2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)
Amity University, Noida, India. Sep 3-4, 2021

2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) | 978-1-6654-1703-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRITO51393.2021.9596480

Covert Communication using Address Resolution Protocol Broadcast Request Messages

Arti Dua
Depatment of Computer Science
University of Delhi
Delhi, India
arti.batra@bcas.du.ac.in

Vinita Jindal
Keshav Mahavidyalaya
University of Delhi
Delhi, India
vjindal@keshav.du.ac.in

Punam Bedi
Depatment of Computer Science
University of Delhi
Delhi, India
pbedi@cs.du.ac.in

***Abstract*—A covert channel is a hidden communication channel that is used to transfer information in a way that breaks the security policy of a system. A Network covert channel uses network protocols to send the secret information. In this paper, a technique for Network covert communication is proposed that uses Address Resolution Protocol (ARP) Broadcast Request messages to communicate covertly over a Local Area Network. The proposed technique uses a common seed value already known to both the sender and receiver to encode the secret data and thereby strengthening the imperceptibility of the secret message. The proposed technique is robust against traffic normalization and is also resilient to frequency analysis based decoding techniques. Further, it offers a covert capacity of 7 bits per ARP Broadcast Request packet.**

Keywords—Covert Channel, Address Resolution Protocol, Broadcast messages, Local Area Network, Information Hiding.

I. INTRODUCTION

Covert communications as a way of Information Hiding has been existing since ages. The use of covert channel is not a new concept. It was initially implemented in 440 B.C by the Greek ruler Histaeus [1]. At that time to convey a secret message, the Greek ruler used to shave the heads of their slaves, tattoo the secret message on their heads and wait for the hair to grow again on the their heads to conceal the message naturally. Once that was done, the slave was sent to the recipient of the secret message where he used to again shave off the slaves' heads to read the secret message. From this example, it is evident that covert communication is not a new concept and is being used since ages. The need to communicate in a hidden manner has always motivated researchers to develop more sophisticated covert communication techniques. In digital era also, covert channel requires a medium to carry a secret message. Covert channels can be created using various media, for example images, audio, video, network protocols etc. A Network covert channel can be created using the components of a network such as a Network Protocol packet. A Network Protocol packet has two parts: The Network Protocol Header and its Payload. The Network Protocol Header contains all the control information about the packet whereas the payload contains the actual data being carried by the packet. On the basis of the way in which secret data is hidden using the Network Protocol packets, the covert channel techniques can be classified into two categories which are Storage Network Covert Channels and Timing Network Covert channels [2]. Storage Network covert channels are a category of covert channels that hide the secret data in the Header and/or Payload part of the Network Protocol Packet. Timing Network covert channels are the other category of covert channels that hide secret data in inter-packet ordering/timing interpretations to hide secret data.

In this paper, we propose a technique for Network covert communication that uses the Address Resolution Protocol (ARP) to implement secret communication over a Local Area Network (LAN). The proposed technique uses the last octet of Target Protocol Address field of an ARP Broadcast Request message to hide the secret data. The contribution made by this work is that firstly, instead of hiding secret data directly in the Target Protocol Address field, the proposed technique uses random numbers and ascii code (corresponding to the character to be hidden) to encode and strengthen the imperceptibility of secret data on this channel. Further, the proposed technique uses different encoding values for the same occurrences of any character in the covert message which makes the frequency analysis based deductions of characters difficult. The proposed technique works on an assumption that the covert sender generates and sends ARP Broadcast Request messages only for covert communications. The Structure of rest of this paper is as follows. Section II gives the background and working of ARP. Section III presents the literature review. Sections IV gives the details of proposed technique. Section V describes the Experimental Study, Results obtained and Analysis of proposed covert channel based on Capacity, Robustness and Undetectability. Section VI concludes the paper.

II. BACKGROUND

The term Covert Channel was first introduced by Lampson [3]. He defined covert channels as the channels which are not meant for information transfer at all.

