International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

A Novel Algorithm to detect Replay Attack in WLANs

Rajinder Singh, Satish Kumar

Abstract*: Nowadays, security of wireless local area networks (WLANs) is the biggest concern for many organizations. Protecting wireless networks from unauthorized users is a major concern to be addressed. The broadcasting nature of the wireless networks makes it vulnerable to carry different types of attacks. Main wireless threats are man in the middle attack, eavesdropping, evil twin attack and replay attack etc. In this paper Replay Attack threat is highlighted. An attacker can use many different tools available freely on the internet to carry this type of attack. We propose a methodology to detect the Replay attack. In this paper a variant of replay attack called address resolution replay attack is discussed. To carry this attack Ettercap tool is used. In this type of attack, attacker floods the wireless network with ARP traffic. So the performance of the wireless network degrades. To detect ARP request/replay attack our proposed method uses protocol field and nature of frame field. A Replay attack in which all the traffic captured and replayed is also discussed. To detect this type of attack, main parameters which are considered are the number of retransmitted packets, number of duplicate ACK and number of TCP spurious retransmitted packet.
Keywords: MITM, RAP, RSSI, RTT

I. INTRODUCTION

A replay attack is a type of attack in which an attacker eavesdrops the data transmission and replays it. Replay attack means intercepting and retransmission of the data. The attacker does not need to decrypt data packets. Even the network is secured by the various security algorithms, this attack is still successful. It is successful by simply replaying the traffic [1]. For example, when an attacker replays a message which was earlier sent by the legitimate user. Messages may be encrypted, but an attacker replaying them can get the access to the network. He can get the access to the resources present on the network. The attacker can steal the sensitive information exchanged between the two communicating parties. Computers which are subjected to replay attack, see the replay packets as legitimate packets [2]. In this paper, we are considering two variants of replay attack. One attack is an ARP request/replay attack. The other attack is conducted by replaying all the packets which were captured earlier. The ARP request replay attack is possible due to ARP protocol vulnerability. ARP protocol is used to map IP address with the corresponding MAC address.

One purpose for which replay attack is used to crack WEP keys. At least 10000 to 15000 data packets are needed to collect the initialization vector. In case of lesser traffic is flowing into the network, the attacker need to sit there for many hours to get the desired information. Another solution is simply replay a number of ARP packets and gathering thousands of packets in a small interval of time. When sufficient numbers of packets are captured, it is very easy to break the encrypted password [3].

A. Types of Replay Attack

Replay attacks can fall into two types of categories. 1) Live Replay Attack 2) Passive Replay Attack. In case of live replay attack, attacker captures the traffic, which is flowing between the victim machine and AP and starts to replay it. In case of a passive attack, attacker captures the traffic and uses it later to find the secret information.

B. Tools used for carrying replay attack

Main tools used to carry the ARP replay attack are:

a) Ettercap

This tool is freely available on the internet. It can run on various UNIX like distributions. It is used for capturing wireless traffic, passwords and for active eavesdropping [4].

b) Cain and Able

This tool is a free password recovery tool and can be used for WEP cracking as well as for wireless packet injection [5].

c) Aircrack-ng

This free tool is a suite of tools which can also be used for packet injection and replay attack [6].

d) Scapy

It is a based on python language can be used to create various types of protocol packets [7].

Main tools which are used to replay all the captured packets are:

a) Colasoft Packet Player

This tool is a free software tool and it allows the users to open the captured file. It also allows the users to replay the captured traffic [8].

b) Tcpreplay

This tool is a free tool for editing and replaying captured files. This tool can be used to create new pcap files from the captured traffic and can replay them [9].

In lab experiments Ettercap, Cain and Able and Colasoft packet player tools are used.

II. PROBLEM STATEMENT (REPLAY ATTACK)

As shown in Fig.1, client machine is connected through routers and servers. Attacker first connects its machine with this network without authorization.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Rajinder Singh*, DCSA, Panjab University SSGRC, Hoshiarpur, Punjab, India. Email: rajinderid@gmail.com
Dr. Satish Kumar, DCSA, Panjab University SSGRC, Hoshiarpur, Punjab, India. Email: satishnotra@yahoo.co.in

After successfully connecting to this network, he chooses the victim machine. Attacker then places itself between the victim machine and router. After successfully placing, all the wireless traffic goes through the attacker machines. Now attacker can catch this traffic and replays it. He can catch all the traffic and replays it or he can replay it later for different purpose. If the attacker replays it, during the time client is connected to the server then it is called live attack. If an attacker replays this traffic later then it is called passive attack.

