BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **18**, No 4
Sofia • 2018 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2018-0049

TSCBA-A Mitigation System for ARP Cache Poisoning Attacks

B. Prabadevi, N. Jeyanthi

School of Information Technology and Engineering, VIT University, Vellore, India
E-mails: prabadevi.b@vit.ac.in njeyanthi@vit.ac.in

***Abstract*:** *Address Resolution Protocol* (*ARP*) *cache poisoning results in numerous attacks. A novel mitigation system for ARP cache poisoning presented here avoids ARP cache poisoning attacks by introducing timestamps and counters in the ARP messages and ARP data tables. The system is evaluated based on criteria specified by the researchers and abnormal packets.*

***Keywords*:** *ARP cache poisoning, TimeStamp, MiTM, DDoS, Bombing packet attack.*

1. Introduction

The Address Resolution Protocol (ARP) by Plumber is the most prominent one for any host in a network to communicate with other hosts [1]. All hosts maintain ARP cache in the network holding the IP-MAC pair of the other hosts. So whenever a host wants to communicate with other hosts, it hunts the data from ARP cache. ARP helps the host to retrieve the MAC address for a given IP address. Though the protocol is the most prominent, it is not secured. It is prone to more number of vulnerabilities because of its stateless nature. There are two types of ARP cache entries which co-exist viz., Static and Dynamic [2, 3]. The static entry of ARP table is secure compared to dynamic entries, but it incurs colossal maintenance cost. Statically entered entries are not removed from the ARP table till the next boot but still depend on the operating system. Dynamic entries are cleared as per the network setup. As most of the evolving applications are distributed and dynamic, they prefer dynamic entries for their network.

A noteworthy feature of ARP is that it is authentication free protocol which replies to any ARP requests without validating the packets received. This is the most protuberant reason for most of the attacks on the network through ARP. ARP cache poisoning is when an attacker or an impersonator (a malevolent host) sends spoofed or forged ARP request-reply messages to the victim in the network. This act makes a fake entry in the ARP table and poisons the table. This is also called ARP spoofing.

Some of the attacks are Sniffing, ARP Spoofing, Man-in-the-Middle attack, cloning, host impersonation, connection hijack, DoS and its variants [4]. These DoS attacks and their variants have tremendous effects on cloud [5] and different solutions

coexist for combatting these attacks [6, 7] on large data centres. This paper provides a framework for mitigating this Cache poisoning through an efficient mechanism. The ARP Request and reply scenario are depicted in Figs 1 and 2, respectively.

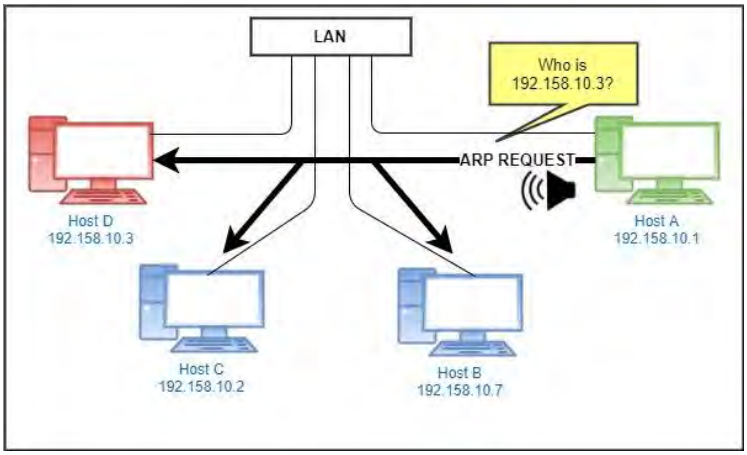


Fig. 1. ARP request scenario

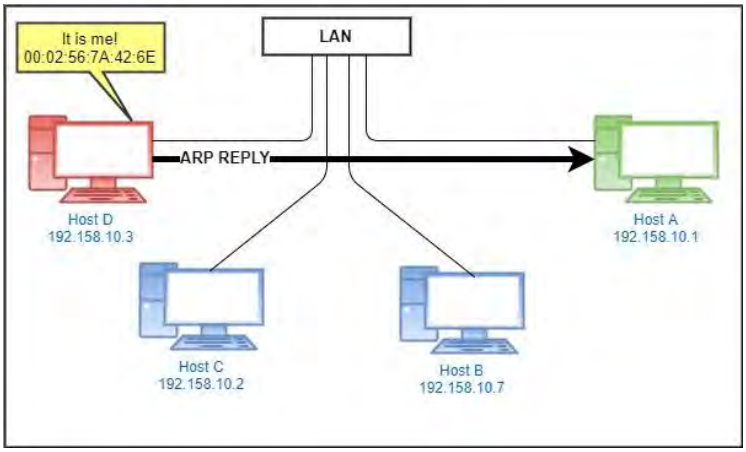


Fig. 2. ARP response scenario

2. Literature survey

Recent researchers have proposed many techniques to mitigate ARP poisoning attacks in various networks. Table 1 provides details about various mitigation techniques available in the literature. The various mitigation techniques mentioned in Table 1 include centralised server based, cryptographic, non-cryptographic techniques, IDS, extensions to ARP protocol and resolved solutions by improvising the static as well as dynamic entries made. However, ARP remains unsecured. The proposed work mentioned in section III deals with non-cryptographic, timestamp and counter based approach for ARP cache poisoning.

