Detection of Stealth Man-In-The-Middle Attack in Wireless LAN

Vikas Kumar, Sandip Chakraborty, Ferdous A Barbhuiya, Sukumar Nandi
Department of Computer Science and Engineering,
Indian Institute of Technology Guwahati, Assam, India
Email: {vikas.k,c.sandip,ferdous,sukumar}@iitg.ernet.in

***Abstract*—Wireless Local Area Networks (WLANs) are acquiring their hold in all the verticals of life. WLANs have gone through rapid changes with respect to their security standards in near time. Man-in-the-Middle (MITM) attack is one of the most catastrophic attacks in WLAN. Stealth MITM (SMITM) attack is a new way of doing MITM based on Address Resolution Protocol (ARP) poisoning. In this attack, ARP poisoning is done directly to the victim by forging the frame ARP response protocol structure and exploiting WPA2 key management. In this paper we propose a Wireless Intrusion Detection System (WIDS) for SMITM attack. The proposed WIDS successfully detects the SMITM attack and other similar attacks like MITM (using ARP poisoning) and IP Spoofing. The proposed WIDS system is simulated in NS-3 network simulator and the scheme is found to work correctly when the attacker is static and is under the coverage of a single sensor during the complete period of attack.**

I. Introduction

Wireless Local Area Network (WLAN) based on IEEE 802.11 [1] is widely used now a days for ease of deployment, maintenance and low end-user cost. IEEE 802.11 can be operated in two modes - infrastructure and ad-hoc. In case of infrastructure IEEE 802.11 WLAN, the access points (APs) works as the centralized coordinator and are connected with the outside Internet, whereas ad-hoc network does not contain any APs and the clients maintain a network connection among themselves. In this paper, IEEE 802.11 WLANs in infrastructure mode is considered with basic service set (BSS) where an AP, connected with a distribution system, provides network access to the end-users or stations.

Security is a major concern for IEEE 802.11 WLANs because of its inherent security vulnerabilities due to broadcast nature in communication. Every data packets sent in a wireless medium is broadcast by nature where any user in the communication range can capture the packet. There are several security standards for IEEE 802.11 WLANs, such as Wired Equivalence Privacy (WEP) [2], Wireless Protected Access version 2 (WPA2) [3] etc. The newest version of security measurement for IEEE 802.11 WLAN defined in IEEE 802.11i [4]–[6] is WPA2 that uses Group Temporal Key (GTK) and Pairwise Transition Key (PTK) for mutual authentication between access points and end-users. GTK is used to encrypt/decrypt broadcast packets whereas PTK is used for unicast traffic.

The second author of this paper is supported by TATA Consultancy Services Research Fellowship, 2011, India

WPA2 is vulnerable to Stealth Man-in-the-Middle (SMITM) attack [7], [8] based on stealth ARP poisoning [8]. In the conventional MITM attack [9], when the AP sends a ARP probe message, the attacker forges an ARP reply to the AP as a false gateway. The conventional MITM attack is shown in Fig. 2. In case of SMITM [8] based on stealth ARP poisoning, the attacker does not reply to the AP directly. In this attack an attacker comes in between two victim clients and forwards all the packets from one victim to another. Here one victim can be an end-user and another victim can be the actual network gateway.

SMITM attack is based on circular shift vulnerability [8] in IEEE 802.11 WLAN frame structure. For the sake of completeness, a brief description of IEEE 802.11 WLAN frame format and circular shift vulnerability in IEEE 802.11 WLAN frame is described here. As mentioned in [1], IEEE 802.11 WLAN frame mainly contains the following basic components,

1) A MAC header, which comprises frame control, duration, address, and sequence control information, and, for QoS data frames, QoS control information;
2) A variable length frame body;
3) A Frame Check sequence (FCS), which contains an IEEE 32-bit Cyclic Redundancy Code (CRC).

A general frame formate of a MAC header in IEEE 802.11 frame is shown in Fig 1. There can be four type of MAC addresses in IEEE 802.11 network - destination MAC address (DA), source MAC address (SA), receiver address (RA) indicating MAC address of the station in BSS that have to receive frame, transmitter address (TA) indicating MAC address of the station which have transmitted frame inside the BSS. Let BSSID demotes BSS identifier - an unique ID to differentiate between BSSs. There are four addresses used in the MAC header depending on the ToDS and FromDS bits defined in the control field, as follows;

- Address 1 is always the recipient address, that is address of the station in the BSS who is the immediate recipient of the packet. If ToDS is set, this is the address of the AP. If ToDS is not set, then this is the address of the end station.