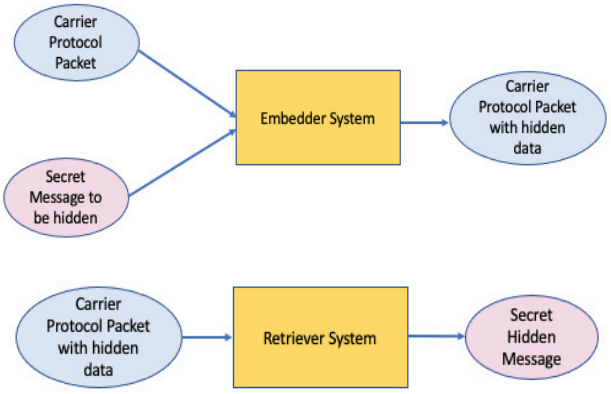

Fig. 1. Embedding (Top) and Retrieval (Bottom) of a hidden secret message in a Network Covert channel

Covert channels were initially interpreted as a threat, but if used in a positive way they can provide as a means of efficient information hiding mechanism like watermarking which refers to the embedding of a lucent digital signature (containing a data useful in the applications like broadcast monitoring, anti-tampering etc.) into a host signal [4]. As shown in figure 1, Network covert channels require a carrier protocol and an embedder system at the sender side to carry out covert transmissions. And at the receiver side, it requires a message retrieval system to fetch the covert message from the carrier protocol packet. In the proposed technique, the carrier protocol chosen for covert communication is Address Resolution Protocol. The next subsections briefly explain the header structure and the working of ARP protocol.

A. *Address Resolution Protocol*

The Address Resolution Protocol is one of the most important protocols of Local Area Networks. This protocol helps in mapping an Internet Protocol Address of a device to its Link Layer Address. The elaborate description of this protocol is given in RFC 826 [5]. The header format of an ARP packet is as shown in figure 2. The first field is the Hardware field which is 2 bytes long. It defines the protocol being used at the Link Layer. For example, for Ethernet this field has a value 1and for IEEE 802. standard this field has a value 6. The next field is Protocol Type which is also 2 bytes long. It defines the protocol being used at layer 3. For Internet Protocol version 4 (IPv4), the value of this field is 0x0800. Next is Hardware Address Length field which is 1 byte long. This field denotes the number of octets used in hardware address. For Ethernet addresses, this field holds the value 6. On similar line, the Protocol Address Length specifies the number of octets used in Protocol Address. For IPv4, this field has the value 4. Next field is Operation field which is 2 bytes long. This field specifies the type of ARP message. For ARP Request packets the operation value is set to 1 and for ARP Reply packets the operation value is set to 2. The Sender Hardware Address field denotes the hardware address of the source device. The Sender Protocol Address field denotes the Network Protocol Address of the source device. The Target Hardware address and the Target Protocol Address field denotes the hardware address and Network Protocol address of the target. In ARP Broadcast Request packet since the hardware address of the target is solicited, hence the Target Hardware Address field is set to all zeros.

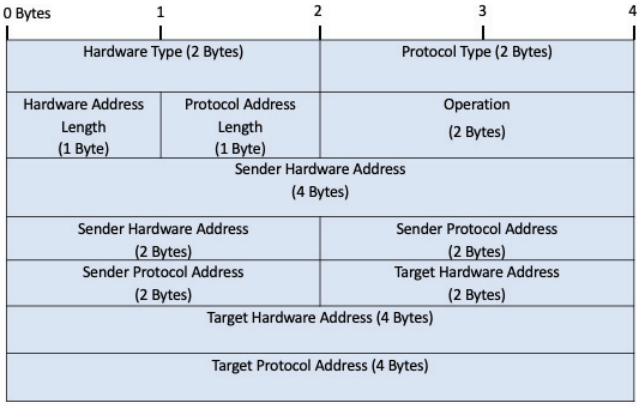

Fig. 2. ARP Header Structure

B. *Working of ARP*

A network layer switch needs to know the link layer hardware address of the next hop/destination in order to forward the traffic because a network layer switch encapsulates a network layer packet inside a link layer packet. Thus a mapping is required between the network layer address and the link layer address. This mapping is provided by Address Resolution Protocol. The Address Resolution Protocol helps in finding the hardware address of the device whose IP address is already known. For doing that, a device that wishes to know the hardware address of another device, generates a ARP Broadcast Request over the LAN.