Table 1. ARP mitigation techniques

Mitigation method	Feature	Type of network and ARP entry
A kind of Client Server Protocol which automatically configures the static ARP entries [8]	**Detected and Prevented** ARP Spoofing attack with fewer loads	MANET, Static and DHCP
Uses ICMP protocol. By making use of packet sniffers, it separates the valid packets and invalid packets and performs the different level of detection of two varieties using ICMP echo packets [9]	It **detects** the IP-MAC pair of the legitimate host and malicious hosts during the attack	LAN, Static and Dynamic
A New method to be deployed on IDS was proposed. It performs cross-layer consistency checking, identifies invalid IP-MAC entries by building static and dynamic entries and detects ARP message spoofing [10]	ARP-based MiM and DoS attacks can be **prevented**. They had suggested six basic requirements to be met by any security mechanisms to detect ARP spoofing	Switched LAN
A trusted Centralized server which maintains the IP-MAC pair of all hosts in the network both legitimate and illegitimate hosts. Every host updates its information to CS during DHCP and updates malicious host with its IP by broadcasting an ARP packet with its IP [11]	May **prevent** lower and higher level layer attacks	LAN, supports both Static and DHCP
A client Server based automatic and Scalable Static ARP entries with more scalability feature enhanced [12]	ARP Spoofing Attack will be **prevented**. Experiment results proved with fewer authentications in less time	LAN, supports both Static and DHCP
IDS- Host-based which maintains four different tables for ARP-Request, ARP-reply, IP-MAC verification, IP-MAC Binding table and six different algorithms to perform various authentication, verification and identification of spoofed addresses [13]	It had been checked with different attack scenarios and detected host impersonation attack, ARP spoofing attack	LAN, Static and Dynamic
Two methods have been used. One is to poison the attacker ARP cache by sending ARP reply as reply to ARP reply received and validates legitimate user by sending IP probe, whereas the other method uses CAM table [14]	**Prevents** MiTM and DoS attacks. It is an extension to ARP	Switched Ethernet LAN
Trabelsi and El-Hajj [15] provided an experimentation analysis on various security mechanisms and detailed categorisation of detection vs. prevention mechanisms. They proposed an algorithm that meets all the requirements stated by Al-Hemairy	**Detects** all spoofing attacks	LAN, Static and Dynamic
Probe-based E-SDE categorises the attacker into three types based on their spoofing capability and detects them with the help of verifying table and handler algorithm [16]	Uses ARP–ICMP probe packets, **detects** the ARP spoofing and identifies the legitimate IP-MAC pair	LAN
A centralized mechanism ACS is used to manage all the ARP Entries in the hosts. All clients update their information to the ACS which is maintained in long-term secondary ARP cache along with static ARP table so that it makes a double check with these tables for each new ARP messages. ACS is protected using Antidote Scheme [17]	**Detects** MiTM attacks. If ACS is attacked, then all the activity will fail	LAN, Static

Table 1 (c o n t i n u e d)

Mitigation method	Feature	Type of network and ARP entry
A centralised server mechanism which purely depends on the CS polling score elected and remains other hosts and ICMP messages [18]	**Detects and prevents** ARP poisoning attack	LAN, Static
A Secondary cache mechanism using ICMP messages. The secondary cache mentioned is a text file thus alleviating the cost [19]	Solves IP Exhaustion Problem	LAN, Static

3. Timestamp and counter based approach for ARP cache poisoning

The proposed system TimeStamp and Counter Based Approach (TSCBA) for ARP cache poisoning attack is depicted in Fig. 3. The input to the system is ARP Request or ARP Reply packets received by any host. The expected outcomes are abnormal packet list tables, Timestamped ARP request / Reply packets.

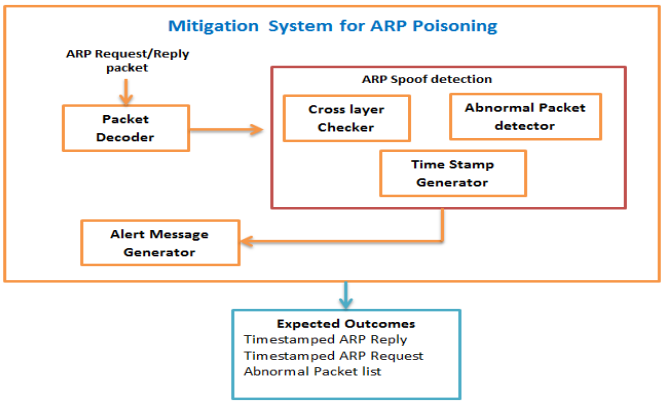


Fig. 3. TSCBA working components

3.1. Packet Analyzer/Decoder

Packet Analyzer/Decoder performs the pre-processing activities required to retrieve the contents of Ethernet Header and ARP header. Packet Analyser will filter the ARP Packets for making it available for cross-layer checking. The ARP packets such as ARP Request, ARP Reply, Unicast Alert message and Broadcast Alert message will be filtered.

3.2. Cross-Layer checker

Cross-Layer checker specified in Fig. 4 inspects whether the Ethernet header MAC address and ARP header MAC address are consistent with each other. If a match is found, then it is sent for the next level of checking. Otherwise, the packet is added to abnormal packets list. The cross-layer checker may also ensure whether the type of addressing for ARP and Ethernet pair are same or different, because this may lead to an anomaly.

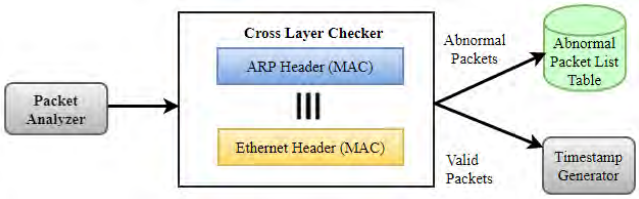


Fig. 4. Cross-Layer checker

3.3. Abnormal Packet detector

Abnormal Packet detector depicted in Fig. 5 works based on the above inspection. Based on the consistency results, it updates the abnormal packet list table. A packet is detected as abnormal if it satisfies the below conditions:

- IP-MAC pair is not found in the ARP Table,
- IP-MAC pair found, but timestamp is incorrect,
- In ARP Reply packet received by any host if Source IP is Multicast or Broadcast and the cross-layer consistency is not met, i.e., there is an unexpected IP – MAC address in the packets.
- If IP-MAC pair is in Abnormal packet list table.