978-1-4673-2925-5/12/$31.00 ©2012 IEEE

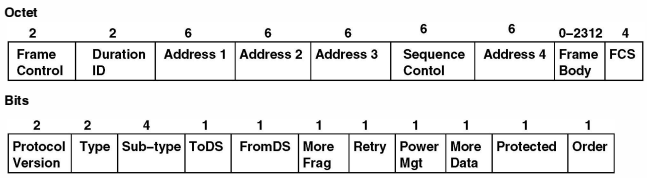

Fig. 1. MAC Header in WLAN 802.11

TABLE I
ADDRESS SETTING IN IEEE 802.11 MAC FRAME

ToDS	FromDS	Address 1	Address 2	Address 3	Address 4
0	0	DA	SA	BSSID	-
0	1	DA	BSSID	SA	-
1	0	BSSID	SA	DA	-
1	1	RA	TA	DA	SA

- Address 2 is always the transmitter address, that is the address of the station in the BSS who is physically transmitting the packet. If FromDS is set, then this is the address of the AP. If FromDS is not set, then this is the address of the end station.
- Address 3 is the original source address if FromDS is set to 1. If ToDS is set, then is is the original destination address.
- Address 4 is used when the frame is transmitted from one AP to another AP in a wireless distribution system, when both ToDS and FromDS bits are set. Then address 4 is the original source address.

The address combination in IEEE 802.11 MAC frame is shown in Table I. There can be two types of traffic in a BSS - uplink traffic (from station to AP) and downlink traffic (from AP to station). The circular shift vulnerability in IEEE 802.11 MAC frame is based on the principle that a right circular shift operation on FromDS, ToDS bits and address 1, address 2, address 3 may change the direction of a frame. That is, if the combination of (ToDS, FromDS) is (0,1) then, a right circular shift makes the recipient address to the transmitter address and the transmitter address to the recipient address. Then another circular shift over address 1, address 2 and address 3 changes the three address and thus the direction of the frame is changed from uplink to downlink. A malicious user may exploit this vulnerability to launch SMITM attack. The steps to be taken by the attacker to launch SMITM attack are as follows,

- Attacker prepares a forged ARP frame.
- Attacker performs the right circular shift operation on the values of FromDS, ToDS and Address-1, Address-2, Address-3 on the IEEE 802.11 frame header, which is added to ARP frame. Thus the direction of frame is changed from uplink to downlink.
- Attacker encrypts the above forged ARP frame with GTK using Hole 196 attack [3].
- Attacker transmits this ARP frame to victim.
- When victim receives this frame, it has no way to know that it is not from AP but from attacker. Victim updates it cache according to the forged frame sent by the attacker and victim's cache get poisoned.

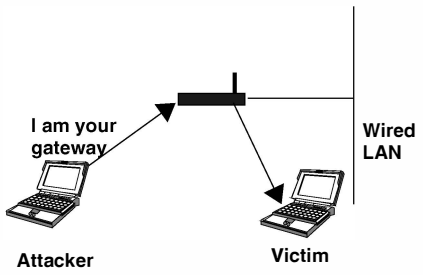

Fig. 2. Convetional Man-In-The-Middle Attack on 802.11

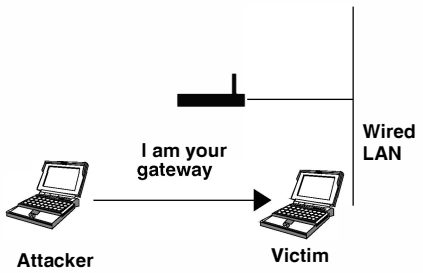

Fig. 3. Stealthy way of Man-In-The-Middle Attack on 802.11

SMITM is different from Conventional MITM attack as shown in Fig 2 and Fig 3. In Conventional Way of MITM as the ARP frames are transmitted via AP, it may be visible to a wired tool and consequently can be detected by a wired tool. However in stealth mode ARP poisoning, transmitted frames are invisible to AP. It can not be detected by network based ARP cache monitoring tool at wireless distribution system.