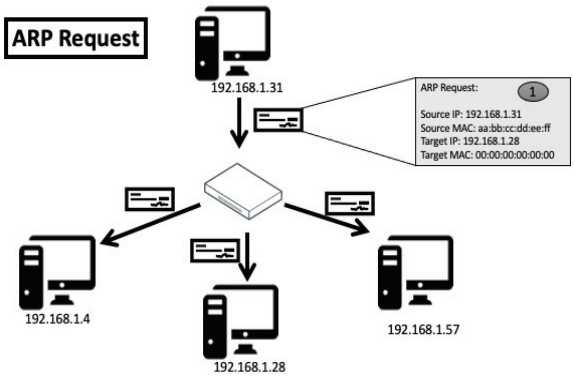

Fig. 3. ARP Request

The ARP Broadcast Request has operation field set to 1 and the Target Protocol address field set to the IP address of the device whose hardware address is not known. Since it is a broadcast request, all the active nodes over this LAN receive this ARP request but only that particular node whose IP address matches with the IP address mentioned in Target Protocol address field of this ARP Broadcast Request responds with an unicast ARP reply message. This ARP reply containing the hardware address of this node, is sent to the sender of the corresponding ARP Request message. The ARP Request and Reply are shown in figure 3 and figure 4 respectively.

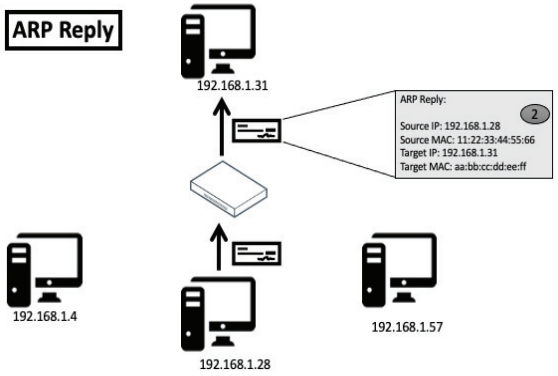

Fig. 4. ARP Reply

III. Related Works

Network Covert Channels is a popular field of research. Many Network covert channel techniques have been proposed over the existing protocols in the networks. A lot of

work has already been done in developing covert channels or techniques for Network Steganography (Techniques that hide data in existing network protocol packets) over protocols used in TCP/IP model. Szczypiorski suggested the possibility of creation of Network Steganography in various protocols including IP, ICMP, UDP and TCP which are the most commonly used protocols of the internet [6]. Trabelsi et al proposed a covert channel that used the record route option of IPv4 header [7]. Bedi et al proposed the use of overflow field in the Timestamp option of IPv4 packet for Network Steganography [8]. Further, Bedi et al suggested the use of presence and absence of Extension Headers in a particular order in an IPv6 header to transfer a five bits long secret message over a LAN [9]. Next, the covert channels developed using the ARP protocol are discussed. Jankowski et al suggested inter-protocol Network Steganography wherein they utilized two different protocols namely ARP and TCP to exploit the Etherleak vulnerability i.e. improper Ethernet padding for hidden communication [10]. Schmidbauer et al utilized a third party node's ARP cache as a dead drop for storing the secret messages [11]. The covert receiver retrieves message from this third party node using SNMP protocol. This technique largely depends on the usage of third party node for covert communication. Bedi et al used the concept of partial spoofing in ARP to create a covert channel over a LAN [12]. L. Ji et al utilized the target address field of the ARP broadcast packets [13]. They used the last 't' bits (for t lying between 4 and 7) of last byte of IP address to store the secret information. To identify the ARP requests carrying covert information, the first '8-t' bits of the last byte encode the sending second. The 't' bits which carried the secret data could easily be read by any node listening to ARP Broadcast request over the network. In this paper, a technique to implement covert communication is proposed that also hides data in the last octet of target address field in ARP Broadcast Request messages but in an encoded form in such a way that only the sender and the intended receiver can intercept the secret data. Randomization with common seed value is used at both the sender's and the receiver's side to ensure the imperceptibility of covert data such that the last octet of Target Protocol Address do not carry the secret data directly but in an encoded form. The secret data is encoded and decoded with the same set of random numbers generated at the sender's side and the receiver's side respectively with a seed value known in prior to both the sender and the receiver. The proposed technique is explained in detail in Section IV below.