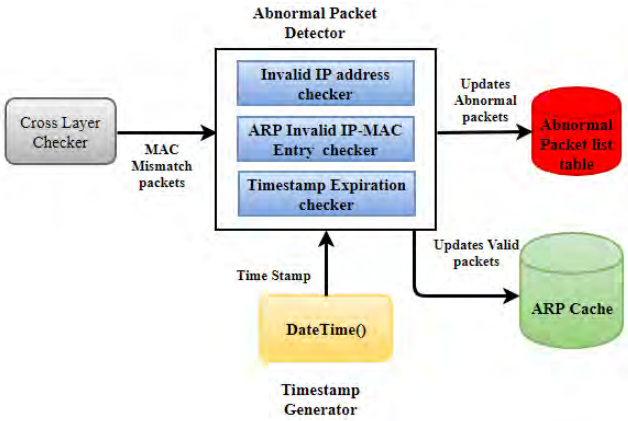


Fig. 5. Abnormal packet detector and TimeStamp generator

3.4. TimeStamp generator

This component generates a timestamp and appends it to ARP Request, Reply messages and unicast/Broadcast Alert messages. The timestamp field has two parts viz., the time TS_g at which the ARP has replied, Request and broadcast Alert messages are generated and the time TS_t, is the time till which these messages are valid. The threshold for this timestamp depends on network latency and delay. A Sample timestamp is depicted in Fig. 6.

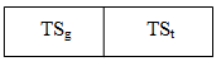


Fig. 6. TimeStamp field

3.5. Alert message generator

The Alert message generator depicted in Fig. 7 generates a broadcast alert message and updates the abnormal packet list table when an abnormal packet is detected. The sample broadcast and unicast alert messages are depicted in Fig. 8 and Fig. 9, respectively.

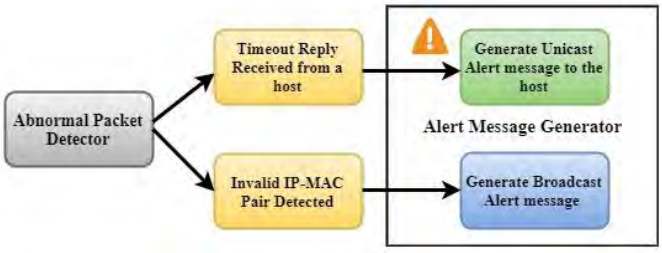


Fig. 7. Alert message generator

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. 8. Broadcast alert message

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

Fig. 9. Unicast alert message

The system clears the ARP cache at every 20 minutes (can be changed based on the frequency at which a new node joins the network). As suggested by [7], the six fundamental requirements which should be satisfied by any ARP mitigation system in LAN networks have been met by this system and they are:

1. Cross-layer inspection is carried out by the second component.
2. ARP spoof detection is performed by components 2, 3 and 4, which make the ARP stateful.
3. Unexpected IP-MAC detection by component 3.
4. Building manual and dynamic ARP table.
5. The 4th component can avoid ARP storm.

ARP scanning can be done using tools, which are not covered by this system. The algorithm for the proposed system is given below.

3.6. **Algorithm 1.** ARP mitigation technique

Assumptions:

i. A network with *n* nodes.
ii. ARP cache is cleared at every 20 minutes.
iii. $\mathbf{TS_t = TS_g + 10\ s}$ (it may vary based on *n* and network latency).

Table 2. Nomenclature of TSCBA

Eth_MAC	MAC address in Ethernet Header
arp_IP	IP address in ARP Header
arp_MAC	MAC address in ARP Header
TS_g	Timestamp generation time
TS_t	Timestamp validity time
N	No of nodes in network1 to the maximum capacity of the LAN
$Packet_{req}$	ARP Request Packet
$Packet_{rep}$	ARP Reply Packet
$Packet_{bst}$	ARP Broadcast Alert Packet
$Packet_{ust}$	ARP Unicast Alert Packet

The proposed system has 2 data tables namely: Traditional ARP cache with an added entry viz., timestamp update TS_{up} which is the system time and Abnormal Packet list table. The nomenclature of TSCBA is given in Table 2. The contents of these two tables are specified in Table 3 and Table 4.

Table 3. Modified ARP cache

Protocol	IP Address	MAC Address	ARP Type	Interface	TS_{up}(time)
Internet IP/TCP	172.168.0.1	00:50.79:66.68: 01	ARPA	FastEthernet 0/1	2016-07-14 04:32:26

A new entry has been added, which may help to determine the lifetime of the host in the network and it as well helps in the setup of the cache clearing time.

Table 4. Abnormal packets list table

Index	IP Address	MAC Address	Count	TS_g(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	2016-07-123:12:01

The timestamp field helps to avoid the DoS attacks, avoids redundancy of data from handler trying to attack the network. A new field Timestamp is added to the ARP Request and Reply messages.

This TSCBA is efficiently combatting and preventing MiTM, DoS and host impersonation attack but will incur maintenance cost and construction cost. As per the requirement stated by Al-Hemairy, Amin and Trabelsi [10], ARP storm detection is performed, but ARP scanning is not addressed.

TSCBA's ARP Request Processing

The TSCBA's request processing algorithm generates an ARP request packet with a timestamp whenever the communicating host does not know the destination host's MAC address. This algorithm processes the ARP request by performing the cross-layer consistency check, opcode check, and abnormal packet detection by timestamp validation/expiration, unsolicited target addresses in ARP request

received. When the packet fails to satisfy the above checks, it is appended in the abnormal packet list table to avoid host from being affected by the same attacker. The abnormal packet list table maintains a counter indicating the number of times an entry is made by the same IP-MAC pair.

```
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
            if (IP-MAC Pair in Abnormal_Packet _list_table) then
                count++ /*Increment the count of the entry*/
            else
                update. Abnormal packet list table
            end
            Generate Broadcast Alert Message with TimeStamp
        end
    end
end
```