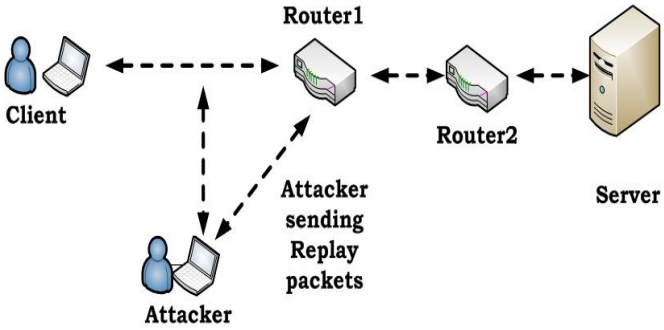

Figure 1. Replay Attack.

III. LITERATURE REVIEW

Replay attack is carried by replaying valid data which is captured earlier. This valid data is sent into the network maliciously or captured data is delayed [10]. It is carried in two steps: First adversary sniffs the wireless network channel for desired packet. In next step he retransmits the data across the network after modifying it or without modifying it. An attacker sends the repeated copies of the packets to the victim. So he can exhaust the energy of the operating devices in the network or even power supply. Attacker can also crash poorly designed applications [11]. According to Authors in paper [1] even if a wireless network is protected by an encryption technique it is still prone to replay attack. Replay attack can also be used by the attacker to misuse the network resources. According to them an attacker can also use this type of threat to launch Denial of Service attack on network. Authors in the paper [12] have discussed that how replay attack can be used to create congestion and interference in the wireless network. According to them performance of the network can be degraded up to 61% by a single attacker. Their suggested method can cope with this attack.

IV. ARCHITECTURE

The test bed for detecting Replay attack is shown in Fig. 2. It consists of a wireless network, an AP, client devices and an attacker machine. A server machine for monitoring and scanning wireless traffic in the network is also used. First attacker gathers information about the wireless network, information about the connected clients and information about the AP with which the client devices are communicating. Now attacker chooses the victim device. After getting the sufficient information, attacker connects itself with the wireless network. After capturing the wireless traffic, which is flowing between the AP and the victim machine, he/she starts replaying captured traffic.

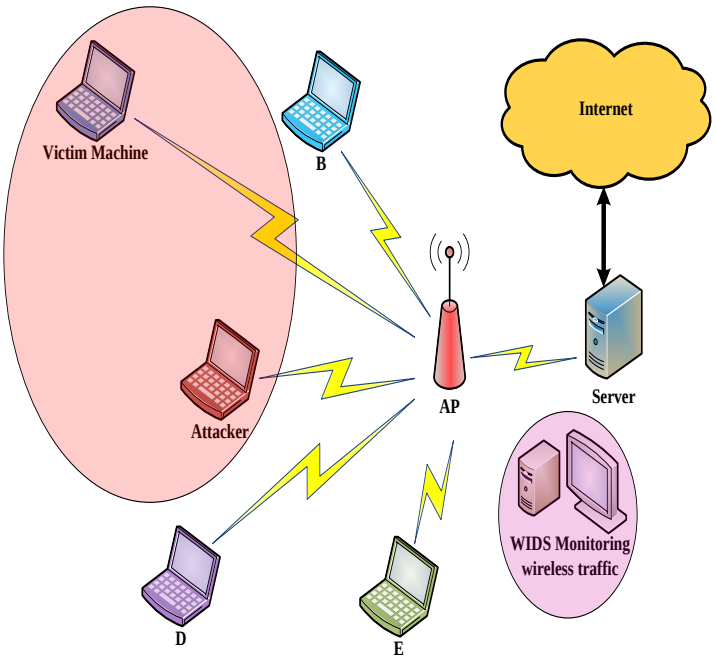

Figure 2. Test Bed for detection of Replay Attack.

V. RESEARCH METHODOLOGY

Here we are considering two types of replay attack. First is an ARP request Replay attack which is conducted by Ettercap tool. The second Replay attack is conducted by capturing all the packets while opening the home page of a site and then replaying all of these packets with the tools available on the Internet.

ARP request replay attack: ARP protocol is used to map MAC address or hardware address of a device to the dynamic IP address allocated to it. Address resolution means to find the dynamic IP address and associated hardware address of the device. Another function of the ARP is to translate 32 bit IP address of a device to its 48 bit hardware address. A network device uses ARP (Address Resolution Protocol) to find the MAC address of another host present on the local network (Figure 3).