IV. Proposed Technique

In this paper, a technique for Network covert communication is proposed which uses ARP Broadcast Request packets over the LAN for executing secret communications. The use of ARP Broadcast Request messages as a carrier was done because as observed in normal traffic scenarios, ARP Broadcast packets are usual packets seen in good frequency over a Local Area Network. The proposed technique uses the last octet of Target Protocol Address field of ARP Broadcast Request packet to carry secret data. Further, randomization with common seed value is used at both the sender's and the receiver's side to ensure the imperceptibility of covert data in a way that the last octet of Target Protocol Address field of ARP Broadcast Request packet do not carry the secret data directly. These secret data bits are encoded at the sender's site and decoded at the receiver's site with the same set of random numbers generated using a common seed value known to both the covert sender and the covert receiver in prior.

A. Sender Side

In the proposed technique, the sender side algorithm begins with inputting a common seed value known in prior to both the covert sender and the covert receiver. The purpose of using a common seed value is to generate same set of random numbers at the sender's and the receiver's side. These random numbers further help in encoding the secret bits to be carried in the last octet of Target Protocol Address field of ARP Broadcast Request packet. In the next step, a covert message that uses the character set from ASCII Character code is input. The 2^7 (128) characters supported by standard ASCII sufficiently represent all standard English alphabets, numerals, and punctuations. Once the message string is input it is stored in a string variable named covert_msg. Next, the first character is fetched from the covert_msg string and converted into its corresponding ascii code value. The standard ascii code value for all characters in ascii encoding lies between 0 to 127 inclusive of both 0 and 127. In the next step, the first random number is generated using a random number generator initialized with a seed value inputted at the start of the algorithm. The random numbers are generated between a fixed range of values starting from 128 to 255, inclusive of both 128 and 255. In the next step, the ascii code value of the fetched character is subtracted from the random number just generated. This subtraction would lead to any number between 1 to 255 which is put in the last 8 bits of Target Protocol Address field of ARP Broadcast Request Packet and this packet is broadcasted over the LAN. This procedure starting from fetching of a character from covert_msg to the sending of ARP Broadcast Request message is repeated till the time the complete covert_msg is not sent. An example of flow of events at the sender side is shown in figure 5. The Sender side algorithm for the same is given in figure 7.

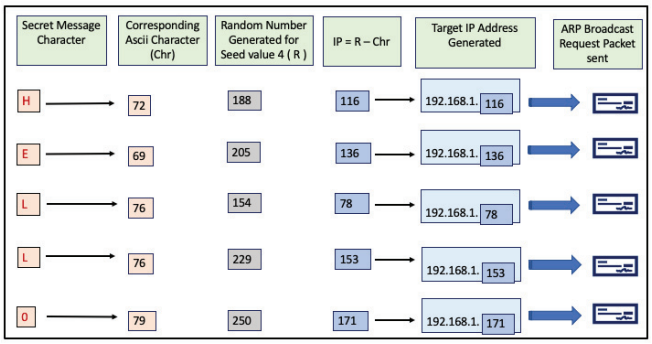

Fig. 5. Flow of events at Sender Side

The use of seed and the random numbers generated from this seed to encode the sending alphabet offers a few advantages. Firstly, as ARP Broadcast Request packets are received by every node connected to a LAN, hence if the data is put directly in the last octet of Target Protocol Address field then any node can read and interpret that data. But with the use of randomization with a prior known seed

value makes the decoding of the last octet of Target Protocol Address field of ARP Broadcast Request Packet dependent upon the random number generated in the same order with this seed value. This seed value and hence the random numbers values is only known to the covert sender and the intended covert receiver, thus nobody else can decode the secret bits. Secondly, if a character occurs more than once in a covert message then the 8 bit long encodings (to be put in last octet of Target Protocol Address field) generated for each occurrence will be different due to the use of different random numbers in the proposed technique. This can be seen in figure 5 where different Target Protocol Addresses are generated for same occurrence of character 'L' belonging to covert message string 'HELLO'.

