen in the packet details section of wireshark.



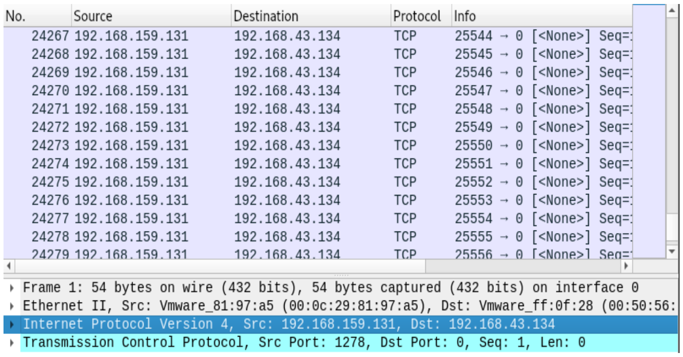
```
No.  Source           Destination       Protocol  Info
192  71.1.210.80      116.102.152.58    IPv4
193  53.40.33.121     213.180.221.29    IPv4
194  118.255.63.18    101.97.145.15     IPv4
195  59.90.152.95     221.199.148.108   IPv4
196  64.23.7.99       46.109.2.115      IPv4
197  23.98.131.93     148.167.187.71    IPv4
198  190.247.32.116   98.103.168.92     IPv4
199  254.38.192.22    232.47.28.123     IPv4
200  142.135.134.42   39.0.106.66       IPv4
201  66.135.112.67    119.13.18.106     IPv4
202  182.9.222.11     15.36.95.23       IPv4
203  95.49.34.104     3.184.100.6       IPv4
204  235.119.11.50    61.151.122.62     IPv4

Frame 198: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
Ethernet II, Src: e4:ec:b4:50:19:27 (e4:ec:b4:50:19:27), Dst: 26:0d:10:7d:f1:1f (26:0d
Internet Protocol Version 4, Src: 190.247.32.116, Dst: 98.103.168.92
```

Fig 5: MAC flooding attack captured using wireshark

3. DOS Attack Detection

Suppose wireshark is allowed to capture the traffic on the network. When packets are observed closely, it is found that a huge amount of TCP connections with SYN flag activated are generated from a single source IP address. These TCP SYN requests do not receive any acknowledgement from the server. It can be noticed that in a very short period of time, large number of connection attempts are made by a device to the target machine which receive no acknowledgement from the server, thus making it practically impossible to complete the three way handshake. During the wait time, when the server waits for acknowledgement to come, more packets keep coming which trigger new connections[11]. Fig 6 shows how DOS attack looks like when captured using wireshark.



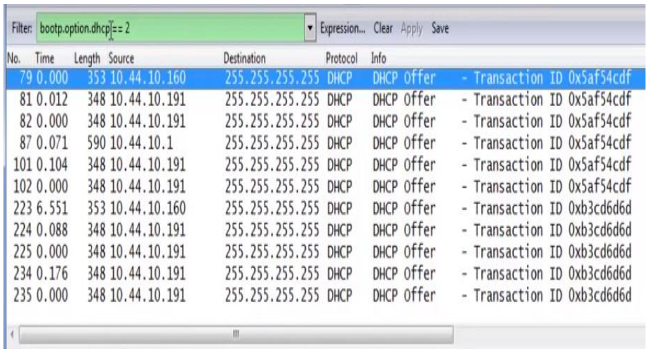
```
No.    Source           Destination     Protocol  Info
24267  192.168.159.131  192.168.43.134  TCP       25544 → 0 [<None>] Seq=
24268  192.168.159.131  192.168.43.134  TCP       25545 → 0 [<None>] Seq=
24269  192.168.159.131  192.168.43.134  TCP       25546 → 0 [<None>] Seq=
24270  192.168.159.131  192.168.43.134  TCP       25547 → 0 [<None>] Seq=
24271  192.168.159.131  192.168.43.134  TCP       25548 → 0 [<None>] Seq=
24272  192.168.159.131  192.168.43.134  TCP       25549 → 0 [<None>] Seq=
24273  192.168.159.131  192.168.43.134  TCP       25550 → 0 [<None>] Seq=
24274  192.168.159.131  192.168.43.134  TCP       25551 → 0 [<None>] Seq=
24275  192.168.159.131  192.168.43.134  TCP       25552 → 0 [<None>] Seq=
24276  192.168.159.131  192.168.43.134  TCP       25553 → 0 [<None>] Seq=
24277  192.168.159.131  192.168.43.134  TCP       25554 → 0 [<None>] Seq=
24278  192.168.159.131  192.168.43.134  TCP       25555 → 0 [<None>] Seq=
24279  192.168.159.131  192.168.43.134  TCP       25556 → 0 [<None>] Seq=

Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
Ethernet II, Src: Vmware_81:97:a5 (00:0c:29:81:97:a5), Dst: Vmware_ff:0f:28 (00:50:56:
Internet Protocol Version 4, Src: 192.168.159.131, Dst: 192.168.43.134
Transmission Control Protocol, Src Port: 1278, Dst Port: 0, Seq: 1, Len: 0
```