In this paper, a wireless intrusion detection system (WIDS) is proposed for SMITM attack. The proposed WIDS successfully detects the SMITM attack and other similar attacks like MITM (using ARP poisoning) and IP Spoofing. In the proposed model a set of sensor nodes are placed in the BSS which are capable of sensing the ongoing transmissions between the stations and AP. The sensor nodes are equipped with the proposed WIDS system. The WIDS detects possible attacks based on the analysis of the overhear frames at the sensor nodes. The proposed WIDS system is simulated in NS-3 network simulator and the scheme is found to work correctly when the attacker is static and is under the coverage of a single sensor during the complete period of attack.

II. PROPOSED METHODOLOGY

In the proposed framework, a sensor node is placed inside the BSS that is capable of overhear all the ongoing transmissions inside the BSS. The sensor node is equipped with the WIDS system. The proposed WIDS maintains four tables namely PTable, VTable, OTable and BLTable. AP maintains one table called AP_PTable. The details of these tables are as follows,

- **Verification Table (VTable)** : This table contains verified IP-MAC pairs of all the stations inside the BSS.
- **Probing Table (PTable)** : This table contains three tuples. The first tuple is a nonce which is kept to map the incoming probe response with the probe requests. Other two tuples are the IP address and the MAC address for which probing request has been sent.
- **Other Probing Table (OTable)** : This table keeps the IP-MAC pair against which another IP-MAC pair, having same IP but different MAC address, has been probed earlier by this WIDS node and its response is not received yet.
- **Blocked Table (BLTable)** :This table keeps those IP-MAC pairs which are previously detected as possible attacks or has been negatively responded by the AP.
- **Access Point Probing Table**(AP_PTable) : This table has five tuples. The first tuple contains another table with ¡sensor id, nonce¿ pair for received probe request of the IP-MAC pair. The second and third tuples are the corresponding IP address MAC address which are being verified by the AP. The fourth tuple keeps the time at which this probe request was received, and the fifth tuple keeps a flag whose value is set to 1 when a positive reply for this IP-MAC pair has been received, otherwise the value is set to 0.

A. Working of the WIDS System

The symbols used to describe the working procedure of the WIDS is shown in Table II-A. In the proposed WIDS system, a negative probe reply means IP-MAC is not genuine and a positive reply means that IP-MAC is genuine.

Whenever sensor node capture a ARP request/response, Detector method, as shown in Algorithm 1, checks ARP_SIP-ARP_SMAC existence in PTable to avoid the Re-Probing for the same ARP_SIP-ARP_SMAC at the same time. If both ARP_SIP-ARP_SMAC are in PTable it does nothing but if ARP_SIP is there with different MAC then it keeps the ARP_SIP-ARP_SMAC in OTable. If ARP_SIP-ARP_SMAC is not in PTable, then it checks in BLTable to see whether it has been blocked earlier. If both ARP_SIP-ARP_SMAC are found in BLTable then this indicates that the attacker is retrying the attack on same victim. If only ARP_SMAC is in the BLTable then attacker is attacking to other victims. In this case "Check SMITM", shown in Algorithm 3 is called which checks for the repetition of this MAC in BLTable and VTable and generate an alarm if found.

If ARP_SIP-ARP_SMAC is not in BLTable then it is searched in VTable. If both ARP_SIP-ARP_SMAC are in VTable then "Check SMITM" is called and alarm is raised accordingly. If only ARP_SIP is present in the VTable, then an alarm of IP Spoofing is raised. If none of the ARP_SIP and ARP_SMAC is found then it calls Verify function for ARP_SIP-ARP_SMAC. The "Verifier" is described in Algorithm 2. The "Verifier" generates a nonce, and for ARP_SIP-ARP_SMAC it sends a Probe Request to AP with the nonce and make an entry in PTable for ARP_SIP-ARP_SMAC with the nonce $n1$.

AP runs another method called "AP Probe Handler", shown in Algorithm 5, which is executed on receiving a probe request from the sensor node. The "Probe Response Handler" method, shown in Algorithm 4 is executed when a probe response from AP is overheard at the sensor node. On receiving the probe request from the sensor node, if $PRqIP$ and $PRqMAC$ belongs to the AP then it immediately sends positive probe reply. If the $PRqIP$ belongs to AP but $PRqMAC$ is different from MAC_AP, then the AP sends a Negative reply. If none of the $PRqIP$ and $PRqMAC$ pair belongs to AP, then it sends an ARP request for $PRqIP$ and waits for reply. The reply is handled in Algorithm 6. There may be five cases as follows,

Algorithm 1 Attack Detector

Input : Captured ARPs.
Output: Raise Alarm in case of SMITM or IP-Spoofing.