TSCBA's ARP Reply Processing

The *TSCBA's reply processing algorithm* generates an ARP reply whenever the opcode of the received packet is requested, IP-MAC details of the received packet is valid, cross layer consistency check, IP-MAC entry is found in the ARP cache and Timestamp has not expired. If unsolicited MAC is received in the request or cross

layer consistency is not ensured, then these details are added to abnormal list table and counts are maintained for each entry. If timestamp has expired, a unicast alert message is generated to alert the source host about the expiration.

```
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
            if (IP-MAC Pair in Abnormal_Packet _list_table) then
                count++ /*Increment the count of the entry*/
            else
                update. Abnormal packet list table
            end
            Generate Broadcast Alert Message with TimeStamp
        end
    end
end
```

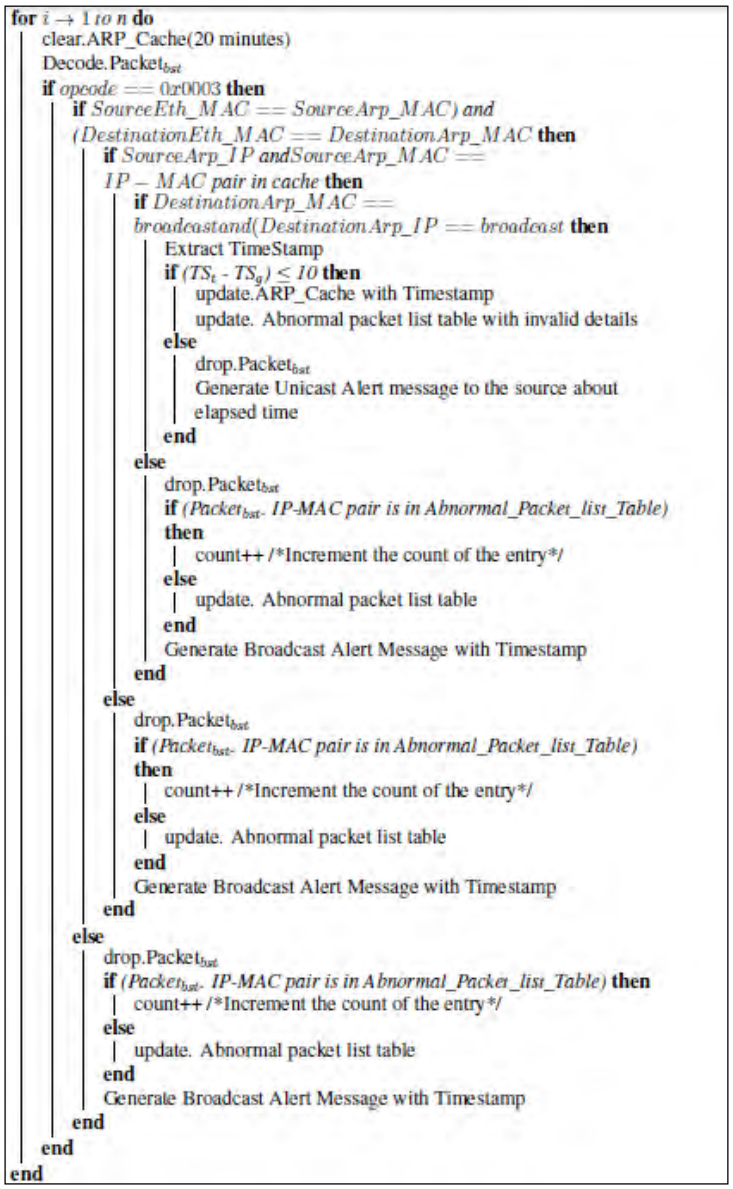
TSCBA's Unicast Alert Message Processing

The *TSCBA's unicast alert message processing* algorithm generates unicast alert message whenever an expired timestamp is received in a packet. It processes the packet with opcode 4, ensures consistency with cross layer check and timestamp checks. If any of these fails it generates unicast and broadcast alert messages accordingly.

```
for i → 1 to n do
    clear.ARP_Cache(20 minutes)
    Decode.Packet_ust 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
                if (Packet_ust. IP-MAC pair is in Abnormal_Packet_list_Table)
                then
                    count++ /*Increment the count of the entry*/
                else
                    update. Abnormal packet list table
                end
                Generate Broadcast Alert Message with Timestamp
            end
        else
            drop.Packet_ust
            if (Packet_ust. IP-MAC pair is in Abnormal_Packet_list_Table) then
                count++ /*Increment the count of the entry*/
            else
                update. Abnormal packet list table
            end
            Generate Broadcast Alert Message with Timestamp
        end
    end
end
```

TSCBA's Broadcast Alert Message Processing

The *TSCBA's broadcast alert message processing algorithm* generates broadcast Alert message with an opcode 3, whenever an abnorml packet is detected at a host. It processes the message by performing opcode test, consistency test and timestamp expiration. If any of these tests outperforms, then generates the respective alert messages



4. Performance of proposed system

The proposed ARP framework is different from the traditional ARP by its features stated in Table 5. The TSCBA achieves the cross-layer checking by comparing the headers viz., ARP and Ethernet headers ensuring the MAC address are the same as intended. This incurs in the sequence of steps say $O(1)$. It also scans the ARP cache to find the existence of the IP-MAC pair which is a sequential search incurring $O(K)$ steps where K is the size of the ARP cache which varies as per the cache clearance.