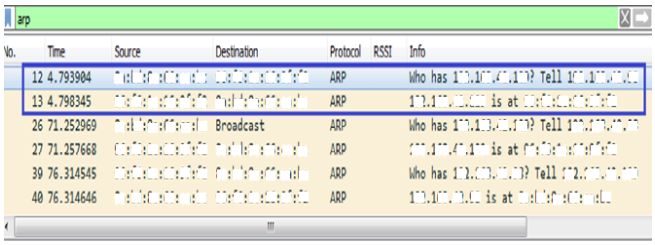

Figure 3. ARP packets shown by Wireshark tool.

When a packet addressed to this computer arrives at the gateway, it uses ARP protocol to find MAC address associated with IP address. All the operating systems maintain a table called ARP cache, which consist of IP addresses and associated MAC addresses. When a device wants to communicate with another device on the same LAN, it checks ARP cache to find IP and MAC address of that machine. If it is there, then no ARP request is sent. If this entry does not exist, then the ARP request is made. It is a broadcast request and sent to all machines on the LAN [13]. When a computer is powered on the network, its IP address is used for the communication. It's a two way process. First this device sends a broadcast packet for an IP address, on the local network. The device which has the requested IP addresses sends a unicast ARP replies to the sender network device.
ARP request uses a unicast source address and broadcast destination address. ARP reply use unicast source address and unicast destination address [14] [15].

A. Design for Detecting for ARP request replay attack

The proposed system detects the ARP request attack in passive mode. First traffic is captured from the victim node. It is observed during the ARP attack by Ettercap tool a lot of ARP packets are injected into the network in a very short period of time. Figure 4 given below shows the traffic captured by the Wireshark tool during the ARP attack.

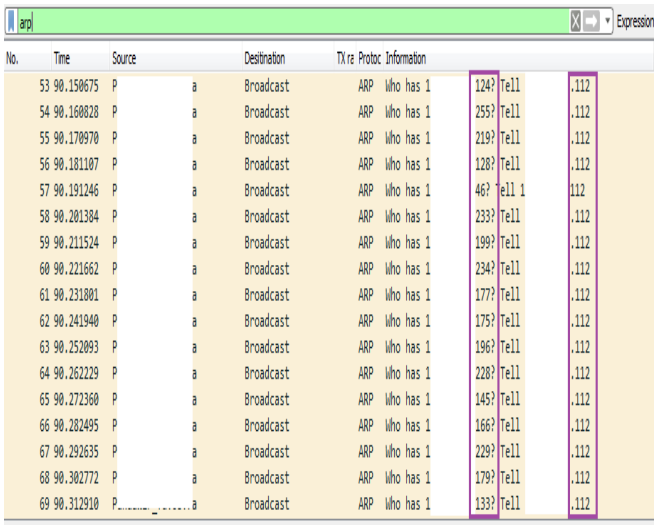

Figure 4. ARP packets

From the picture it is clear that during the attack victim node generates ARP packets in a very short time period and it is sent to all the nodes numbered between 1 to 255. When attacker use Cain and Able tools for ARP attack similar kind of behavior is observed in the network.
So for detecting this type of attack, following three parameters are considered. i) Protocol type ii) Nature of the frame (unicast, multicast or broadcast) iii) Frames are sent to all the nodes within the same network domain. ARP cache table contains entries for MAC address and associated IP address as shown in Figure 5 given below.

Figure 5. ARP cache table.

ARP age factor in case of Windows is calculated as given below: Time = (Base Reachable Time) x (Random value between 0.5 and 1.5).
Figure 6 given below shows the Base Reachable Time in case of Windows.

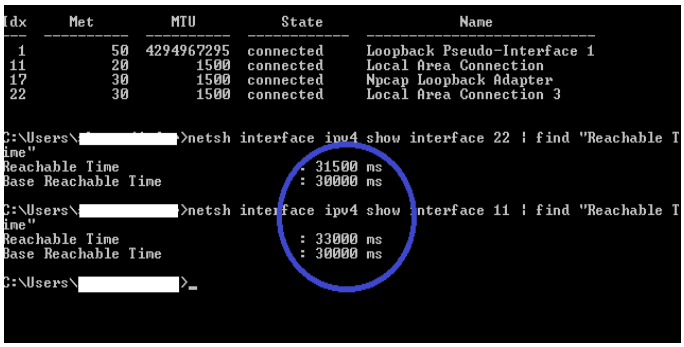

Figure 6. Base Reachable Time.

This entry varies from device to device. So all the nodes present within the network domain will not generate the ARP traffic at the same time and if one of them is generating ARP traffic and packet type is of broadcasting nature, it means it is a victim node [16].

