Int J Syst Assur Eng Manag
https://doi.org/10.1007/s13198-019-00919-1

ORIGINAL ARTICLE

An analysis of security solutions for ARP poisoning attacks and its effects on medical computing

B. Prabadevi[1] · N. Jeyanthi[1] · Ajith Abraham[2]

Received: 8 June 2018 / Revised: 1 November 2019

Abstract Network utilization reached its maximum level due to the availability of high-end technologies in the least cost. This enabled the network users to share the sensitive information like account details, patient records, genomics details for biomedical research and defence details leading to cyber-war. Data are vulnerable at any level of communication. The link-layer Address Resolution Protocol (ARP) is initiated for any data communication to take place among the hosts in a LAN. Because of the stateless nature of this protocol, it has been misused for illegitimate activities. These activities lead to the most devasting attacks like Denial of Service, Man-in-the-Middle, host impersonation, sniffing, and cache poisoning. Though various host-based and network-based intrusion detection/prevention techniques exist, they fail to provide a complete solution for this type of poisoning. This paper analyzes the existing defence systems against ARP attacks and proposes three different techniques for detecting and preventing the ARP attacks. The three techniques ensure security of traditional ARP and its impact in Medical computing where a single bit inversion could lead to wrong diagnosis.

✉ B. Prabadevi
prabadevi.b@vit.ac.in

N. Jeyanthi
njeyanthi@vit.ac.in

Ajith Abraham
ajith.abraham@ieee.org; abraham.ajith@gmail.com

[1] School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India

[2] Scientific Network for Innovation and Research Excellence, MIR Labs, NW, Auburn, USA

Keywords Address resolution protocol · Spoofing · Cache poisoning · DoS · MitM

1 Introduction

1.1 ARP and ARP cache

On a LAN, two hosts can communicate only if they knew their MAC addresses, if not an ARP broadcast request message would be sent to all the other hosts in the network and turn the one with matching IP address will reply with a unicast ARP reply message with its MAC address. ARP Cache table, {Protocol address (32 bits), a Hardware address (48 bits), Type, Interface is stateless, not secured and paves the way for the attacker to poison the cache with bogus entries. ARP cache poisoning may lead substantial devastation effects on the network. Malevolent hosts in the network can perform many types of attacks to the network like ARP spoofing, Man-in-the-Middle (MitM) attack, Denial of Service (DoS) Attack (Prabadevi and Jeyanthi 2014) impersonating the hosts, over flooding the network traffic and so.

ARP cache poisoing, with fake IP –MAC pairs can be carried by anyone with some scripting knowledge and by using various open source tools for carrying out this spoofing (Al-Hemairy et al. 2009). ARP cache can be populated in two ways either statically by a smaller network or dynamically by the larger number of hosts. Static cache entry incurs enormous maintenance overhead, whereas the dynamic cache entry reduces the maintenance cost involved (Trabelsi 2016) since the OS administers.

The problems with spoofing and other types of attacks are more prone to dynamic cache entry method as no

Published online: 13 November 2019

security had been ensured by the ARP Protocol (Plummer 1982). ARP serves depending on the scenario as:

- *ARP* (Plummer 1982) Resolves the given 32 bit IP address to 48-bit MAC address in LAN
- *Proxy-ARP or ARP subnet Gateway* (Quarterman 1987) Proxy ARP in the gateway resolves the address on behalf of the target host.
- *Reverse ARP (RARP)* Maps given MAC Address to its corresponding IP address.
- *Gratuitous ARP (GARP)* A special ARP packet used for making ARP announcements in dynamic IP configuration or each time the interface goes down to update ARP cache with entries. It is also used for IP conflict detection (Cheshire 2008).
- *Inverse ARP* (Bradley et al. 1998) It apes the ARP protocol. It resolves a known IP address to Data Link Connection Identifier (DLCI) of Frame Relay stations.

2 Related works

Various solutions like S-ARP (Bruschi et al. 2003),T-ARP (Lootah et al. 2007), G-ARP (Dangol et al. 2011), P-ARP (Saputro and Akkaya 2015) ES-ARP (Hammouda and Trabelsi 2009), DS-ARP (Song et al. 2014) for ubiquitous environment, IS-ARP (Samvedi et al. 2014) and tools like Sax2,XArp, Snort, ARPWatch, Anti-arpSpoof, ARP-ON, Antidote,ARP-Guard, Prelude, Seconfig to defend against these attack (Hingne and Jain 2016). Table 1 provides details about the IDS defence mechanisms. Though most robust versions of operating systems are evolving, they are still vulnerable to ARP attacks. As in (Trabelsi 2016), Windows OS provide ARP stateful examination whereas MAC does only stateless inspection. However, the older versions of OS like Solaris, Windows allows static ARP to be updated with ARP reply request messages. Unlike Apple's OS versions, different versions of Linux OSs also provide the same level of resilience as with Windows versions (Jana 2017). So an effective algorithm to detect and prevent ARP attacks must be deployed irrespective of the operating system being used.

3 Enhancements to existing protocol feature by cryptographic techniques

These cryptographic techniques provide a good solution but may reduce the performance of the ARP protocol.

Rupal et al. (2016), proposed an ICMP based utility which includes user registration, ICMP based secondary cache and a detection algorithm for detecting and preventing ARP cache poisoning attack. Authors used radius server and MySql for the authentication process; data is encrypted using hash value of user id created during authentication. Though it detects and prevents ARP based DoS and MitM, the utility has to be installed in all the hosts.

3.1 Centralized server/middleware approaches

The researchers used a central server for voting, Certificate distribution, and granting tickets as mentioned in the Table 1, which may impose additional overhead and may even lead to the central point of failure.

3.2 Intrusion detection and prevention systems

Commercialized and open source tools are available in the market for the different operating environment to mitigate ARP attacks caused by either malicious activity or by some tools like Net-cut, CommView, frameip, Arpspoof, Arpoison and so (Kaur and Dhanda 2014).

3.3 ICMP and DHCP based approaches

Researchers proposed the modified ARP protocol by making use of the ICMP and DHCP (Rupal et al. 2016; Issac 2009).

3.4 Defence systems based on port feature

The switches specified in Table 1. are capable of detecting and preventing the ARP cache poisoning attacks by using Dynamic ARP Inspection-DAI feature can detect and mitigate ARP cache poisoning attacks, but are expensive (Al-Hemairy et al. 2009; Song et al. 2014).

The defence mechanism against ARP attacks can be classified into five categories [A–E] as mentioned above. Apart from these one common way of doing this manually configuring the static entries in all PC, but it will not be feasible in case of larger network.

4 Proposed methodology

Of all the approaches that we have discussed in the survey only a few are performing the cross-layer check. Al-Hemairy et al. (2009) stated the set of requirement that any security solutions or defence against to ARP must possess. In this paper, we recommend three technique which satisfies the basic requirements stated by (Al-Hemairy et al. 2009; Trabelsi and El-Hajj 2010). The techniques are Cross-layer Consistency Checking (CLCC), Timestamp and Counter based approach (TSCBA) and Extended TCBA in large data centre networks. (TCDCN). The overall working of these techniques is depicted in Fig. 1.