1: Capture the ARPs which are not induced by the verification being run by AP.
2: **if** $ARP\_SIP = PIP_i \wedge ARP\_SMAC \neq PMAC_i$ **then**
3: ADD $\{ARP\_SIP, ARP\_SMAC\}$ to OTable.
4: **end if**
5: **if** $ARP\_SIP \neq PIP_i \wedge ARP\_SMAC \neq PMAC_i$ **then**
6: **if** $ARP\_SIP = BLIP_i \wedge ARP\_SMAC = BLMAC_i$ **then**
7: **if** $CheckSMITM(ARP\_SIP, ARP\_SMAC) = True$ **then**
8: ALARM SMITM.
9: **else**
10: ALARM Already Black Listed.
11: **end if**
12: **else if** $ARP\_SIP = BLIP_i \wedge ARP\_SMAC \neq BLMAC_i$ **then**
13: **if** $CheckSMITM(ARP\_SIP, ARP\_SMAC) = True$ **then**
14: ALARM SMITM.
15: **end if**
16: **end if**
17: **end if**
18: **if** $ARP\_SIP = VIP_i \wedge ARP\_SMAC = VMAC_i$ **then**
19: **if** $CheckSMITM(ARP\_SIP, ARP\_SMAC) = True$ **then**
20: ALARM SMITM.
21: **else if** $ARP\_SIP = VIP_i \wedge ARP\_SMAC \neq VMAC_i$ **then**
22: ALARM IP-Spoofing.
23: **else if** $ARP\_SIP \neq VIP_i \wedge ARP\_SMAC \neq VMAC_i$ **then**
24: Verifier(ARP_SIPs, ARP_SMACs)
25: **end if**
26: **end if**

Algorithm 2 Verifier

Input : ARP_SIP, ARP_SMAC.
Output: Sends the Verification Probes Request.

1: n1←Nonce().
2: $Send_{prob\_request}$ $\{ARP\_SIP, ARP\_SMAC, n1\}$.
3: ADD $\{ARP\_SIP, ARP\_SMAC\}$ to PTable.

TABLE II
SYMBOLS USED IN THE ALGORITHMS

Symbol	Meaning
ARP_SIP	Source IP of ARP Request / Response captured at sensor node
ARP_SMAC	Source MAC of ARP Request / Response captured at sensor node
$PRsn1$	A random nonce received with Probe Response
$PRsIP$	IP address received with Proble Response
$PRsMAC$	MAC address received with Proble Response
$PRqn1$	A random nonce received with Probe Request
$PRqIP$	IP address received with Proble Request
$PRqMAC$	MAC address received with Proble Request
PIP_i	IP at the i^{th} level of PTable
$PMAC_i$	MAC at the i^{th} level of PTable
VIP_i	IP at the i^{th} level of VTable
$VMAC_i$	MAC at the i^{th} level of VTable
$BLIP_i$	IP at the i^{th} level of BLTable
$BLMAC_i$	MAC at the i^{th} level of BLTable
OIP_i	IP at the i^{th} level of OTable
$OMAC_i$	MAC at the i^{th} level of OTable
n	total number of entries
$vcount$	Counter value for Vtable
$blcount$	Counter value for BLTable
ARP_RIP	IP received from ARP Reply to AP
ARP_RMAC	MAC received from ARP Reply to AP
$Time\_Difference_i$	Current Time - Fourth tuple in AP_PTable
$Waiting\_Threshold$	Time duration for which AP have to wait for ARP reply after forwarding ARP Request
IP_AP	IP address of AP
MAC_AP	MAC address of AP

Algorithm 3 Check SMITM

Input : ARP_SIP, ARP_SMAC.
Output: True if SMITM is there or False.

1: $vcount \leftarrow 0$
2: $blcount \leftarrow 0$
3: **for** $i = 0 \rightarrow n$ **do**
4: **if** $ARP\_SMACs == VMAC_i$ **then**
5: $vcount \leftarrow vcount + 1$
6: **end if**
7: **if** $ARP\_SMACs == BLMAC_i$ **then**
8: $blcount \leftarrow blcount + 1$
9: **end if**
10: **end for**
11: **if** $vcount = 1 \wedge blcount = 1$ **then**
12: Return False.
13: **else if** $vcount \geqslant 2 \wedge blcount \geqslant 2$ **then**
14: Return True.
15: **end if**
16: Return False.