Fig 6: DOS attack data flow capture

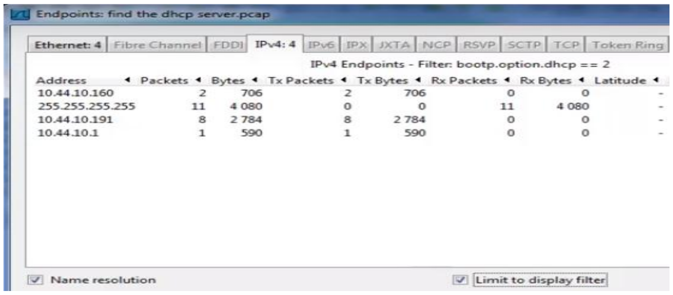
4. DHCP Spoofing Attack Detection

Consider a situation in which a network administrator receives a complaint that some people in the network are not able to access the network resources. The first step is to execute wireshark on a system to begin packet capture, then release and renew the IP address of the system through “iprelease” and “iprenew” commands respectively. We now filter the packets to only view DHCP offer packets as depicted in Fig 7, since these are from server [12]. Under the statistics tab, we go to endpoints and check the "limit to display filter" checkbox. Doing this, will show all DHCP server IP addresses. The address of the genuine DHCP server is actually the destination IP address of the release packet when we had released our actual IP configuration. So, the rest of DHCP servers which have different IP addresses as shown in Fig 8 are rogue, which are responsible for carrying out DHCP spoofing.


Filter: bootp.option.dhcp == 2
Expression... Clear Apply Save

```
No. Time   Length Source        Destination      Protocol Info
 79 0.000  353 10.44.10.160     255.255.255.255 DHCP     DHCP Offer   - Transaction ID 0x5af54cdf
 81 0.012  348 10.44.10.191     255.255.255.255 DHCP     DHCP Offer   - Transaction ID 0x5af54cdf
 82 0.000  348 10.44.10.191     255.255.255.255 DHCP     DHCP Offer   - Transaction ID 0x5af54cdf
 87 0.071  590 10.44.10.1       255.255.255.255 DHCP     DHCP Offer   - Transaction ID 0x5af54cdf
101 0.104  348 10.44.10.191     255.255.255.255 DHCP     DHCP Offer   - Transaction ID 0x5af54cdf
102 0.000  348 10.44.10.191     255.255.255.255 DHCP     DHCP Offer   - Transaction ID 0x5af54cdf
223 6.551  353 10.44.10.160     255.255.255.255 DHCP     DHCP Offer   - Transaction ID 0xb3cd6d6d
224 0.088  348 10.44.10.191     255.255.255.255 DHCP     DHCP Offer   - Transaction ID 0xb3cd6d6d
225 0.000  348 10.44.10.191     255.255.255.255 DHCP     DHCP Offer   - Transaction ID 0xb3cd6d6d
234 0.176  348 10.44.10.191     255.255.255.255 DHCP     DHCP Offer   - Transaction ID 0xb3cd6d6d
235 0.000  348 10.44.10.191     255.255.255.255 DHCP     DHCP Offer   - Transaction ID 0xb3cd6d6d
```