Table 1 CLCC versus TSCBA versus TCDCN

Defence mechanism	Type	Cross-layer check	Attacks detected	Comments
XArp 2	IDS	Yes	ARP spoofing	Only detection Should installed in hosts
Sax2	IDS	No	Prevents ARP storm and Scanning	No detection Should installed in hosts
Snort	IDS	Yes	ARP spoofing, DoS, MitM	Only detection Should installed in hosts
Anti-ARP,	IDS-Windows	No	Detects and prevents MitM and DoS	Only detection
Anti-NetCut, No-cut	IDS- Linux and MAC	No	Does not detects but prevents MitM and DoS	Defence against Netcut, but fails when DoS is heavy
Juniper EX3200 Cisco Catalyst 6500	Switch, Port Security feature	Yes	Detects MAC cloning, MitM and DoS, does not prevent ARP spoofing	Expensive, detects invalid pair by DHCP snooping
ARPWatch, ARPStar	IDS Linux	No	Detects Passive ARP poisoning attacks	Uses time stamp
ARP Guard	IDS-Mac, Android	No	Detects active and passive attacks	Works for both wired and wireless
ARP-ON	IDS-Linux	No	Prevents MitM	Performs Static and dynamic ARP inspections
Antidote & Anticap	Linux Kernel based appliance	No	MitM and DoS	Rejects ARP replies containing a MAC address different from ARP cache
Anti-ARP	IDS Windows	No	MitM and DoS	Detects and prevents but not in all cases
zANTI and cSploit	Android IDS	No	MiTM	Detects passive attacks
Colasoft CAPSA	IDS windows	No	MitM and DoS	Detects passive ARP poisoning
AVASS and DAPS (Puangpronpitag and Masusai 2009)	Kernel-based patches for Windows and Linux	No	MitM and DoS	Detects and prevents. Uses digital signatures for authentication
SARP	Cryptographic solution	No	Prevents ARP poisoning	Adds authentication to ARP, but incurs key Distribution overhead and Central point of failure with CA, Requires kernel module and user daemon
S-UARP (Issac 2009)	Cryptographic Approach, Follows unicast ARP Request reducing the broadcast congestion	No	Prevents ARP poisoning	Extends DHCP, Additional Maintenance overhead on Key management
ES-ARP	Cryptographic solution	No	It may prevent ARP poisoning	Not implemented and tested
DS-ARP	Network routing trace approach	No	Prevents Host impersonation, DoS, MitM	Real-time implementation not discussed
E-ARP	Voting based and long-term cache	No	MitM	Prevents MiTM, but takes a long time and huge storage space
IS-ARP	Centralized server	No	Prevents ARP spoofing, DHCP based DoS	The central point of failure
TARP	Cryptographic and ticket agent-based- tested on Linux	No	MAC spoofing	Causes Ticket flooding leading to DoS and host Impersonation. Requires kernel module and user daemon
GARP	Cryptographic approach and broadcast ARP reply	No	Detects and prevents ARP poisoning attack	Requires a Key Distributor, induces additional overhead
PARP	Elliptic curve cryptographic approach	No	Prevents ARP cache poisoning	Induces additional overhead because of ECC

Table 1 continued

Defence mechanism	Type	Cross-layer check	Attacks detected	Comments
KARP	Cryptographic approach	No	ARP spoofing	Uses Ticket Granting Server which incurs addition computational cost
E-SDE (Pandey 2013)	Probe-based technique	No	ARP spoofing	Invites additional space and overhead
Arpsec	TPM based security protocol coexists with ARP	No	Spanning tree attack and VLAN attacks	Introduces about 15% of overhead than ARP

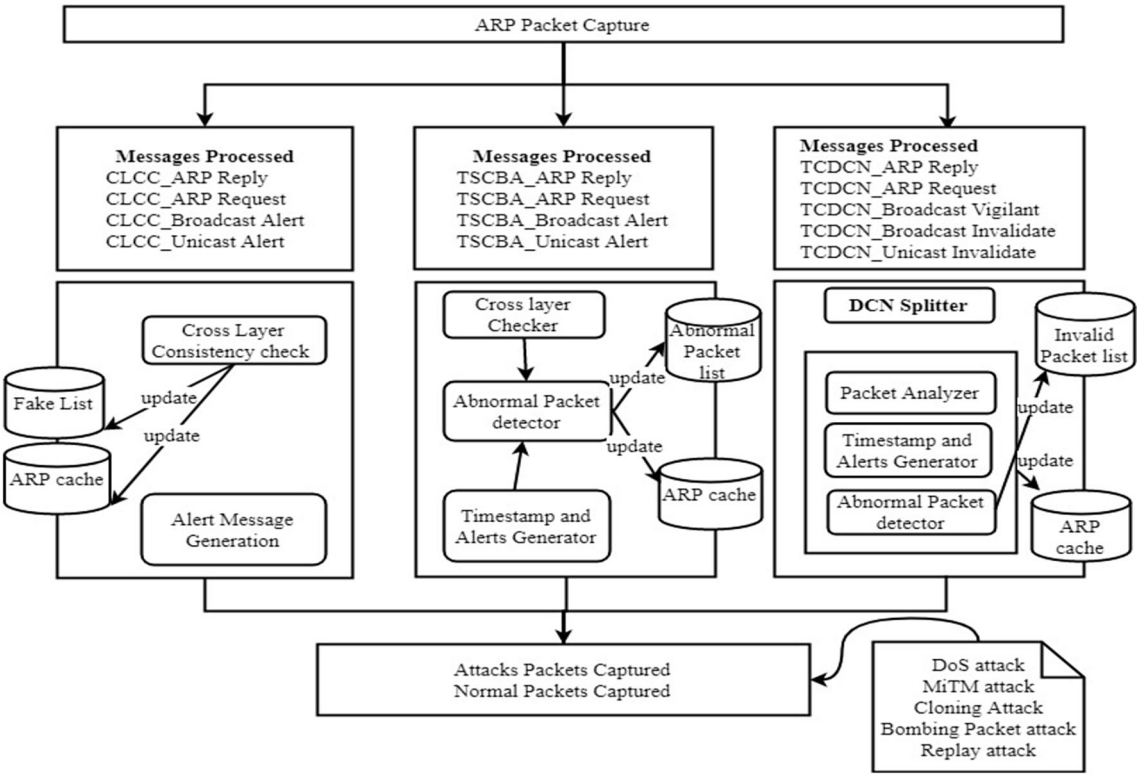

Fig. 1 ARP mitigation techniques

4.1 Cross-layer consistency checking (CLCC) approach

The CLCC approach performs three processes viz., Cross-layer Consistency checking, Alert Message Generation and Fake list Table updation. It maintains two tables: ARP cache and Fake list Table containing details of IP-MAC pairs of the hosts in the network and IP-MAC pair of fake packets introduced in the network respectively. The cross-layer consistency checking is performed by cross-checking the MAC address in the Ethernet layer and MAC address in ARP Layer. If the MAC addresses match the ARP message is accepted otherwise discarded, and the entry is added to Fake list table. Also, CLCC approach clears ARP cache for every 10 min. The alert message generation process introduces two new message viz., Unicast Alert message and Broadcast Alert message as depicted in Figs. 2 and 3 respectively. In the figures, IP-R refers to IP address of the router and IP-D, MAC-D refers to IP and MAC addresses of host D. When the cross-layer checking fails the respective host will send a unicast alert message to the Router about the fake entry and updates the fake list. Meanwhile before cross-layer checking if IP-MAC pair is found in fake list entry, a broadcast alert message is sent to