Table 5. Proposed system features traceability

Features	Description	The component that attains the feature
Authenticated and Stateful	Unlike traditional ARP, this will not process any replies without a request being sent, and it will not update the ARP cache as soon as the ARP reply is received, but there is no mechanism imposed for keeping track of the request-reply pair	It makes use of timestamps and cross-layer checker to make cache updates
Avoids Broadcast storms	Since the alert messages are sent with a timestamp, the floods with the broadcast alert message will never be generated	The ARP request storms are prevented by making use of timestamps and Broadcast Alert messages
Prevents ARP based DoS attacks	The ARP cache is also modified with a new entry of Timestamps. Timed out replies are processed by Unicast Alert messages to the source. Apart from this, it maintains a list of abnormal packets which will help to avoid DoS and DDoS attacks to incur	Abnormal Packet list table, Unicast Alert message and Timestamp feature
Prevents MiTM attacks, DDoS	Depicted in Fig. 14	Cross-Layer inspection and Timestamps

Timestamps are generated by the current date and time of the system clock using the inbuilt function DateTime() which incurs in $O(n)$ where n is the number of parameters required to compute the timestamp. The algorithm updates the malicious packet details in Abnormal Packet list table which incurs $O(1)$. For each packet received the algorithm performs one comparison to perform opcode test which incurs $O(1)$ each. Entire complexity depends on table size K.

The experimental setup of the TSCBA is depicted in Fig. 10. The proposed algorithm has been implemented consisting of the following nodes: A) bearing an IPA-192.169.1.10 MACA-00:5:79:66:68:01, B) bearing IPB-192.169.1.11 MACB-00:5:79:66:68:02, and C) bearing an IPC- 192.169.1.12 MACC-00:05:79:66:68:03.

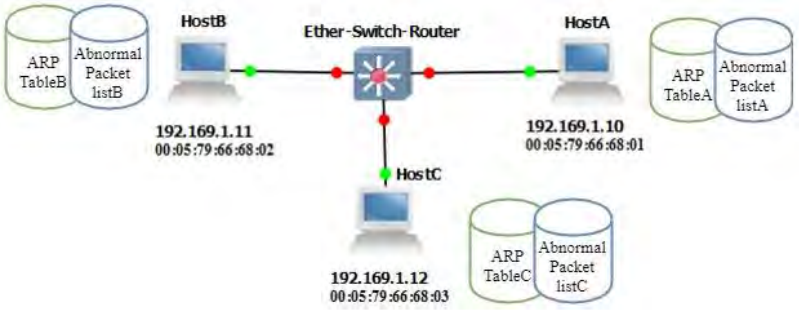


Fig. 10. Experimental setup

Consider a packet with details of A in the source with invalid IP, i.e., 192.168.1.16 and valid MAC is sent to C. At C, it matches the cross-layer check, but IP-MAC pair will not be validated since A's IP is forged. The contents of the Abnormal list table of C and B is shown in Fig. 11.

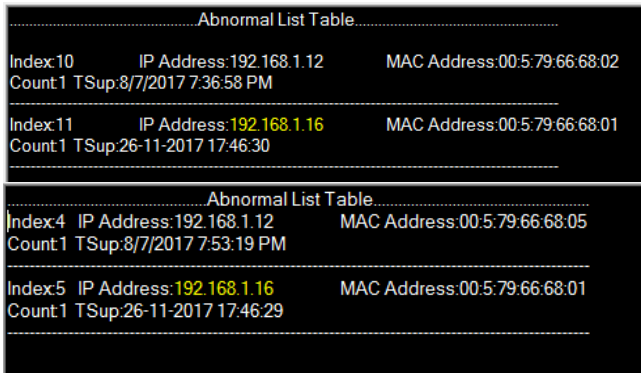


Fig. 11. Abnormal List tables of C and B after receiving a reply with the forged IP address of A

In the above case the packet details are not updated in ARP cache as in traditional ARP, instead added to abnormal packet lists. In turn, C will alert A with a unicast alert message to A, to confirm that it has been forged.

Consider A has received a reply from C with IP:192.168.1.12 and MAC Address: 00:5:79:66:68:34 which is not C's MAC. The ARP Reply is dropped, and Abnormal List table of A is updated with these details, and other hosts were intimated about this. The contents of the table are specified in Fig. 12.

The abnormal list table will be updated as above whenever the following types of packets are being received:

Pkt1. IPF, MACV in source host of ARP request message
Pkt2. IPF, MACV in destination host of ARP request message (broadcast MAC)
Pkt3. IPV, MACF in source host of ARP request message
Pkt4. IPV, MACF in destination host of ARP request message (broadcast MAC)
Pkt5. IPF, MACV in source host of ARP response message
Pkt6. IPF, MACV in destination host of ARP response message

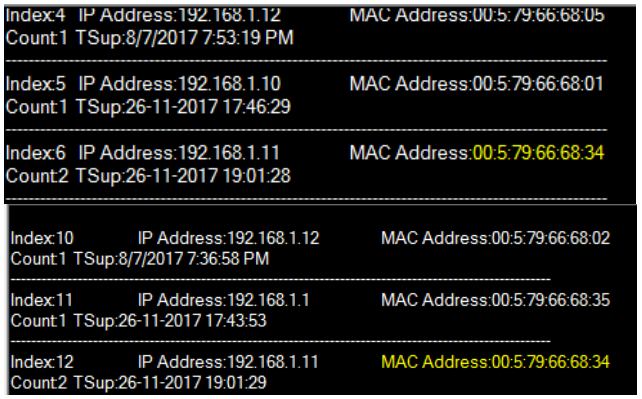


Fig. 12. Abnormal List tables of C and A after receiving a reply with forged MAC address of B

Pkt7. IPV, MACF in source host of ARP response message
Pkt8. IPV, MACF in destination host of ARP response message
Pkt9. IPF, MACV in source/destination host of ARP unicast alert message
Pkt10. IPF, MACV in the source of ARP broadcast alert message

Pkt11. IPV, MACF in source/destination host of ARP unicast alert message

Pkt12. IPV, MACF in destination host of ARP broadcast alert message (here IP and MAC is broadcast)

Pkt13. IPV, MACF in source host of ARP broadcast alert message

Pkt14. IPV, MACF in Source host of gratuitous ARP request message

Pkt15. IPV, MACF in Source host of gratuitous ARP reply message

Pkt16. MAC mismatch between source and destination host of gratuitous ARP reply message

IPF➔Invalid IP address this includes all other types of IP addresses other than the one assigned to the hosts.