B. Design for detecting replay attack (all the captured traffic replayed)

TCP protocol provides reliable and error checked delivery of the packets. Main common network errors are network collisions, network congestions, checksum errors, or TCP packets which are arriving out of order. Retransmission of the packets means resending those packets which are lost or damaged. This retransmission ratio should be below 2%. In case of our proposed algorithm we are considering number of retransmitted packets, number of duplicate ACK and number of TCP spurious retransmitted packet. These three parameters are considered because they are directly linked with abnormal behavior of the traffic flowing into the network [17].

VI. RESULTS AND DISCUSSIONS

In a normal network operation when there is no ARP request replay attack, the numbers of ARP packets flowing within the network with respect to time are shown in the Figure 7 given below. From the picture it is clear that only a few number of ARP requests are generated under normal operation of the network. In this case total eight ARP packets are generated and sent to the network.

But in case of this attack many ARP packets are generated within a small interval of time as shown in the picture given below. We can check that in case of no attack lesser number of ARP packets are generated and time interval taken for generating these packets and injecting into the network is big.

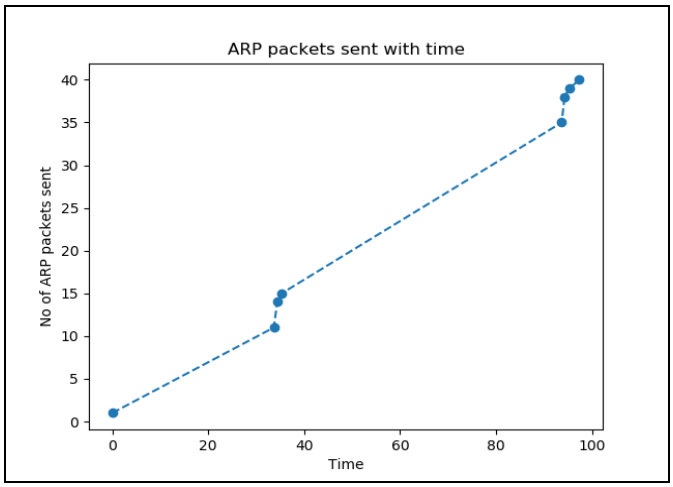

Figure 7. ARP packets during normal network opeartion.

Figure 8 given below shows the ARP packets generated and injected within a very short interval of time.

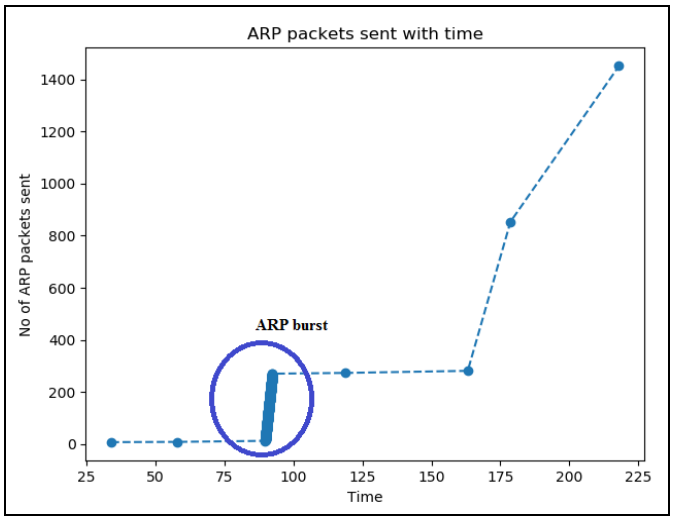

Figure 8. ARP packets during attack.

When we zoom the picture we get the details of the ARP packets and the time when they are injected into the network (Figure 9).

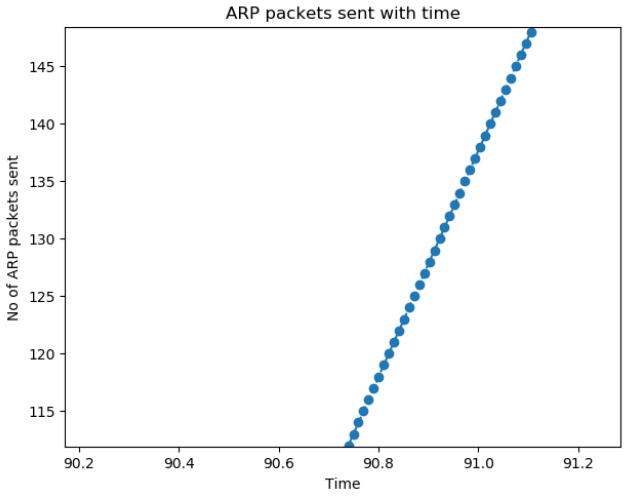

Figure 9.ARP packets within short interval.