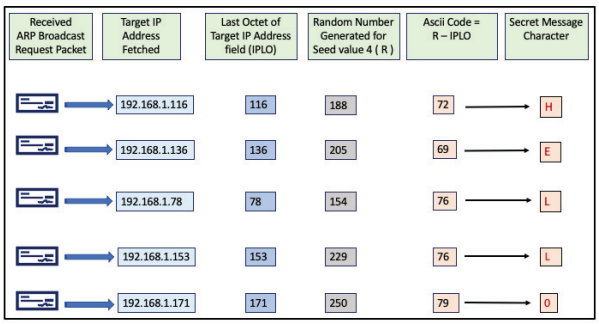

Fig. 6. Flow of Events at Receiver Side

B. Receiver Side

At the receiver side, the receiver algorithm begins with inputting the common seed value known to both the sender the receiver in prior. A variable string covert_msg is initialized to null to store the complete covert message. Next with the inputted seed value, the first random number is generated. After that the covert receiver listens for ARP Broadcast Requests having Source Protocol Address field set to the local IP address of the covert sender. As soon as an ARP Broadcast Request comes in from the covert sender, the receiver fetches the bits in last octet of Target Protocol Address field and converts it to decimal.

Input: A Secret Message M, Seed S
Output: ARP Broadcast Request Packets

1. Input a Seed value S and initialize random number generator with S.
2. Input a Secret Message and store it as M.
3. **While** len(M) != 0:
    a. Fetch the first character from the secret message M and store it as M_ch.
    b. Get the ascii code for character M_ch and store it as Ch_ascii.
    c. Update M by removing the M_ch character fetched in step 3a.
    d. Generate a random number between 128 to 255 using the random number generator initialized in step 1 and store it as R.
    e. Set IP = R – Ch_ascii
    f. Set IP_Addr = 192.168.1.IP
    g. Create an ARP Request packet with the Target Protocol Address Field set to IP_Addr.
    h. Broadcast this ARP Request.
4. Exit.

Fig. 7. Sender Side Algoithm

In the next step, this decimal value is subtracted from the random number generated in initial step to get the ascii value of hidden character. With this ascii value, the receiver deduces the character from the ascii table and appends this character to covert_msg string. These steps are repeated for every ARP Broadcast Request received from the covert sender. The flow of events at the receiver side is shown figure 6. The receiver listens to ARP Broadcast Requests till the time no more broadcast requests are received for an interval of one minute from the covert sender. After that the receiver reads the string covert_msg as the complete covert message received from the covert sender.

V. Experimental Study

A. Implementation

The proposed technique was developed, tested and experimented in a Local Area Network which consisted of a covert message sender and a covert message receiver having their corresponding local IP addresses. There were other devices connected in this Local Area Network as well. Scapy [14] library with python was used to craft and send the messages at the sender's side and to sniff and decode the messages at the receiver's side. Wireshark [15], a packet analyzer tool, was installed at both the Sender's and the receiver's side to anlyze and confirm the sending and receiving of ARP broadcast messages from the sender side and at the receiver side.

Input: Local IP Address of Covert Sender, Seed S
Output: Secret message

1. Input the common seed value S.
2. Initialize Covert_msg string to NULL.
3. Initialize random number generator with seed S.
4. Listen for incoming ARP Broadcast Requests having Source Protocol Address field as the local IP address of Covert Sender. Stop listening only when no ARP Broadcast Request packet is received for a time interval of one minute.
    a. **If** an ARP Broadcast Request is received having the Source Protocol Address as that of covert sender,
        i. Fetch the last octet from the IP address received in the Target Protocol Address field of this ARP Broadcast Request packet and store it as IP_Recd.
        ii. Generate a random number between 128 to 255 with the random number generator initialized in step 2. Store this number as R.
        iii. Set temp = R – IP_Recd
        iv. Get the character corresponding to temp in ascii encoding and store it as ch.
        v. Append ch to covert_msg string.
        vi. Goto beginning of step 4.
    b. **Else** If no ARP broadcast Request is received from the covert sender for one minute, then stop listening for more ARP requests and goto step 5.
5. Read the covert_msg string as the complete covert message sent by the sender.
6. Exit.