Algorithm 4 Probe Response Handler

Input : Probe Response.
Output: Analyzes the Probe Response from Access Point.

1: Recieve a probe response with nonce $PRsn1$.
2: **for** $i = 0 \rightarrow n$ **do**
3: **if** $PRsn1 == nounce_i$ **then**
4: $place \leftarrow i$
5: **end if**
6: **end for**
7: **if** $PRsIP = PIP_{place} \wedge PRsMAC = PMAC_{place}$ **then**
8: ADD $PRsIP$, $PRsMAC$ to VTable.
9: **else**
10: **if** CheckSMITM($PRPsIP, PRsMAC$) == $True$ **then**
11: ALARM SMITM.
12: **else**
13: ALARM IP-Spoofing.
14: **end if**
15: ADD $PRsIP$, $PRsMAC$ to BLTable.
16: **end if**
17: Delete $PRsIP, PRsMAC$ from PTable
18: **if** $PRsIP = OIP_i$ **then**
19: Verifeir($PRsIP, OMAC_i$).
20: Delete $PRsIP, OMAC_i$ from OTable.
21: **end if**

1) Attacker and victim both reply for the ARP request.
2) Only victim node reply and attacker does not.
3) Neither attacker reply nor victim reply for the ARP request.
4) Attacker does not let victim reply for ARP request and the attacker itself sends the reply.
5) Only victim node reply because there is no attacker.

In the first three cases AP understands that the $PRqIP$,$PRqMAC$ pair is not genuine and reply negative. In cases 4 and 5, AP replies positive (Case 4 is taken care by the Detector method as shown in Algorithm 1). In case 1 and 2, AP immediately replies for probe as it is clearly an attack. However in case 3, $Waiting\_Threshold$ is doubled and ARP request is sent again because even a genuine station node may not receive ARP request due to heavy traffic. In case 4 and case 5, when an ARP reply is received, the AP checks whether the row in AP_PTable corresponding to ($PRqn1$, $PRqIP$, $PRqMAC$) is older than the waiting threshold. If not then AP wait for more reply as there might be the case when attacker node has replied and a genuine station node may reply after that. This is handled in a way similar to the first case.

Algorithm 5 AP Probe Handler

Input : A Probe Request with PRqIP-PRqMAC.
Output: Send ARP Request/Probe Response.

1: **if** $PRqIP = IP_{AP} \wedge PRqMAC = MAC_{AP}$ **then**
2: $Send_{prob\_Response}\ \{SID, n1, PRqIP, PRqMAC\}$.
3: **else if** $PIP = IP_{AP} \wedge PMAC \neq MAC_{AP}$ **then**
4: $Send_{prob\_Response}\ \{SID, n1, PRqIP, NULL\}$.
5: **end if**
6: $Send_{arp\_request}\ \{PRqIP\}$.
7: ADD $\{PRqSID, PRqn1, PRqIP, PRqMAC, Current\_Time, False\}$.

Algorithm 6 AP Probe Responder

Input : ARP Reply for $PRqIP_i$ in AP_PTable.
Output: Send "Probe Response".

1: **if** $PRqIP_i = ARP_RIP \wedge PRqMAC_i = ARP_RMAC$ **then**
2: $PRq - Flag_i \leftarrow True$.
3: **if** $Time\_Difference_i < Waiting\_threshold$ **then**
4: Wait.
5: **else**
6: $Send_{prob\_Response}\ \{SID_i, n1_i, PRqIP_i, PRqMAC_i\}$.
7: Delete $\{PRqIP_i, PRqMAC_i\}$ from AP_PTable.
8: Return.
9: **end if**
10: **else if** $PRqIP_i = ARP_RIP \wedge PRqMAC_i \neq RP_RMAC$ **then**
11: $Send_{prob\_Response}\ \{SID_i, n1_i, PRqIP_i, NULL\}$.
12: Delete $\{PRqIP_i, PRqMAC_i\}$ from AP_PTable.
13: Return.
14: **else if** $Time\_Difference_i \geqslant Waiting\_threshold \wedge PRq - Flag_i = False$ **then**
15: Waiting Threshold is over.
16: **if** $Time\_Difference_i \geqslant 2 \times Waiting\_threshold$ **then**
17: $Send_{arp\_request}\ \{PRqIP_i\}$.
18: **else**
19: $Send_{prob\_Response}\ \{SID_i, n1_i, PRqIP_i, NULL\}$.
20: Delete $\{PRqIP_i, PRqMAC_i\}$ from AP_PTable.
21: Return.
22: **end if**
23: **else if** $Time\_Difference_i \geqslant Waiting\_threshold \wedge PRq - Flag_i = True$ **then**
24: $Send_{prob\_Response}\ \{SID_i, n1_i, PRqIP_i, PRqMAC_i\}$.
25: Delete $\{PRqIP_i, PRqMAC_i\}$ from AP_PTable.
26: Return.
27: **else if** $Time\_Difference_i \leqslant Waiting_threshold \wedge PRq - Flag_i == False$ **then**
28: Wait.
29: **end if**