In normal scenario 8 ARP packet generated and time taken by these packets to flow into the network is 0.08215078561053. But in case of ARP attack time is 1.4174331966178284. Figure 10 given below shows the fowchart of the proposed method.

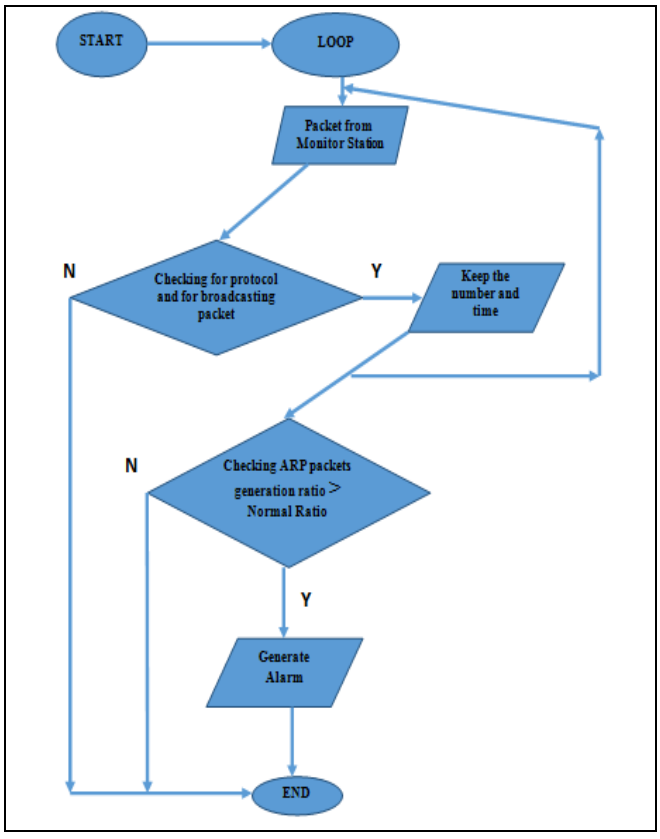

Figure 10. Flowchart for detecting ARP replay attack.

Same kind of behavior was observed for the second node of the same network when it was attacked and we get the following information (Figure 11).

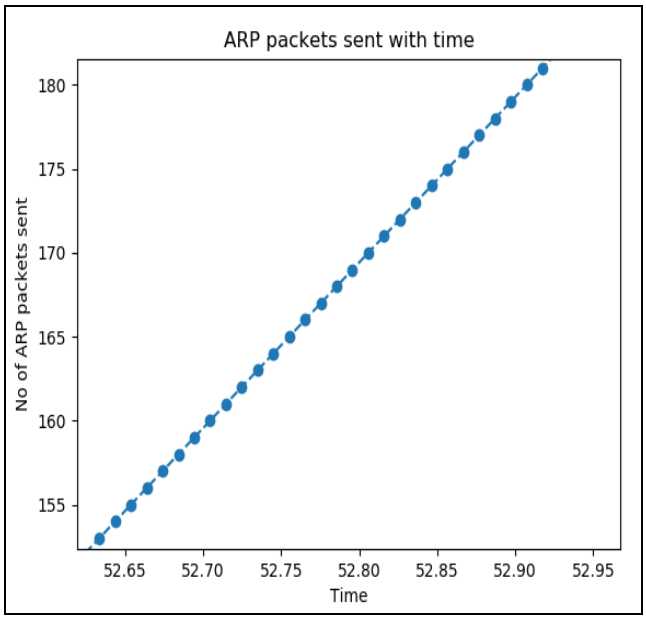

Figure 11. ARP attack on the other node of same network.

Case 2: When all the packets all replayed.

For conducting this type of replay attack , traffic is captured with the help of wireshark while opening the home page of a given site. Avearge number of traffic captured is shown below (Figure 12).