Fig 7: DHCP offer packets



```
IPv4 Endpoints - Filter: bootp.option.dhcp == 2
Address          Packets  Bytes  Tx Packets  Tx Bytes  Rx Packets  Rx Bytes  Latitude
10.44.10.160           2    706           2       706           0         0         -
255.255.255.255       11  4 080           0         0          11     4 080         -
10.44.10.191           8  2 784           8     2 784           0         0         -
10.44.10.1             1    590           1       590           0         0         -
```

☑ Name resolution ☑ Limit to display filter

Fig 8: DHCP servers including Rogue servers

VI. CONCLUSION

Wireshark is the world's main and mostly used network protocol analyzer. Network administrators profusely use this tool to troubleshoot network problems. Proper ways of capturing and analyzing network traffic using wireshark have been discussed. Also, an insight into some of the common cyber attacks; how they can be detected using wireshark along with proposed mitigation techniques have been provided. From the above findings we conclude that wireshark is actually an excellent tool for monitoring network traffic.

However, it is not possible for wireshark to warn users beforehand that an attack is likely to happen. If it is well known in advance that a bug is likely to affect the network, it will be a boon to the IT industry and a massive prevention against loss of resources. Network analysts need to investigate and develop methods which would enable wireshark to predict the flow of data streams.

REFERENCES

[1] C. Sanders, *Practical Packet Analysis With Wireshark*. .

[2] S. Mishra, L. Jena, and A. Pradhan, "Networking Devices and Topologies: A Succinct Study," 2012.

[3] S. Hijazi and M. S. Obaidat, "Address resolution protocol spoofing attacks and security approaches: A survey," *Secur. Priv.*, p. e49, Dec. 2018.

[4] D. Bruschi, A. Ornaghi, and E. Rosti, "S-ARP: a secure address resolution protocol," in *19th Annual Computer Security Applications Conference, 2003. Proceedings.*, pp. 66–74.

[5] M. Hamedi, *Insider Attack and Cyber Security*, vol. 39, no. 2. Boston, MA: Springer US, 2008.

[6] S. Pavithirakini, D. D. M. M. Bandara, C. N. Gunawardhana, K. K. S. Perera, B. G. M. M. Abeyrathne, and D. Dhammearatchi, "Improve the Capabilities of Wireshark as a tool for Intrusion Detection in DOS Attacks," *Int. J. Sci. Res. Publ.*, vol. 6, no. 4, p. 378, 2016.

[7] "Denial of service (DoS) attack prevention through random access channel resource reallocation," Dec. 2010.

[8] R. Droms, "Dynamic Host Configuration Protocol," Mar. 1997.

[9] X. Gu and R. Hunt, "Wireless LAN Attacks and Vulnerabilities," *Networks and Communication Systems*. ACTA Press.

[10] L. Senecal, "Understanding and preventing attacks at layer 2 of the OSI reference model," in *4th Annual Communication Networks and Services Research Conference (CNSR'06)*, 2006, p. 1 pp.

[11] J. Biswas and A. Ashutosh, "An Insight in to Network Traffic Analysis using Packet Sniffer," *Int. J. Comput. Appl.*, vol. 94, no. 11, pp. 39–44, 2014.

[12] S. Naaz and F. A. Badroo, "Investigating DHCP and DNS Protocols Using Wireshark Investigating DHCP and DNS Protocols Using Wireshark," no. May 2017, pp. 0–8, 2016.

Authors Profile

Mr. Haroon Iqbal pursed Bachelor of Technology from Jamia Hamdard (Deemed to be University), New Delhi in 2017 and Master of Technology from Jamia Hamdard (Deemed to be University) in the year 2019.

Dr. Sameena Naaz pursed B.Sc Engg. (Computers) from Aligarh Muslim University in 1998, Master of Technology in Communication and Information System) from Aligarh Muslim University in 2000, and Ph.D (Computer Science) from Jamia Hamdard, New Delhi in 2014. She is currently working as Assistant Professor in Department of Computer Science and Engineering, Jamia Hamdard, New Delhi since 2004. His main research work focuses on Fuzzy Based Algorithms for Load Balancing in a Distributed Environment. She has 16+ years of experience in the field of teaching, learning and research.