MACF➔Invalid MAC addresses, the MACs which are not matched with cross-layer check, and this include NULL MAC, Multicast, and Broadcast MAC addresses.

IPV➔Correct IP address of the sender/receiver is specified in the packet

MACV➔Correct MAC address of the host is specified in the packet

The packets of types specified in Pkt3, Pkt4, Pkt7, Pkt8, Pkt11 and Pk12 will help to combat DDoS, MiTM and host impersonation attacks whereas other packets may also avoid these attacks based explicitly on IP spoofing. Though the algorithmic checks are performed to avoid the attacks based on ARP, these are performed after ARP scanning. Some of the checks like NULL MAC addresses, unused MAC addresses, Multicast addresses can be detected before cross-layer inspection which may reduce the cost. Based on the different types of packets captured by the proposed algorithm, the attack prevention ratio is calculated as

$$\text{Attack\_Prevention\_Rate (\%)} := \frac{\#\text{Attack type packets captured}}{\#\text{Attack type packets injected}}. \quad (1)$$

The attack type packets are computed as follows:

$$\#\text{Attack type packets captured} = \sum_{i=1}^{16} \text{count}(P^i), \quad (2)$$

where P^i is the packet type Pkt_i.

The type of attacks considered for analysis includes MiTM, DDoS, cloning attack, MAC spoofing, IP spoofing and Bombing packet attack (malicious entry in ARP cache). The APR (Attack Prevention Rate) based on the attack type is depicted in Fig. 13 and the attack type packets captured by TSCBA is described in Table 6. The detection of abnormal packet types by various mitigation techniques is described in Table 7. For this scenario, the packets of type Pkt9 to Pkt13 is not considered since it is newly introduced in the proposed system.

4.1. Prevention against MiTM attacks

In Fig. 14, if the host C who knows the IP and MAC address of host A and host B, can be compromised and become malicious when it tries to exploit the communication between A and B with its spoofed MAC as follows:

- Host C captures the request sent by host A to host B and responds host A with ARP reply (IP-B, MAC-C, TSg, TSt). However, host A will not accept this because the host C will be caught in the cross-layer check.
- Host C captures the request sent by host A to host B and modifies it as ARP request (IP-A, MAC-C, TSg, TSt). However, again it will be caught at host B by

cross-layer inspection. In this way, it avoids host impersonation or ARP cloning attack

Table 6. Attack type packets captured by TSCBA

Attack types	% of attack packets detected	Attack packets captured
MiTM	86.67	Pkt3, Pkt4, Pkt5, Pkt7, Pkt8, Pkt14, Pkt15, Pkt16
DDoS	93.33	Pkt1, Pkt2, Pkt3, Pkt4, Pkt5, Pkt6, Pkt7, Pkt8, Pkt9, Pkt10, Pkt11, Pkt13, Pkt14, Pkt15, Pkt16
Cloning attack	93.94	Pkt3, Pkt4, Pkt5, Pkt6, Pkt7, Pkt8, Pkt9, Pkt10, Pkt11, Pkt13, Pkt14, Pkt15
IP spoofing	84.44	Pkt1, Pkt2, Pkt5, Pkt6, Pkt9, Pkt10
MAC spoofing	90.67	Pkt3, Pkt4, Pkt7, Pkt8, Pkt11, Pkt13, Pkt14, Pkt15, Pkt16
Bombing packet Attack	93.89	Pkt1, Pkt3, Pkt4, Pkt5, Pkt6, Pkt7, Pkt8, Pkt9, Pkt10, Pkt11, Pkt13, Pkt14, Pkt15

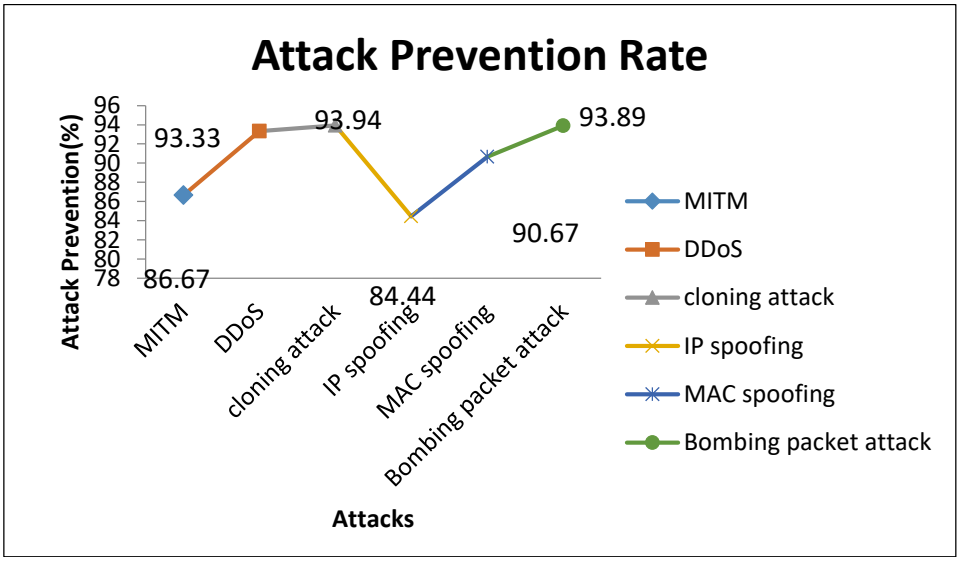


Fig. 13. TSCBA's Attack Prevention Rate

4.2. Prevention against DoS and DDoS attacks

When the malicious host C, with the aim of exploiting the victim B does the following:

- It can send numerous ARP requests between the same time interval TSg and TSt; host B checks the abnormal packet list and if the entry is found, it updates the count and alerts all other hosts else on receiving two or more requests simultaneously, host B will add it to B's abnormal packet list. However, this will be successful only 50% as the host B will not always inspect the ARP requests.
- In case of DDoS attacks, more than one host will send abnormal requests, unlike DoS. If requests are sent to spoof MAC, there is 100% chance for preventing the attack.