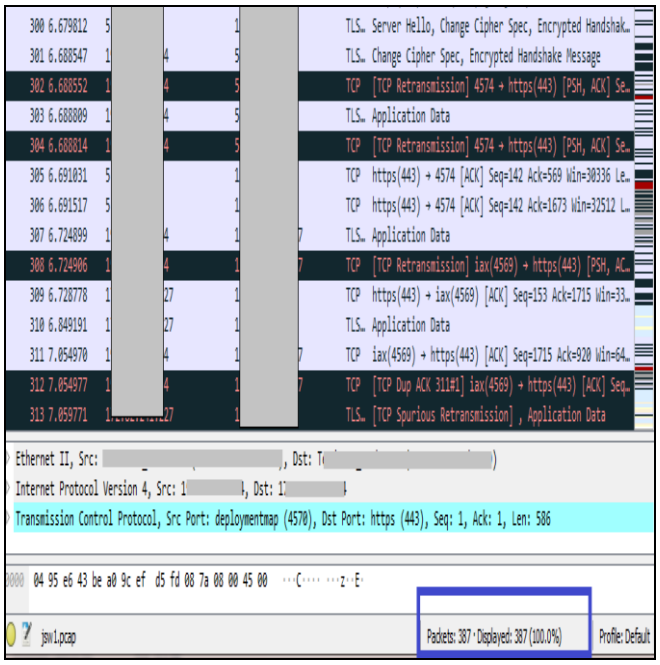

Figure 12. Number of packets captured.

Expert information feature of wireshark gives the following details of the packets of this capture. The user interface consists of following components: a) Severity b) Protocol c) Summary d) Number of packets. From the Figure 13 given below, we can check that main transmission errors which are occurred are retransmission and duplicate ACK.

Severity	Summary	Group	Protocol	Count
Warning	DNS query retransmission. Original request...	Protocol	LLMNR	4
Warning	Connection reset (RST)	Sequence	TCP	8
Warning	This frame is a (suspected) out-of-order se...	Sequence	TCP	20
Note	TCP keep-alive segment	Sequence	TCP	1
Note	This frame is a (suspected) spurious retran...	Sequence	TCP	6
Note	This session reuses previously negotiated k...	Sequence	TLS	5
Note	This frame is a (suspected) retransmission	Sequence	TCP	54
Note	Duplicate ACK (#1)	Sequence	TCP	52
Chat	M-SEARCH * HTTP/1.1\r\n	Sequence	SSDP	11
Chat	Connection finish (FIN)	Sequence	TCP	25
Chat	Connection establish acknowledge (SYN+...	Sequence	TCP	12
Chat	Connection establish request (SYN): server ...	Sequence	TCP	24

Figure 13. Expert information shown by Wireshark.

Now during the attack scenario, home page of this site is opened again and at the same time captured packets are replayed using Colasoft packet player tool. Now on the server side traffic is captured and analyzed with Wireshark.

Severity	Summary	Group	Protocol	Count
Error	New fragment overlaps old data (retransmissi...	Malformed	TCP	1
Warning	Ignored Unknown Record	Protocol	TLS	4
Warning	DNS query retransmission. Original request in...	Protocol	LLMNR	12
Warning	Previous segment(s) not captured (common ...	Sequence	TCP	8
Warning	DNS query retransmission. Original request in...	Protocol	mDNS	2
Warning	Connection reset (RST)	Sequence	TCP	18
Warning	DNS query retransmission. Original request in...	Protocol	DNS	9
Warning	This frame is a (suspected) out-of-order seg...	Sequence	TCP	112
Note	ACK to a TCP keep-alive segment	Sequence	TCP	1
Note	TCP keep-alive segment	Sequence	TCP	4
Note	This frame is a (suspected) fast retransmission	Sequence	TCP	24
Note	A new tcp session is started with the same po...	Sequence	TCP	48
Note	This frame is a (suspected) spurious retransmi...	Sequence	TCP	15
Note	This session reuses previously negotiated key...	Sequence	TLS	5
Note	This frame is a (suspected) retransmission	Sequence	TCP	347
Note	Duplicate ACK (#1)	Sequence	TCP	190
Chat	M-SEARCH * HTTP/1.1\r\n	Sequence	SSDP	30
Chat	Connection finish (FIN)	Sequence	TCP	86
Chat	Connection establish acknowledge (SYN+AC...	Sequence	TCP	96
Chat	Connection establish request (SYN): server no...	Sequence	TCP	72

Figure 14. Transmission errors shown by expert infprmation

From the captured packets it is clear that the traffic is doubled and that the various types of transmission errors are also increased.We can see that retransmitted packets and duplicate ACK number is also increased (Figure 14). The retransmission traffic rate should not be more than 2%. If it is higher it means that network may be affected [18]. Keeping this mind the proposed method is suggested which use the number of retransmitted packets and number of duplicate ACK .

In case of normal network operation(When there is no Replay attack) :

Figure 15 given below shows the graph between the packets generated and time period when these are injected into the network. Only number of retransmitted packets, number of duplicate ACK and number of TCP spurious retransmitted packets are taken.