Sender-IP : IP-D	Sender-MAC : MAC-D
Target-IP : 255.255.255.255	Target-MAC: ff:ff:ff:ff:ff:ff
Opcode= 25	

Fig. 2 CLCC broadcast alert message

Sender-IP : IP-D	Sender-MAC : MAC-D
Target-IP : IP-R	Target-MAC: MAC-R
Opcode= 26	

Fig. 3 CLCC unicast alert message

all the other hosts in the network to avoid forging from the same attacker. CLCC has the following snags: NULL MAC addresses, and Multicast MAC addresses are not focussed, Gratuitous ARP packets were also left uncovered and the fake list entry is not updated or cleared frequently which can be vulnerable to certain attacks or can create unwanted chaos in the network.

The CLCC's Request reply processing at hosts and Router has described the algorithm below.

Algorithm CLCC's ARP Request Reply Processing at Router

ARP Request Reply Processing at Router R
Assume router has received a Unicast_Alert_Message
if $opcode == 26$ **then**
  if $arp\_IPR == IPR$ **then**
    if $arp\_MACA == Eth\_MACA$ **then**
      if arp_IPA *and* arp_MACA *exists in Router cache* **then**
        Accept Alert
      else
        Add arp_IPA, arp_MACA to fake list
        Ignore Alert
      end
    else
      Ignore Packet
      Add arp_IPA, arp_MACA to fake list
    end
  else
    Ignore Packet
    Add arp_IPA, arp_MACA to fake list
  end
end
Clear the ARP-Cache for every 10 minutes

CLCC's ARP Request Reply processing at Router

4.1.1 Variables in CLCC algorithm

arp_IPA ← IP address of host A in ARP header
arp_IPD ← IP address of host D in ARP header
arp_MACA ← MAC address of host A in ARP header
arp_MACD ← MAC address of host D in ARP header
Eth_MACA ← MAC address of host A in Ethernet header
Eth_MACD ← MAC address of host D in Ethernet header
arp_IPR ← IP address of Router R in ARP header
Eth_MACR ← MAC address of Router R in Ethernet header
Assumption: The host A does not know the MAC address of Host D where:
Sender details: arp_IPA,arp_MACA and Eth_MACA
Receiver details: arp_IPD,arp_MACD and Eth_MACD

Algorithm CLCC's ARP Request Reply Processing

Generate ARP Request At A
set opcode ← 1 in ARP Header
send $ARP\_Request\_Message(ARPHeader, EthernetHeader)$
ARP_Request_Processing At D
Decode the packet
if $opcode == 1$ **then**
  if $arp\_MACA == Eth\_MACA$ **then**
    Update ARP cache with IP-MAC pair in the ARP Request
    set opcode ← 2 in ARP Header
    send $\_ARP\_Reply\_Message(ARPHeader, EthernetHeader)$
  else
    Store the forged IP-MAC address into fake_list
    set opcode ← 26 in ARP Header
    send $Unicast\_Alert\_Message$ to the Router or gateway
    if arp_IPA *and* arp_MACA *already exists in fake_list* **then**
      $setopcode$ ← 25 in ARP Header
      $sendBroadcast\_Alert\_Message$
    end
  end
end
ARP_Reply_Processing At A
if $opcode == 2$ **then**
  if $arp\_MACD == Eth\_MACD$ **then**
    Update ARP cache with IP-MAC pair in the ARP Reply
  else
    Store the forged IP-MAC address into fake_list
    set opcode ← 26 in ARP Header
    send$Unicast\_Alert\_Message$ to the Router or gateway
    if arp_IPA *and* arp_MACA *already exists in fake_list* **then**
      set opcode ← 25 in ARP Header
      $sendBroadcast\_Alert\_Message$
    end
  end
end
Clear the ARP-Cache for every 10 minutes

CLCC's Request Reply Processing at host

4.2 Timestamp and counter based approach (TSCBA)

TSCBA approach has the following processes viz., Cross-layer checking, abnormal packet detection, Timestamp generation, Alert message generation and Abnormal Packet list updation. Cross-layer checking is as performed in CLCC. The data tables are TSCBA's ARP cache and abnormal list tables. The traditional ARP table is modified by a new entire Timestamp (TS) as depicted in Fig. 4. The Fake list of CLCC is modified with two entries Count and TS to generate abnormal list table as depicted in Fig. 5 to further enhance the security. The alert messages used in TSCBA has two new entries TS_g, TS_t timestamp generation and timestamp expiration time to avoid the security issues with these type of packets. These messages are depicted in Figs. 6 and 7.

The count field in abnormal list table helps to avoid ARP cache poisoning from the same attacker trying to

Protocol	IP Address	MAC Address	ARP Type	Interface	TS_{arp}(time)
Internet IP/TCP	172.168.0.1	00:50:79:66:68:01	ARPA	FastEthernet 0/1	2016-07-14 04:32:26

Fig. 4 TSCBA's ARP cache

Index	IP Address	MAC Address	Count	TS_0(time)
1	198.164.0.3	00:3:44:56:22:34	1	2016-07-13 05:32:29
2	165.178.0.5	00:98:98:76:34:56	10	2016-07-12 03:12:01

Fig. 5 TSCBA's abnormal list table

Source-IP	Source-MAC
Destination-IP (broadcast address)	Destination-MAC
Fake IP	Fake MAC
ICMP Message	"Beware of this host"
Opcode= 3	
TS_g	TS_t

Fig. 6 TSCBA's broadcast alert message

launching a bombing packet attack and DDoS attack. The abnormal list table is updated whenever cross-layer checking fails, or abnormal packets like invalid MAC, NULL MAC, Multicast MAC are suspected during transmission and count is maintained to overcome the above-said attacks.