Table.7. Detection of malicious packets by various mitigation techniques

Detection techniques	Pkt1	Pkt2	Pkt3	Pkt4	Pkt5	Pkt6	Pkt7	Pkt8	Pkt14	Pkt15	Pkt16
Static ARP	N	N	N	N	N	N	N	N	N	N	N
Centralized Approach	Y	Y	Neutral	Neutral	Y	Y	Neutral	Neutral	N	N	N
SARP	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
GARP	Y	Y	Y	Neutral	N**	N**	N**	N**	N	N**	N
Proposed	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Neutral→partially detects; Y→detect;N→does not detect GARP does not detect reply based malicious packets because the system uses broadcast reply.											

5. Comparison of the proposed system with the existing technique

The proposed system is compared to the existing system based on six security requirements specified in [7]. It is described in Table 8. It is clear that the proposed system does not perform ARP scan. GARP works well, but one disadvantage is that it uses more tables and certifiers which incur additional overhead. The comparison graph is generated by analysing the packet types to ARP, Centralized Approach, GARP, SARP and Proposed Approach. The traditional ARP can detect any packet only if the static entry is maintained, whereas in dynamic nature it can detect only when the entry exists in ARP cache. The packets of the type with NULL MAC and Multicast MAC were also analysed. In such cases, the proposed system can detect by using a cross-layer check, but it can be properly checked before scanning the tables.

(3) $$\text{APPR}\ (\%) := \frac{\#\text{malicious packets captured}}{\#\text{malicious packets sent}}.$$

The Abnormal Packet Prevention Rate (APPR) is calculated as specified in (3). From the graph, it is clear that the traditional ARP is most vulnerable to all these types of malicious packets, whereas the proposed technique shows a better detection rate. Table 9 shows the detection rate of TSCBA against existing techniques.

Table 8. Comparison of the proposed system with existing solutions

Features	Static ARP	Centralized Approach [14]	SARP [17]	GARP [19]	Proposed
Feat1	No	No	Yes	Yes	No
Feat2	No	No	No	No	Yes
Feat3	DD	D	D	DP	DP
Feat4	No	Yes	Yes	Yes	Yes
Feat5	S	SD	D	SD	SD
Feat6	DD	DD	DD	DD	DD
DD→Doesn't Detect;D→Detects;DP→Detects and Prevents Feat1 – Cryptographic based; Feat3 – ARP storms; Feat2 – Cross-Layer inspection; Feat4 – ARP Stateful; Feat5 – Static (S) and Dynamic (D) entries; Feat6 – ARP Scanning;					

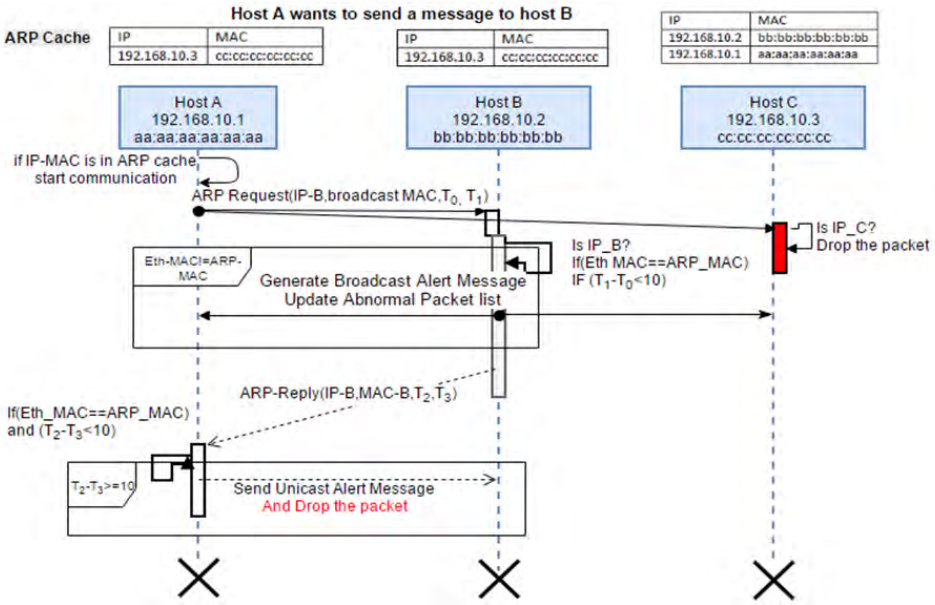


Fig. 14. Timeline chart depicting the proposed ARP request, reply, unicast and broadcast alert messages

Table 9. TSCBA versus existing mitigation techniques

Mitigation techniques	No of malicious Packets injected	No of malicious Packets detected	Detection rate (%)
ARP	1250	150	12
SARP	1250	725	58
GARP	1250	687	55
TARP[18]	1250	625	50
EARP	1250	800	64
TSCBA	1250	1037	83

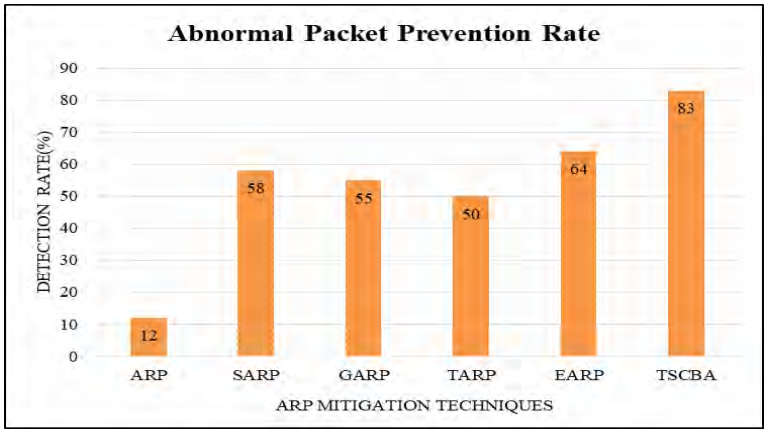


Fig. 15. Comparison of existing techniques

6. Conclusions and future work

The new mitigation techniques for ARP cache poisoning attacks have been analysed, and types of mitigation has been presented. Though these techniques combat the attacks, a novel mitigation system with modifications to traditional ARP messages for the purpose of enhancing their features has been proposed. Also, this new system uses two more messages and data tables to improve its efficiency. A detailed algorithm for the proposed system is developed. Though this technique may be a little costlier than traditional ARP, this system can combat ARP cache poisoning attacks and will prove its cost-effectiveness. The future study is to detect the ARP Scanning by incorporating with any tools available.