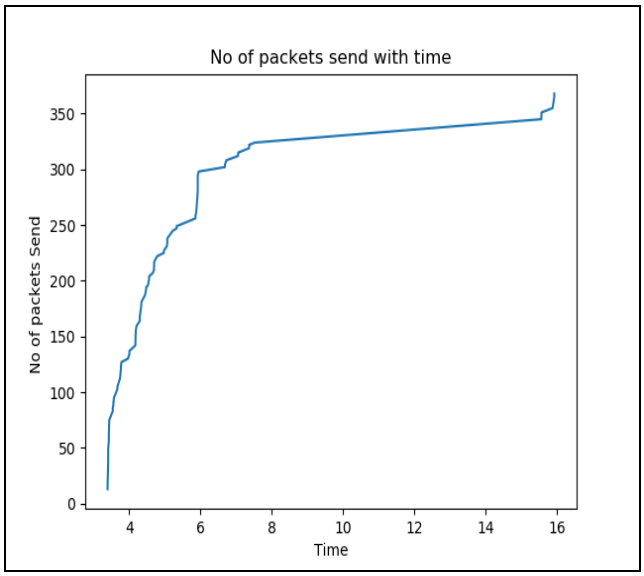

Figure 15. Number of packets injected under normal network condition.

In case of replay attack:

Figure 16 shows the number of different packets are generated and injected into the network are now doubled.

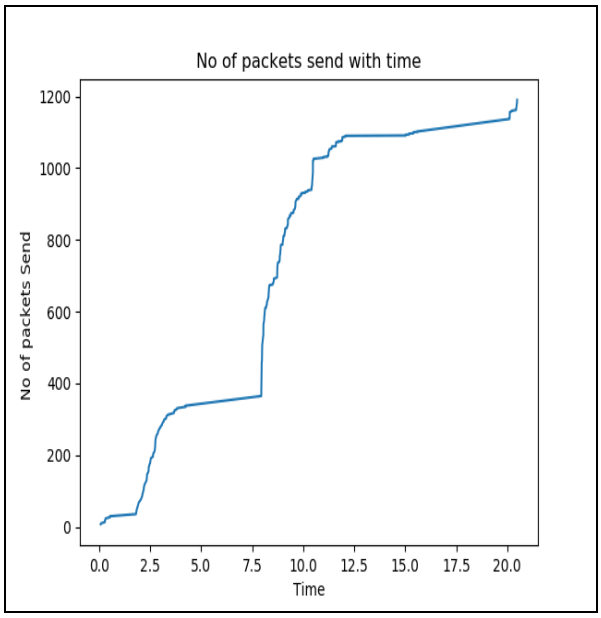

Figure 16. Number of packets injected during attack

Same kind of behavior is observed for the 2nd web site (Figure 17).

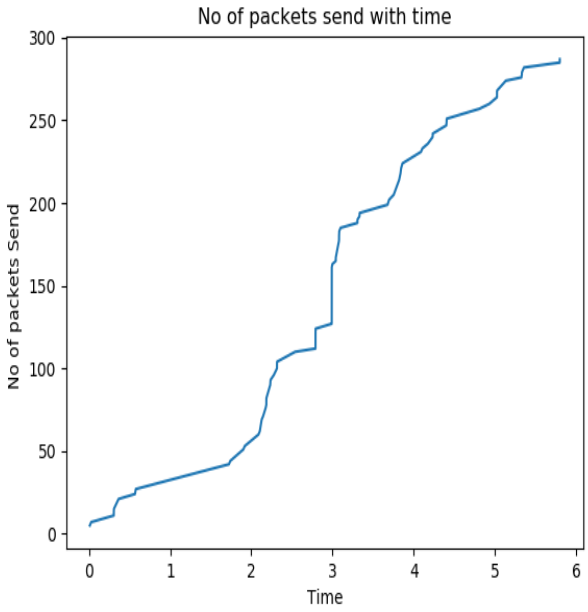

Figure 17. Number of packets under normal network condition.

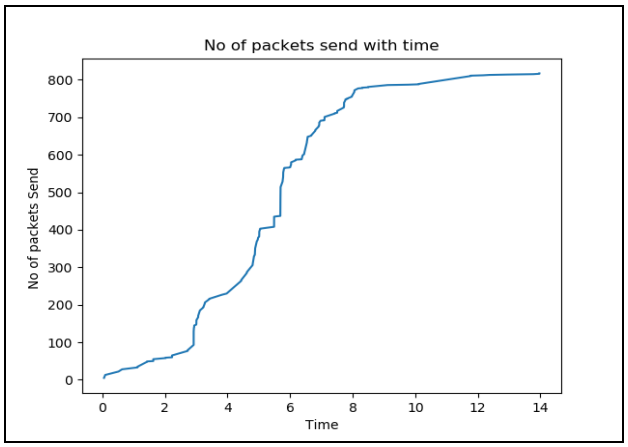

Figure 18. Number of packets during attack.