Fig. 8. Receiver Side Algoithm

At the Sender side, first of all the seed value and covert message is input. Each character is fetched from the covert message string and encoded using its ascii code and random number generated with a random number generator initialized with inputted seed. With this, an IP Address value carrying encoded character in the last octet, is created which is put in the Target Protocol Address field of ARP Request packet. This ARP Request packet is broadcasted over the network. These steps, starting from fetching of a covert

character from the message string, to the sending of ARP Broadcast Request are repeated till the time the complete covert message is not sent.

At the receiver's side, the receiver also begins with inputting the common seed value. The receiver also uses this seed value to generate same set of random numbers as done at the sender's side. The receiver then waits and listens for ARP Broadcast Requests from Covert sender. As soon as an ARP Broadcast Request is received, the local IP address from the Target Protocol Address field is fetched. The last octet of this fetched IP address is decoded with the help of the random number generated with random number generator function initialized with seed value to get the hidden ascii code. Next the character corresponding to this ascii code is found and is appended to a message string that holds the previously received covert characters (if any). When no ARP Broadcast Request is received from covert sender for an interval of one minute, the receiver stops listening for more messages and reads the message from the string containing the covert bits as the complete covert message.

B. Results

The covert message sender (Host A) and the covert message receiver (Host B) were connected to a Local Area Network having other actively connected nodes. Both Host A and Host B were assigned unique local IP addresses in the LAN. The secret message inputted at the sender side was "This is a covert message.". This message was successfully sent from the sender's side using ARP Broadcast Request messages. The successful sending of the secret message can be seen in the console snapshot taken at the sender's site as shown in figure 9. Further, figure 11 shows the successful sending of ARP Broadcast packets with Wireshark, running at the sender side.

The successful receiving of the same covert message at the receiver's side can be seen in the console snapshot taken at the receiver's side as shown in figure 10. Further, figure 12 shows the successful receiving of ARP Broadcast Request packets with Wireshark, running at the receiver side. Each ARP Broadcast Request message carried a single covert character which was encoded using its ascii value and random number generated using the common seed value. Moreover the proposed technique used different Target Protocol address values to communicate a same character that occurs more than once in the covert message which makes this technique resilient to frequency analysis based decoding techniques.