TSCBA algorithm makes following assumption:

- A network with n nodes
- ARP cache is cleared for every 20 min
- TSt = TSg + 10 s (it may vary based on n and network latency) where,
- Opcodes used

  - ARP_Request → opcode = 1; ARP_Reply → opcode = 2; ARP_Broadcast_Alert_Message → opcode = 3;
    ARP_Unicast_Alert_Message → opcode = 4

TSCBA algorithm includes the following segments: TSCBA ARP Request Processing, TSCBA Reply Processing, TSCBA Unicast Alert message Processing, TSCBA's Broadcast Alert message processing details how the ARP requests, replies and alert messages are processed,

Source-IP	Source-MAC
Destination-IP	Destination-MAC
ICMP Message	"ARP Reply time expired"
Opcode= 4	
TS_g	TS_t

Fig. 7 TSCBA's unicast alert message

S.No.	Variables	Description
1	Eth_MAC	MAC address in Ethernet Header
2	Arp_IP	IP address in ARP Header
3	Arp_MAC	MAC address in ARP Header
4	TS_g	Timestamp generation time
5	TS_t	Timestamp validity time
6	n	No of nodes in network1 to maximum capacity of the LAN
7	$Packet_{req}$	ARP Request Packet
8	$Packet_{rep}$	ARP Reply Packet
9	$Packet_{bst}$	ARP Broadcast Alert Packet
10	$Packet_{ust}$	ARP Unicast Alert Packet

Fig. 8 TSCBA algorithm nomenclature

i.e. how the host reacts on receiving a request packet, reply packet, unicast alert message and broadcast alert message respectively. On receiving an TSCBA's ARP message, the system first ensures the received message type through the opcode of the message, in turn it performs the cross layer consistency checks to avoid MAC spoofing. For request it checks whether it has broadcast address in its destination field for generating reply else it will update the corresponding tables and drop the packets. Everytime the timestamp is generated and validated. In case of ARP reply, it ensures it is an unicast and checks for the Gratuituous ARP Reply/Request. If the received packet is alert message then it checks whether is is a unicast or broadcast. The unicast alert is generated in case of timestamp expiration and the broadcast alert is generated once a invalid packet is detected.

Algorithm TSCBA's ARP Request Processing

```
for i → 1 to n do
    clear.ARP_Cache(20 minutes) /**ARP Request Generation**/
    if Destination_MAC is not Known then
        Generate timestamp for ARP Request message to be sent
    else
        Start Data Transfer
    end
    /**ARP Request Processing**/
    Decode.Packet_req
    if opcode == 0x0001 then
        if SourceEth_MAC == SourceArp_MAC) and
        (DestinationArp_MAC == ff : ff : ff : ff : ff : ff then
            if Arp_IP andArp_MAC == IP − MAC pair in cache then
                Extract TimeStamp if (TS_t - TS_g) ≤ 10 then
                    Generate Reply appended with Timestamp
                else
                    drop.Packet_req
                end
            else
                if SourceArp_IP == destinationArp_IP then
                    /*Gratuituous AARP_Request Packet*/
                    update.ARP_Cache with Timestamp TS_g
                else
                    Extract TimeStamp if (TS_t - TS_g) ≤ 10 then
                        Update ARP_cache with TS_g
                    else
                        drop.Packet_req
                    end
                end
            end
        else
            drop.Packet_req update. Abnormal packet list table Generate Broadcast
            Alert Message with TimeStamp
        end
    end
end
```

TSCBA's ARP Request Processing

Algorithm TSCBA's ARP Reply Processing

```
for i → 1 to n do
    clear.ARP_Cache(20 minutes)
    /**ARP Reply Generation**/
    Decode.Packet_rep
    if opcode == 0x0002 then
        if SourceEth_MAC == SourceArp_MAC) and
        (DestinationEth_MAC == DestinationArp_MAC then
            if Arp_IP andArp_MAC == IP − MAC pair in cache then
                Extract TimeStamp if (TS_t - TS_g) ≤ 10 then
                    update.ARP_Cache with Timestamp TS_g
                else
                    drop.Packet_rep Generate Unicast Alert message to the source
                    about elapsed time
                end
            else
                if SourceArp_IP == destinationArp_IP and
                SourceEth_MAC == Sourcearp_MAC and
                DestinationArp_MAC == SourceArp_MAC then
                    /*Gratuituous ARP_Reply Packet*/
                    update.ARP_Cache with Timestamp
                else
                    drop.Packet_rep Generate Broadcast Alert Message with
                    timestamp
                end
            end
        else
            drop.Packet_rep Generate Broadcast Alert Message with timestamp
        end
    end
end
```

TSCBA's ARP Reply Processing

The nomenclature of TSCBA is depicted in Fig. 8.

Algorithm TSCBA's Broadcast Alert Message Processing

```
for i → 1 to n do
    clear.ARP_Cache(20 minutes)
    Decode.Packet_bst
    if opcode == 0x0003 then
        if SourceEth_MAC == SourceArp_MAC) and
        (DestinationEth_MAC == DestinationArp_MAC then
            if SourceArp_IP andSourceArp_MAC ==
            IP − MAC pair in cache then
                if DestinationArp_MAC ==
                broadcastand(DestinationArp_IP == broadcast then
                    Extract TimeStamp
                    if (TS_t - TS_g) ≤ 10 then
                        update.ARP_Cache with Timestamp
                        update. Abnormal packet list table with invalid details
                    else
                        drop.Packet_bst
                        Generate Unicast Alert message to the source about
                        elapsed time
                    end
                else
                    drop.Packet_bst
                    update. Abnormal packet list table with Timestamp
                end
            else
                drop.Packet_bst
                update. Abnormal packet list table with Timestamp
            end
        else
            drop.Packet_bst
            update. Abnormal packet list table with Timestamp
        end
    end
end
```

TSCBA's Broadcast Alert Message Processing

Algorithm TSCBA's Unicast Alert Message Processing

```
for i → 1 to n do
    clear.ARP_Cache(20 minutes)
    Decode.Packet_bst if opcode == 0x0004 then
        if SourceEth_MAC == SourceArp_MAC) and
        (DestinationEth_MAC == DestinationArp_MAC then
            if SourceArp_IP andSourceArp_MAC ==
            IP − MAC pair in cache then
                Extract TimeStamp
                if (TS_t - TS_g) ≤ 10 then
                    Generate new unicast Reply with new TS_g and send it
                else
                    drop.Packet_ust
                    Generate Unicast Alert message to the source about elapsed
                    time
                end
            else
                drop.Packet_ust update. Abnormal packet list table with
                Timestamp Generate Broadcast Alert message
            end
        else
            drop.Packet_ust update. Abnormal packet list table with Timestamp
            Generate Broadcast Alert message
        end
    end
end
```

TSCBA's Unicast Alert Message Processing

Though the TSCBA mitigation technique effectively performs the attack prevention, it still has the following snags: NULL MAC addresses, unused MAC addresses, Multicast addresses can be detected before cross-layer inspection which may reduce the computational time and cost involved in Data table Scanning.

4.3 Timestamp and counter based approach in large data centre network (TCDCN)

TCDCN is an extended approach for preventing ARP cache poisoning attacks in large data centre networks(LDCN). It focusses to avoid broadcast storms caused by ARP in large data centre networks. In LDCN, to avoid storms, it is divided into more number of smaller layer two networks each managed by Top of the Rack (ToR) switches handle their own broadcast traffic. TCDCN concentrates on avoiding host migration issues which more common in LDCN, related to ARP attacks. TCDCN further modifies the TSCBA's ARP cache by adding an entry named C_{req} to keep track number of requests on same cache entry is made. In addition to cross-layer checking, abnormal packets are checking; the timestamp is validated for each time to ensure statefulness and authenticity of ARP. The modified ARP cache and invalid list table are depicted in Figs. 9 and 10 respectively. The invalid list table is updated once an abnormal or malicious packet is detected.