References

1. Plummer, D. An Ethernet Address Resolution Protocol (RFC 826). Network Working Group, 1982.
2. Prabadevi, B., N. Jeyanthi. A Framework to Mitigate ARP Sniffing Attacks by Cache Poisoning. – International Journal of Advanced Intelligence Paradigms, Vol. **10**, 2018, No 1-2, pp. 146-159.
3. Abad, C. L., R. I. Bonilla. An Analysis of the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks. – In: 27th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'07), 2007, p. 60.
4. Prabadevi, B., N. Jeyanthi. Distributed Denial of Service Attacks and Its Effects on Cloud Environment – A Survey. – In: IEEE 2014 International Symposium on Networks, Computers and Communications, 2014, pp. 1-5.
5. Jeyanthi, N. N., C. S. N. Iyengar. Escape-on-Sight: An Efficient and Scalable Mechanism for Escaping DDoS Attacks in Cloud Computing Environment. – Cybernetics and Information Technologies, Vol. **13**, 2013, No 1, pp. 46-60.
6. Jeyanthi, N., P. C. Mogan Kumanr. A Virtual Firewall Mechanism Using Army Nodes to Protect Cloud Infrastructure from DDoS Attacks. – Cybernetics and Information Technologies, Vol. **14**, 2014, No 3, pp. 71-85.
7. Jeyanthi, N., R. Thandeeswaran, J. Vinithra. RQA Based Approach to Detect and Prevent DDoS Attacks in VoIP Networks. – Cybernetics and Information Technologies, Vol. **14**, 2014, No 1, pp. 11-24.
8. Shukla, S., I. Yadav. An Innovative Method for Detection and Prevention Against ARP Spoofing in MANET. – International Journal of Computer Science and Information Technology & Security, Vol. **5**, 2015, No 1, pp. 207-214.
9. Jinhua, G., X. Kejian. ARP Spoofing Detection Algorithm Using ICMP Protocol. – In: 2013 International Conference on Computer Communication and Informatics (ICCCI'13), 2013, pp. 1-6.
10. Al-Hemairy, M., S. Amin, Z. Trabelsi. Towards More Sophisticated ARP Spoofing Detection/Prevention Systems in LAN Networks. – In: 2009 International Conference on the Current Trends in Information Technology (CTIT'09), 2009, pp. 1-6.
11. Srinath, D., S. Panimalar, A. J. Simla, J. Deepa. Detection and Prevention of ARP Spoofing Using Centralized Server. – International Journal of Computer Applications, Vol. **113**, 2015, No 19, pp. 26-30.
12. Abdel Salam, A. M., W. S. Elkilani, K. M. Amin. An Automated Approach for Preventing ARP Spoofing Attack Using Static ARP Entries. – International Journal of Advanced Computer Science and Applications, Vol. **5**, 2014, No 1, pp. 96-104.
13. Barbhuiya, F. A., S. Biswas, S. Nandi. An Active Host-Based Intrusion Detection System for ARP-Related Attacks and Its Verification. – In: International Conference on Computer Science and Information Technology, Springer, Berlin, Heidelberg, 2011, pp. 432-443.

14. K a l a j d z i c, K., A. P a t e l. Active Detection and Prevention of Sophisticated ARP-Poisoning Man-in-the-Middle Attacks on Switched Ethernet LANs. – In: Proc. of 6th International Workshop on Digital Forensics & Incident Analysis, 2011, pp. 81-92.
15. T r a b e l s i, Z., W. E l-H a j j. On Investigating ARP Spoofing Security Solutions. – International Journal of Internet Protocol Technology, Vol. **5**, 2010, No 1-2, pp. 92-100.
16. P a n d e y, P. Prevention of ARP Spoofing: A Probe Packet Based Technique. – In: 2013 IEEE 3rd International Conference on Advance Computing Conference (IACC'13), 2013, pp. 147-153.
17. K u m a r, S., S. T a p a s w i. A Centralized Detection and Prevention Technique Against ARP Poisoning. – In: 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012, pp. 259-264.
18. A r o t e, P., K. V. A r y a. Detection and Prevention Against ARP Poisoning Attack Using Modified ICMP and Voting. – In: 2015 International Conference on Computational Intelligence and Networks (CINE'15), 2015, pp. 136-141.
19. T r i p a t h i, N., B. M. M e h t r e. An ICMP Based Secondary Cache Approach for the Detection and Prevention of ARP Poisoning. – In: 2013 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC'13), 2013, pp. 1-6.
20. B r u s c h i, D., A. O r n a g h i, E. R o s t i. S-ARP: A Secure Address Resolution Protocol. – In: Proc. of 19th Annual Computer Security Applications Conference, 2003, pp. 66-74.
21. L o o t a h, W., W. E n c k, P. M c D a n i e l. TARP: Ticket-Based Address Resolution Protocol. – Computer Networks, Vol. **51**, No 15, pp. 4322-4337.
22. D a n g o l, S., S. S e l v a k u m a r, M. B r i n d h a. Genuine ARP (GARP): A Broadcast Based Stateful Authentication Protocol. – ACM SIGSOFT Software Engineering Notes, Vol. **36**, 2011, No 4, pp. 1-10.

Received 25.09.2018; Second Version 22.10.2018; Accepted 06.11.2018