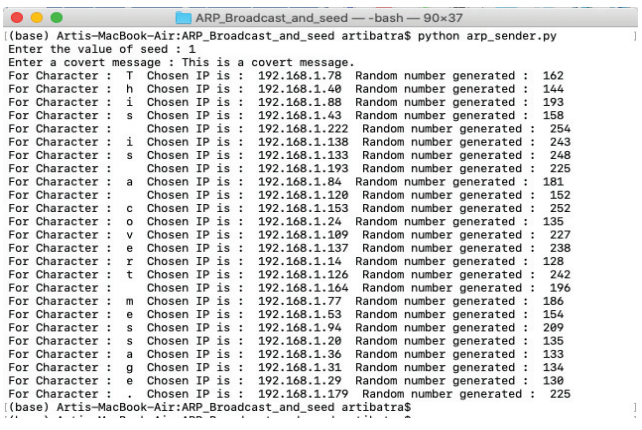

Fig. 9. : Sender's side Console snapshot

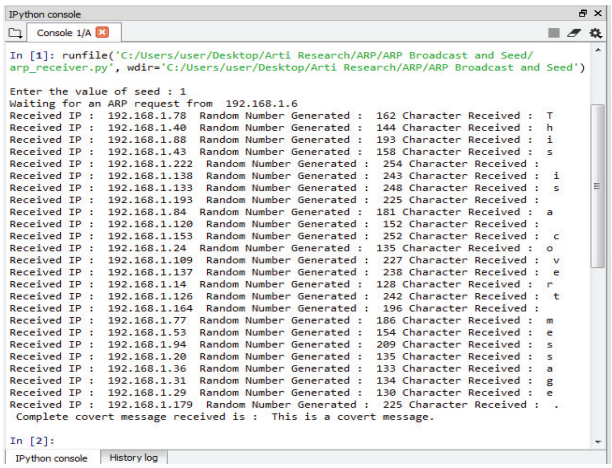

Fig. 10. Receiver's side Console snapshot

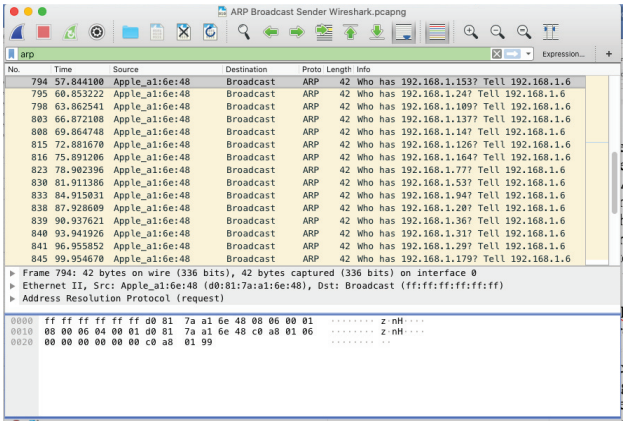

Fig. 11. Wireshark snapshot at Sender Side

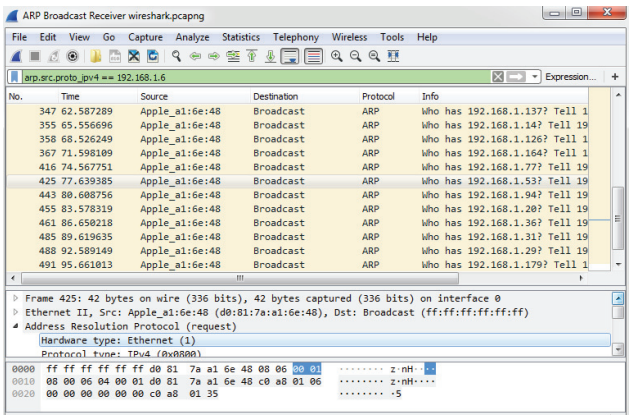

Fig. 12. Wireshark snapshot at Receiver Side

C. Analysis

This section performs the analysis of the proposed technique based on three important features which are capacity, robustness and undetectability.

1) Capacity

As shown in the results, the proposed technique offers a fixed steganography bandwidth of 7 bits per ARP Broadcast Request packet. The capacity of a covert channel not only depends upon the bits sent per ARP Request Packets but also on frequency of ARP Broadcast requests sent over the LAN. The number of ARP Broadcast Request messages over a network generally vary from one network to another.

This number primarily depends upon the activity and congestion over a network. When no response is received from an earlier existing node, ARP Broadcast Requests are sent multiple times or in good numbers. Thus, there is no fixed pattern on number of ARP Broadcast Request that may be sent over a network. In the proposed technique, synthetic ARP Broadcast Request messages are created and sent, each at an interval of 3 seconds. This interval value of 3 seconds was chosen to keep the number of ARP Broadcast Requests to unallocated IP addresses to an optimum number so as to avoid overwhelming the network with large number of ARP Broadcast Request messages. Thus the capacity of the proposed channel in terms of per unit time is 140 bits/minute. The time interval can be increased or decreased depending upon the frequency of ARP packets observed over a network.

2) Robustness

Many of the covert channels discussed in the literature are vulnerable to traffic normalization attacks and hence offer low robustness. The Target Protocol Address field in an ARP Request message field contains the IP Address of the destination whose hardware address is solicited. This field can not be changed for any traffic normalization as this may completely defeat the purpose of an ARP Broadcast Request over a LAN, hence this field is completely resistant to normalization attacks as compared to other fields of ARP Request packet like Target Hardware Address field. The packet loss, which is a common phenomenon over a network may affect the robustness of the proposed technique.

3) Undetectability

A covert channel becomes easily detectable if it raises the amount of network traffic in an unusual way. Thus to maintain undetectability in terms of change in the amount of network traffic, the proposed technique sends controlled number of ARP Broadcast Request packets, which is one ARP Broadcast request packet in every three seconds over the LAN. The proposed covert channel can only be detected if a network aware active warden observes and catches the use of ARP Broadcast Requests to unallocated local IP addresses over the network. Even with the detection of the covert channel, the secret bits cannot be interpreted as it requires the use of seed for generating random numbers which is only known to the covert sender and covert receiver.