TCDCN uses three alert messages; Unicast invalidate message to alert the gateway when a host migrates the network, second one the broadcast invalidate message to all the other hosts to invalidate the migrated hosts and the third one the broadcast vigilant message to alert about invalid entry found in the invalid list. The vigilant message is

Type	IP Address	MAC Address	Interface	TS_{up}(sec)	C_{req}
Static/ dynamic	172.168.0.1	00:05:79:66:68:01	Fast Ethernet 0/1	2016-07-14 04:32:26	3

Fig. 9 TCDCN's ARP cache

Index	IP Address	MAC Address	Count	TS_{lup} (sec)
1	198.164.0.3	00:34:44:56:22:34	1	2016-08-14 03:22:26
2	165.178.0.5	00:98:98:76:34:56	10	2016-08-14 02:12:56
TS_{lup}← last updated timestamp; Count← of times request for this entry was made				

Fig. 10 TCDCN's invalid list table

generated whenever the count of an invalid entry exists more than three times. The flowchart of TCDCN is depicted in Fig. 11.

TCDCN Algorithm:

Assumptions:

- Consider a Host A wants to communicate with Host B in its subnet, but it does not have an entry in ARP cache.
- A and B are on the same subnet
- ARP table tuple: ARP Type, IP addr, MAC Addr, Interface, TSup,Creq
- A knows the IP of B but not it's MAC

Algorithm : Generating ARP Request

if IP_B *Known and* MAC_B *Unknown* **then**
  GEN: $ARP_{REQ} \leftarrow OP_{REQ}, IP_A, MAC_A, IP_B, BDC_{MAC}, TS_{qg}, MSG$
  Set the following variables in ARP_{REQ} *of A:*
  $OP_{REQ} \leftarrow 1$
  $IP_{TAR} \leftarrow IP_B$
  $MAC_{TAR} \leftarrow BDC_{MAC}$
  $IP_{SRC} \leftarrow IP_A$
  $MAC_{SRC} \leftarrow MAC_A$
  $IP \leftarrow IP_B$ and MAC ← NULL in ARP_A **Send** ARP_{REQ}
else
  /**IP_B-MAC_B is in ARP_A **/
  $SEND_{PKT}$;
end

1TCDCN's ARP Request Generation

TCDCN mitigation technique effectively performs the attack prevention by moving NULL MAC addresses, available MAC addresses, Multicast addresses detection before cross-layer inspection, thus reducing the computational time and cost involved in Data tables Scanning. However, it may still incur some considerable cost in maintenance. TCDCN can detect the ARP-based DoS, MiTM, Cloning and host migration issues. The nomenclature of TCDCN algorithm is specified in Fig. 12.

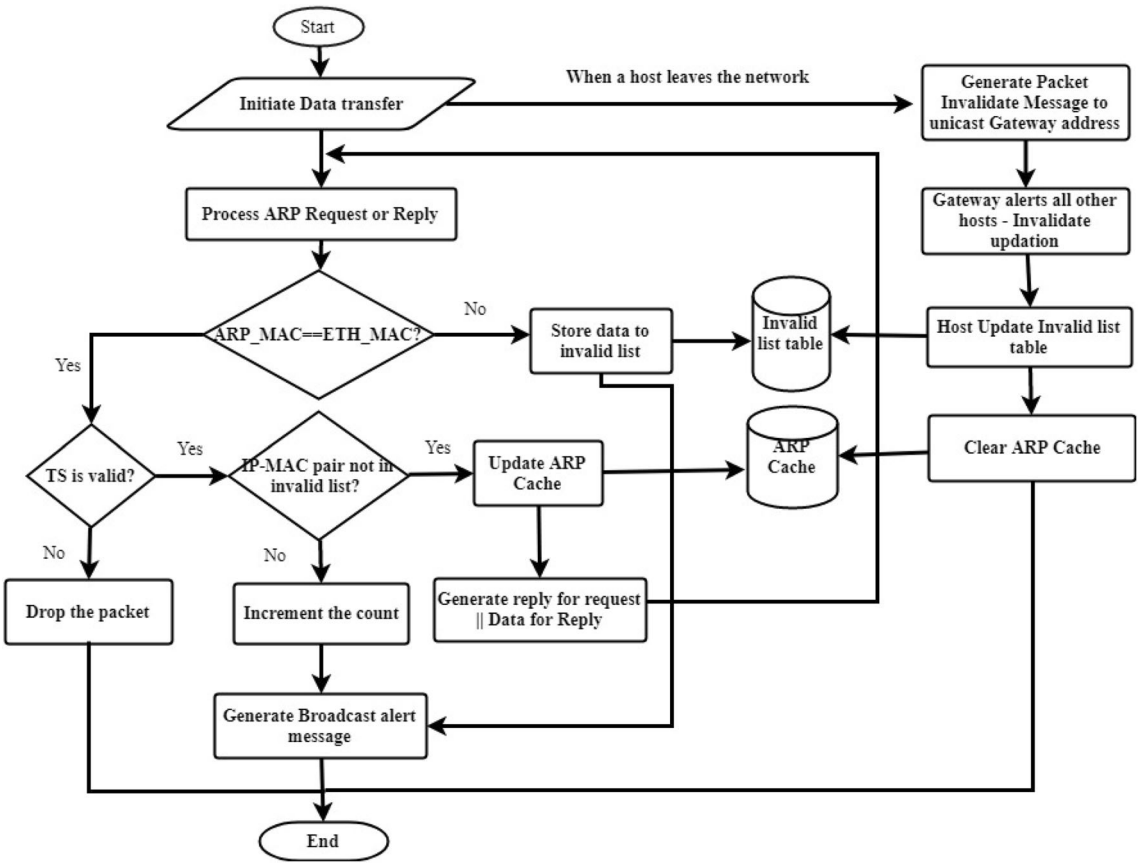

Fig. 11 TCDCN work flow

Algorithm Processing ARP Requests

Assumptions

1. Host B has received the ARP_{REQ} from host A.
2. $OP_{REQ} \leftarrow 1$

For all ARP type packets
if $MAC_{TAR} == BDC_{MAC}$ **then**
  if $ARP_{MAC} == ETH_{MAC}$ **then**
    if IP_A *and* MAC_A *not in* INV_{LISTB} **then**
      if $IP_A - MAC_A$ *is in* ARP_B **then**
        if $(CLK - ARP_{REQ}.TS_{pg}) \leq 10s$ **then**
          if $(ARP_B.C_{req}) \leq 4$ **then**
            GEN: ARP_{REP}
            SET: $ARP_B.C_{req} \leftarrow ARP_B.C_{req} + 1$
          else
            Discard ARP_{REQ}
          end
        else
          $SEND_{ACK}$ to A
        end
      else
        ADD IP_A and MAC_A ,
        $TS_{up} \leftarrow TS_{qg} in ARP_B$
        $C_{req} = 1$
      end
    end
    Update IP_A-MAC_A: $TS_{lup} \rightarrow TS_{qg}$ in INV_{ListB} **if** $INV_{LISTB}.Count \geq 3$ **then**
      Discard ARP_{REQ} GEN: BDC_{VIG} SEND: BDC_{VIG}
    else
      INV_{LISTB}.Count+=1
    end
  end
  ADD: ARP_{REQ} in INV_{LISTB}
end
Discard ARP_{REQ} and ADD ARP_{REQ} in INV_{LISTB}