Extensive experiments are conducted to study the behavior of the network under normal condition and during replay attack. We have tested our method for six clients and it successfully detected the Replay attack. Picture given below (Figure 19) show the result of algorithm written in Python language.

```
Total no of packets in captured pcap file are 44
total no of these packets send in time
98.904147
1    0.0
11   33.689498
14   34.406258
15   35.327718
35   93.695688
38   94.310018
39   95.334021
40   97.381905
 This capture contains 8 ARP broadcast packets
 Broadcasted ARP packets sent in  97.381905 time
 Average time taken to send the ARP broadcast packets is
0.08215078561053
No ARP attack
```

Figure 19. Output of proposed method

In case of attack result is shown in Figure 20.

```
264   92.261802
265   92.271943
266   92.282094
267   92.292227
268   92.302371
269   92.312512
270   92.322646
273   118.870681
281   163.168172
854   178.677788
1452   218.002828
 This capture contains 261 ARP broadcast packets
 Broadcasted ARP packets sent in  184.135662 time
 Average time taken to send the ARP broadcast packets is
1.4174331966178284
ARP replay attack
```

Figure 20. Output of proposed method during attack

VII. CONCLUSION

In this paper, a lightweight solution for detecting Replay attack is given. In this paper for detecting the replay attack two algorithms are suggested. First algorithm uses type of protocol and nature of protocol field for detecting ARP request/replay attack. Second algorithm use different types of transmitted packet errors for detecting the Replay attack. In this paper number of retransmitted packets, number of Spurious Retransmitted packets and number of duplicates ACK are taken. Taking these three parameters together will decrease the false positive rates.

REFERENCES

1. Malekzadeh, Mina, Abdul Azim Abdul Ghani, and Shamala Subramaniam. "Design of cyberwar laboratory exercises to implement common security attacks against IEEE 802.11 Wireless Networks." Journal of Computer Systems, Networks, and Communications 2010 (2010): 5.
2. https://www.techopedia.com/definition/21695/replay-attack.
3. https://www.professormesser.com/security-plus/sy0-501/wireless-replay-attacks/
4. https://www.ettercap-project.org/
5. https://en.wikipedia.org/wiki/Cain_and_Abel_(software)
6. https://www.aircrack-ng.org/index.html
7. https://scapy.net/
8. https://www.colasoft.com/packet_player/
9. http://tcpreplay.appneta.com/
10. https://en.wikipedia.org/wiki/Replay_attack
11. http://resources.infosecinstitute.com/wireless-attacks-unleashed/
12. Feng, Zi, Jianxia Ning, Ioannis Broustis, Konstantinos Pelechrinis, Srikanth V. Krishnamurthy, and Michalis Faloutsos. "Coping with packet replay attacks in wireless networks." In Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on, pp. 368-376. IEEE, 2011
13. https://searchnetworking.techtarget.com/definition/Address-Resolution-Protocol-ARP
14. https://www.oreilly.com/library/view/packet-guide-to/9781449308094/ch04.html
15. https://www.geeksforgeeks.org/how-address-resolution-protocol-arp-works/
16. https://windowswizardry.wordpress.com/2017/05/22/arp-cache-in-windows/
17. https://www.dynatrace.com/news/blog/detecting-network-errors-impact-on-services/
18. https://www.dynatrace.com/news/blog/detecting-network-errors-impact-on-services/

AUTHORS PROFILE

Rajinder Singh is an Assist. Professor in DCSA, PUSSGRC, Hoshiarpur, Punjab, India. He has more than fifteen years of experience of teaching post-graduate classes. His areas of interest are Wireless Network Security, Cyber Security, Artificial Intelligence and Android Security. He is currently pursuing His Ph.D. Degree from P.U. Chandigarh, India. His mail id is rajinderid@gmail.

Dr. Satish Kumar is Associate Professor in Department of Computer Science and Applications in Panjab University (PU), Chandigarh (India), currently posted at Panjab University SSG Regional Centre, Hoshiarpur, Punjab, India (a multi faculty prestigious campus of PU). He has more than fifteen years experience of teaching post-graduate classes. His areas of interest are Image Processing, Pattern Recognition, computer graphics and Artificial Intelligence. He can be reached at satishnotra@yahoo.co.in.