VI. Conclusion

In this paper, a technique for Network covert channel is proposed and experimented, which is capable of sending covert information from a covert sender to a covert receiver over a LAN using Address Resolution Protocol. The proposed technique uses the last octet of Target Address Field of ARP Broadcast Request packet for covert communication. The proposed technique offers a fixed capacity of 7 bits per ARP Broadcast Request message. Secret messages of any size may be sent using the proposed technique by breaking the message into a group of characters and sending each character hidden inside one ARP Broadcast Request message. Moreover, instead of hiding secret data bits corresponding to a character directly in the last octet of Target Address Field, the secret bits are encoded with the help of ascii code corresponding to that character and a random number generated with a common seed value known to both the covert sender and the covert receiver in prior. This increases the imperceptibility of secret data over the LAN. Further, the proposed technique generates two different Target Protocol Address values corresponding to a same character which occurs more than once in a hidden message. This makes frequency based deductions of characters difficult. Also, this covert channel offers high resistance to traffic normalization over the network. In the proposed technique, ARP Protocol is used for implementing this covert channel. Any other protocol that supports broadcast messages in a similar manner may be used to develop a similar type of covert channel over a Network. For future work, we aim to uncover more such covert channels that can be developed using ARP and other Network Protocols.

References

[1] A. Siper, R. Farley and C. Lombardo, "The rise of steganography.," in *Proceedings of student/faculty research day, CSIS, Pace University (2005).*, 2005.

[2] L. Caviglione, "Trends and Challenges in Network Covert Channels Countermeasures," *Applied Sciences,* vol. 11, no. 4, p. 1641, 2021.

[3] B. W. Lampson, "A note on the confinement problem.," *Communications of the ACM,* vol. 16, no. 10, pp. 613-615, 1973.

[4] R. Anand, G. Shrivastava, S. Gupta, S. P. Lung and N. Sindhwani, "Audio Watermarking With Reduced Number of Random Samples," in *Handbook of Research on Network Forensics and Analysis Techniques*, IGI Global, 2018, pp. 372-394.

[5] D. Plummer, "An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware, STD 37, RFC 826,," November 1982. [Online]. Available: DOI 10.17487/RFC0826. . [Accessed 10 July 2021].

[6] K. Szczypiorski, "Steganography in TCP/IP networks," in *State of the Art and a Proposal of a New System–HICCUPS, Institute of Telecommunications' seminar, Warsaw University of Technology*, Poland, 2003.

[7] Z. Talbresi and I. Jawahar, "Covert File Transfer Protocol based on the IP Record Route Option," *Journal of Information Assurance and Security,* vol. 5, no. 1, pp. 64-73, 2010.

[8] P. Bedi and A. Dua, "Network Steganography using the Overflow Field of Timestamp Option in an IPv4 Packet," in *Procedia Computer Science 171*, Trivendrum, 2020.

[9] P. Bedi and A. Dua, "Network Steganography using Extenson Headers in IPv6," in *5th International Conference Information, Communication & Computing Technology (ICICCT-2020)*, Delhi, 2020.

[10] B. Jankowski, W. Mazurczyk and K. Szczypiorski, "PadSteg: Introducing Inter-Protocol Steganography," *Telecommunication Systems,* vol. 52, p. 1101–1111., 2013.

[11] T. Schmidbauer, S. Wendzel, A. Mileva and W. Mazurczyk, "Introducing dead drops to network steganography using ARP-caches and SNMP-walks," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, 2019.

[12] P. Bedi and A. Dua, "ARPNetSteg: Network Steganography using Address Resolution Protocol.," *International Journal of Electronics and Telecommunications,* vol. 66, no. 4, pp. 671-677, 2020.

[13] L. Ji, Y. Fan and C. Ma, "Covert channel for local area network," in *In 2010 IEEE International Conference on Wireless Communications, Networking and Information Security*, 2010.

[14] P. Biondi, "Scapy Website," 2021. [Online]. Available: https://scapy.net. [Accessed 1 July 2021].

[15] "Wireshark," Wireshark Foundation, 2021. [Online]. Available: https://www.wireshark.org. [Accessed 25 June 2021].