TCDCN's ARP Request Processing

Algorithm Processing ARP Replies

Assumption:

1. ARP_{REQ} from A is valid and it has successfully processed all the checks in processing ARP requests
2. ARP_B is updated with host A details

1. Generating ARP Reply GEN: ARP_{REP} with following tuple IP_B,MAC_B,IP_A,MAC_A,TS_{qg} **SET:** $OP_{REP} \leftarrow 2$, $IP_{TAR} \leftarrow IP_A$, $IP_{SRC} \leftarrow IP_B$, $MAC_{TAR} \leftarrow MAC_A$, $MAC_{SRC} \leftarrow MAC_B$, $TS_{qg} \leftarrow$CLK **2. Processing ARP Reply**
Assumption:

1. Host A received ARP_{REP} from B
2. $OP_{REP} \rightarrow 2$

if $MAC_{TAR} == UNI_{MAC}$ **then**
  if $ARP_{MAC} == ETH_{MAC}$ **then**
    if $IP_B and MAC_B not in INV_{LISTA}$ **then**
      if $IP_B and MAC_B \rightarrow NULL in ARP_A$ **then**
        if $CLK - ARP_{REP}.TS_{pg} \leq 10s$ **then**
          SET: $ARP_A.MAC_B \leftarrow ARP_{REP}.MAC_{SRC}$
          $ARP_A.TS_{up} \leftarrow TS_{pg}$
        else
          Discard ARP_{REP}
          GEN: UNI_{ALT} to A
        end
      else
        ADD IP_B-MAC_B pair with $TS_{lup} \leftarrow TS_{pg}$ in INV_{ListA}
        Discard ARP_{REP}
        GEN: BDC_{VIG}
      end
    else
      if $INV_{LISTA}.Count \geq 3$ **then**
        SEND BDC_{VIG}
      else
        INV_{LISTA}.Count+=1
      end
    end
  else
    ADD: ARP_{REP} in INV_{LISTA}
  end
else
  Discard ARP_{REP} and ADD ARP_{REP} in INV_{LISTB}
end

2TCDCN's ARP Reply Processing

5 Results and discussion

The three mitigation techniques CLCC (Prabadevi and Jeyanthi 2018), TSCBA (Prabadevi and Jeyanthi 2018), TCDCN Prabadevi and Jeyanthi 2017a, b) stated above are compared with each other, and it is summarized in Table 2. Of these, TCDCN outperforms in all aspects.

5.1 Performance evaluation

All the three proposed techniques are implemented using C# and Dot NET. The experimental setup of the three techniques are specified in the Table 2.

The performance of the three system are evaluated by detection of following types of abnormal packets.

P#1. IP_{IN},MAC_{VL} in the source host of ARP Request Message

P#2. IP_{IN},MAC_{VL} in the destination host of ARP Request Message(MAC is a broadcast)

P#3. IP_{VL},MAC_{IN} in the source host of ARP Request Message

P#4. IP_{VL},MAC_{IN} in the destination host of ARP Request Message (here MAC can be Multicast, unicast or null address instead broadcast)

P#5. IP_{IN},MAC_{VL} in the source host of ARP Reply Message

P#6. IP_{IN},MAC_{VL} in the destination host of ARP Reply Message

P#7. IP_{VL},MAC_{IN} in the source host of ARP Reply Message

P#8. IP_{VL},MAC_{IN} in the destination host of ARP Reply Message (here MAC can be broad cast, Multicast or null address instead unicast or it might be a wrong MAC address)

P#9. IP_{IN},MAC_{VL} in the source host of Unicast Alert Message

P#10. IP_{IN},MAC_{VL} in the destination host of Unicast Alert Message

Symbols/Notation	Description
ARP_{REQ}, ARP_{REP}	ARP Request and Reply Packet
OP_{REQ} , OP_{REP}	ARP Request and Reply opcodes, takes a value 1and 2 respectively
ARP_{MAC}, ETH_{MAC}	MAC address in ARP header and Ethernet header
IP_{SRC}, IP_{TAR}	IP address of Source and destination host
IP_A, IP_B	IP Address of host A and B
MAC_{SRC}, MAC_{TAR}	MAC Address of Source and Destination host
MAC_A, MAC_B	MAC Address of host A and B
BDC_{IP} , BDC_{MAC}	Broadcast IP and MAC address takes the values 255.255.255.255 and ff:ff:ff:ff:ff:ff
MUL_{IP}, MUL_{MAC}	Multicast IP and MAC addresses
UNI_{MAC}	Unicast MAC address
ARP_A, ARP_B	ARP tables of host A and B
$SEND_{ACK}$	Unicast message to A about TS expiry
$SEND_{PKT}$	Start data communication
TS	Timestamp
INV_{ListA},INV_{ListB}	Invalid list table of host A and host B
BDC_{VIG}	Broadcast Vigilant message
UNI_{ALT}	Unicast Alert Message
CLK	System clock time
$NULL_{IP}$, $NULL_{MAC}$	NULL IP address and NULL MAC address
IP_{VAL},MAC_{VAL}	Valid IP and MAC address

Fig. 12 TCDCN algorithm nomenclature

P#11. IP_{VL},MAC_{IN} in the source host of unicast Alert Message (here MAC can be broad cast, Multicast or null address instead unicast or it might be a wrong MAC address)
P#12. IP_{VL},MAC_{IN} in the destination host of unicast Alert Message (here MAC can be broad cast, Multicast or null address instead unicast or it might be a wrong MAC address)
P#13. IP_{VL},MAC_{IN} in the source host of Broadcast Alert Message
P#14. IP_{IN},MAC_{VL} in the source host of Broadcast Alert Message
P#15. IP_{IN},MAC_{VL} in the destination host of Broadcast Alert Message (IP and MAC should be a broadcast message)
P#16. IP_{VL},MAC_{IN} in the source or destination host of gratuitous ARP request Message
P#17. IP_{VL},MAC_{IN} in the source or destination host of gratuitous ARP reply Message
P#18. Source MAC and Destination MAC does not match in gratuitous ARP Request reply messages
P#19. Invalid ARP Request from Migrated host
P#20. Invalid ARP Reply from Migrated Host

Here the broadcast alert messages covers all types of broadcast Alert messages used in all the three methods proposed.

The Malicious Packet Detection Ratio is calculated using the formulae given in Eq. 1:

$$MPDR\ (\%) = \frac{Number\ of\ Malicious\ Packets\ Captured}{Total\ Number\ of\ Malicious\ Packets\ Injected} \times 100 \quad (1)$$

$$Number\ of\ Malicious\ Packets\ Captured = \sum_{i=0}^{20} Pi \quad (2)$$

where *Pi* is the Malicious or abnormal packets described above

The MPDR(%) is given in the Table 3 and the graphical representation is specified in Fig. 13. Although CLCC method showed 77% of detection ratio individually when compared with existing solutions, when it has been evaluated with TSCBA and TCDCN, its performance is less, and TCDCN proves to be the best as depicted.

The attack types captured by these techniques include IP spoofing, MAC spoofing, MiTM, Host impersonation/ Cloning attack, DDOS and Bombing packet attack. The abnormal packets responsible for these types of attacks are given in the Table 4. and the attack detection ratio is given in Table 5 and Fig. 14.

5.2 Effects of ARP cache poisoning attacks on medical computing

In this digital era, computing plays a vital role in all fields of government sector including digital marketing, e-governance, e-health, digi-bank services, e-agriculture, e-fisheries, e-learning, e-seva, e-land resources and so. Of these, the most crucial sector is health care wherein the medical records of the patients are maintained digitally for globalization. Now we are moving towards an Expert system of medicine where virtual medication, virtual diagnosis and technology in all the aspects of medication. Shortly we may be at this level of digitization where, when we specify our Adhaar number (or any other similar unique identifier) will retrieve entire health record anywhere in the world. Through these electronic health details the experts can prescribe medicines, refer patients admission in hospitals, all test records will be linked automatically to the electronic record of patients. So this paves the way for enhanced security in medical records transaction. The type of medical records that are maintained digitally includes Lab test reports (like blood tests reports, glucose test), Scan reports(CT scan, MRI, Endoscopy), Genetic disorders tracking through Stem cells therapy, Pregnancy cases history and its complications, Prescription details. Telemedicine has grown to the extent where kiosks in public places help the patients to undergo a series of questions for diagnosis of diseases in emergency cases. Doctors operate remotely on patients through the concept of virtual medicine where the patients can order, buy their medicines

Table 2 CLCC versus TSCBA versus TCDCN

Features	CLCC	TSCBA	TCDCN	Comments
Cross-Layer checking	YES	YES	YES	TCDCN outperforms others by validating MAC address before cross Layer check
Invalid MAC addresses Validated (Multicast MAC, Broadcast MAC)	NO	Partial	YES	TCDCN cross checks the MAC address and IP address before performing consistency check
Gratuitous ARP packet related cache poisoning	NO	YES	YES	TCDCN algorithm does not shows it explicitly but it shows only additional features added from TSCBA
Data tables used	ARP cache and Fake list table	TSCBA's ARP cache and Abnormal packet list table	TCDCN's ARP cache and Invalid list table	Though the data tables used are similar, the fields are different from one to another
New field in modified ARP Cache	NIL	Timestamp	Timestamp and count of Requests made	CLCC does not modifies the RFC826 ARP Cache
New ARP Messages introduced	Broadcast Alert Message and Unicast Alert	Timestamped Broadcast Alert and Unicast Alert	Timestamped broadcast Invalidate, broadcast Vigilant and Unicast Invalidate	Alert messages are used to avoid attacks from same attacker in all the techniques
Host Migration	No	No	YES	TSCBA is specifically designed to mitigate issues in ARP with host migration
No of Packets injected	1255	1250	1100	The type of packets injected vary from one system to another
% of detection when compared with existing techniques	77	83	85	Detection ratio is also calculated based on the attack packets injected
Processes that strengthen RFC826	Cross layer checking and Alert Messages	Consistency check, Timestamp generation, counters in abnormal list and alert messages	Timestamped alert messages, Counters in ARP table and host migration issues	Because of cross layer check the ARP is strengthened but requires time
ARP Storm detection	Not done	Partial	YES	CLCC does not perform any storm
Attacks detected	DoS, MiTM, MAC spoofing	DDoS, MiTM, Bombing attack, Cloning attack and IP and MAC spoofing	DDoS, MiTM, replay attack, Cloning attack and IP and MAC spoofing	Basic level attacks will be detected by CLCC
Experimental setup	Three host within a same LAN	Three host within a same LAN	Six hosts, three in one LAN and Three in another LAN	Robustness of the algorithm is proved by number of packets injected

Table 3 Malicious packet detection ratio

Detection methods	No of packets received	No of malicious packets injected	No of malicious packets detected	Detection ratio (%)
ARP	750	500	95	19
CLCC	750	500	244	49
TSCBA	750	500	414	83
TCDCN	750	500	446	89

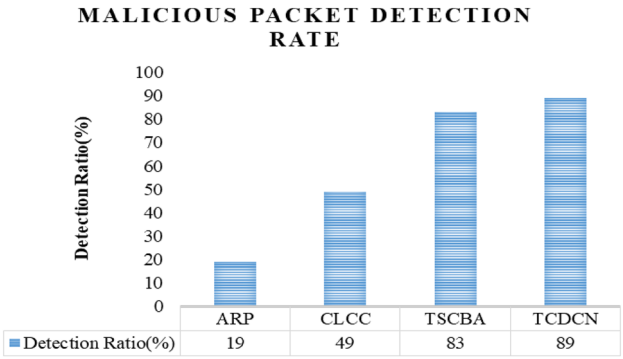

Fig. 13 Performance graph of ARP vs mitigation techniques

online. Also, surgeries under Telepresence has become common nowadays, by which the surgeons can operate remotely without their physical presence or through aided robots. Henceforth, security in the transaction of medical imaging, records is paramount. With the vulnerable protocol ARP yet simple, these details on servers can be hacked when the system is compromised via ARP cache poisoning will create a devasting effect on any well –reputed hospital's data, as health data are more important than bank data.

Privacy issues in medical field include Globalization of patient's details (like a bank account, address, phone number), Unfortunate email with sensitive information to wrong recipients, loading numerous patient's record on a public server and so (Bidgoli 2006). When this kind of public servers are hacked by ARP cache poisoning any of the following may take place:

1. Devastating the entire network by any of the aforementioned attack types like DDoS.
2. Silently altering the medical record of an innocent to facilitate criminal offences like unauthorized Artificial insemination, organ theft, inducing deadliest diseases (HIV), etc.,
3. The mistakenly altered medical record may lead the physician to diagnose correctly for the incorrect patient by prescribing correct medicines leading to harmful effects on the innocent patient.
4. An attacker may use this opportunity to wantonly alter the medical record of any VIP to avoid the disclosure of actual reasons from the public.
5. Hospitals can misuse by ceasing the medical research records of other hospitals
6. Adulteration in the drugs can be made by hacking the medical composition of drugs utilised by a successful and renowned doctors' patent rights
7. Cyberwar by hacking a country medical research history and doctor's sensitive research foundation.

The mitigation techniques proposed will help to avoid these issues by avoiding ARP cache poisoning based medical theft.

6 Conclusion

The telecommunication protocol ARP that plays an essential role in all sectors of digital computing have been discussed. The various solutions for mitigating ARP cache poisoning attacks have been tabulated. Three new techniques for mitigating ARP Cache poisoning attacks viz., CLCC, TSCBA and TCDCN were recommended. The working mechanism of these techniques was elaborated, compared and evaluated using a set of malicious packets. Although these algorithms individually outperform other existing solutions, of these three TCDCN, performs better serving 89% of Malicious packet detection and $\sim$ 95% of ARP cache poisoning attack prevention. Also, effect of ARP cache poisoning in trending medical computing was discussed. The privacy issues concerned with Medical records and problems that may arise if these medical records were ARP cache poisoned was presented. The

Table 4 Attack type packets captured

S. no.	Attack packets captured	Attack types
1	P3, P4, P7, P8, P11, P12, P13, P15, P16, P17, P18, P19, P20	DDoS
2	P1, P2, P3, P4, P5, P6, P7, P8, P16, P17, P19, P20	Host impersonation
3	P3, P4, P7, P8, P11, P12, P15, P16, P17, P19, P20	MiTM
4	P3, P4, P7, P8, P11, P12, P13, P16, P17, P18, P19, P20	MAC spoofing
5	P1, P2, P5, P6, P9, P10, P14, P15, P19, P20	IP spoofing
6	P1, P2, P3, P4, P5, P6, P7, P8, P16, P17, P18, P19, P20	Bombing packet attack
7	P3, P4, P11, P12, P13, P16, P18, P19, P20	Broadcast storms

Table 5 Attack detection ratio

Attack types	Detection ratio (%)		
	CLCC	TSCBA	TCDCN
DDoS	76	80	95.08
Host impersonation	46.67	79.33	95.33
MiTM	74.91	78.18	95.64
MAC spoofing	75	79	95.67
IP spoofing	11.2	74.4	93.6
Bombing packet attack	48.92	80.31	95.38
Broadcast storms	69.33	72.89	94.67

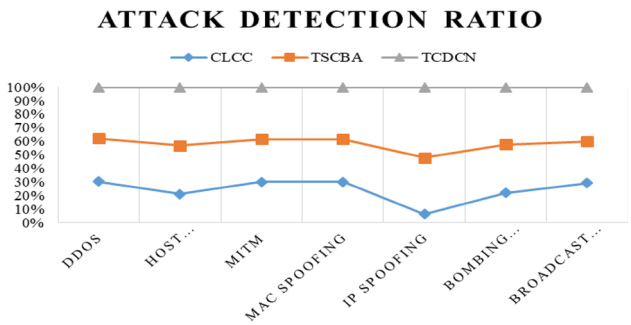

Fig. 14 Graphical representation of attack detection ratio of mitigation techniques

further study is to demonstrate ARP cache poisoning mitigation techniques on any health centres.

References

Al-Hemairy M, Amin S, Trabelsi Z (2009) Towards more sophisticated ARP spoofing detection/prevention systems in LAN networks. In: 2009 international conference on the current trends in information technology (CTIT), IEEE, pp 1–6

Bidgoli H (2006) Handbook of information security, information warfare, social, legal, and international issues and security foundations, vol 2. Wiley, Hoboken

Bradley T, Brown C, Malis A (1998) Inverse address resolution protocol (No. RFC 2390)

Bruschi D, Ornaghi A, Rosti E (2003) S-ARP: a secure address resolution protocol. In: Proceedings. 19th annual computer security applications conference, 2003, IEEE, pp 66–74

Cheshire S (2008) IPv4 Address conflict detection, RFC 4227

Dangol S, Selvakumar S, Brindha M (2011) Genuine arp (garp): a broadcast based stateful authentication protocol. ACM SIGSOFT Softw Eng Notes 36(4):1–10

Hammouda S, Trabelsi Z (2009) An enhanced secure ARP protocol and LAN switch for preveting ARP based attacks. In: Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly, ACM, pp 942–946

Hingne A, Jain S (2016) A survey on various detection and prevention mechanism for MITM and ARP attacks. Int J Innov Res Comput Commun Eng 4(11):19918–19924

Issac B (2009) Secure ARP and secure DHCP protocols to mitigate security attacks. Int J Netw Secur 8(2):107–118

Jana I (2017) Effect of ARP poisoning attacks on modern operating systems. Inf Secur J A Global Perspect 26(1):1–6

Kaur J, Dhanda SK (2014) An analysis of local area network ARP spoofing. Int J Latest Trends in Eng Technol 4(3):117–123

Lootah W, Enck W, McDaniel P (2007) TARP: ticket-based address resolution protocol. Comput Netw 51(15):4322–4337

Pandey P (2013) Prevention of ARP spoofing: a probe packet based technique. In: 2013 IEEE 3rd international advance computing conference (IACC), IEEE, pp 147–153

Plummer DC (1982) An ethernet address resolution protocol-converting network protocol to 48 bit ethernet address for transmission on ethernet hardware. RFC-826

Prabadevi B, Jeyanthi N (2014) Distributed denial of service attacks and its effects on cloud environment-a survey. In: The 2014 international symposium on networks, computers and communications, IEEE, pp 1–5

Prabadevi B, Jeyanthi N (2017a) Security solution for ARP cache poisoning attacks in large data centre networks. Cybern Inf Technol 17(4):69–86

Prabadevi B, Jeyanthi N (2017) A mitigation system for ARP cache poisoning attacks. In: Proceedings of the second international conference on internet of things and cloud computing, ACM, p 20

Prabadevi B, Jeyanthi N (2018) A framework to mitigate ARP sniffing attacks by cache poisoning. Int J Adv Intell Paradig 10(1–2):146–159

Puangpronpitag S, Masusai N (2009) An efficient and feasible solution to ARP Spoof problem. In: 6th international conference on electrical engineering/electronics, computer, telecommunications and information technology, 2009. ECTI-CON 2009, IEEE, vol 2, pp 910–913

Quarterman JS (1987) RFC 1027—using ARP to implement transparent subnet gateways. Request for Comments, 'Online, pp 1–7

Rupal DR, Satasiya D, Kumar H, Agrawal A (2016) Detection and prevention of ARP poisoning in dynamic IP configuration. In: IEEE international conference on recent trends in electronics, information & communication technology (RTEICT), IEEE, pp 1240–1244

Samvedi A, Owlak S, Chaurasia VK (2014) Improved secure address resolution protocol. arXiv preprint arXiv:1406.2930

Saputro N, Akkaya K (2015) PARP-S: a secure piggybacking-based ARP for IEEE 802.11 s-based Smart Grid AMI networks. Comput Commun 58:16–28

Song MS, Lee JD, Jeong YS, Jeong HY, Park JH (2014) DS-ARP: a new detection scheme for ARP spoofing attacks based on routing trace for ubiquitous environments. Sci World J 2014:1–8. https://doi.org/10.1155/2014/264654

Trabelsi Z (2016) The robustness of microsoft windows and apple mac OS X against ARP cache poisoning based network attacks. In: 2016 13th IEEE annual consumer communications & networking conference (CCNC), IEEE, pp 1074–1079

Trabelsi Z, El-Hajj W (2010) On investigating ARP spoofing security solutions. Int J Internet Protoc Technol 5(1–2):92–100