At sensor node "Probe Response Handler" method receives a probe response from AP and checks whether reply is positive or negative, as shown in Algorithm 4. An entry is made in VTable for $PRsIP$, $PRsMAC$ pair in case of positive reply. Otherwise the sensor node checks for SMITM using Algorithm 3. An alarm for SMITM is raised if it is a SMITM attack, otherwise an IP Spoofing attack alarm is raised. An entry is made in BLTable for $PRsIP$, $PRsMAC$ pair. After this it checks the existence of $PRsIP$ in OTable. If it is there then it calls the "Verifier" for this IP-MAC pair of OTable and deletes this row from the OTable. Thus this cycle is repeated for each captured ARP packets at the sensor node.

Fig. 4. Verified IP-MAC pairs at Sensor node 2

B. Network Load

The proposed detection Methodology incurs some load on the network as discussed follows.

1) Normal Scenario: Let the number of station nodes in the BSS is δ. Number of extra packets transmitted in the network by the WIDS system is - the Probe Request and Probe Response, one ARP broadcast by the AP to verify IP-MAC pair and one ARP reply. The extra number of packets transmitted inside the BSS is:

$$\text{Load} = (2 + 1 + 1)\delta = 4\delta$$

2) Attack Scenario: In this case, there are two packets for Probe Request and Probe Response, one ARP broadcast by AP to verify IP-MAC and there may be either one or two or no ARP reply. This much traffic is produced for each station node in the BSS. So the the total number of extra packets inside the BSS is,

$$(2 + 1 + 2)\delta \geqslant \text{Load} \geqslant (2 + 1)\delta$$
$$5\delta \geqslant \text{Load} \geqslant 3\delta$$

III. Simulation Results

The proposed WIDS scheme is simulated using NS-3.13 Network simulator. The network consists a single BSS with a single AP, where 7 sensor nodes equipped with the proposed WIDS system is placed so that all the 7 sensors can collectively cover the complete BSS. 40 stations are placed inside the BSS with random mobility. Out of the 40 stations, a station behaves like the attacker. The attacker does steal ARP poisoning over two different victims. The network setup for the attacker and the victims are as follows;
Attacker′s IP = 10.1.3.1 and MAC = 00:00:00:00:00:01
Victim-1′s IP = 10.1.3.3 and MAC = 00:00:00:00:00:03
Victim-2′s IP = 10.1.3.4 and MAC = 00:00:00:00:00:04

The system is simulated for two different scenarios. One is the normal scenario and the second is the attack scenario. Figure 4 show the VTable of a sensor node containing verified IPs and MACs of the station node in the range of sensor node. In normal scenario BLTable remains empty as there is no attacker.

Fig. 5. ARP Cache of Node 3, Node 4 and AP During Attack

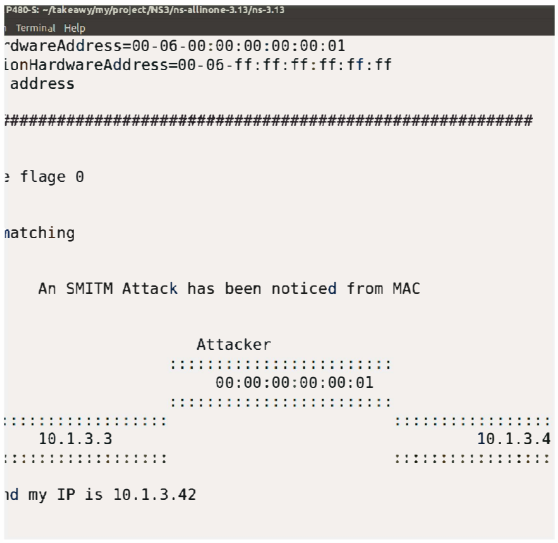

Fig. 6. SMITM Attacke is detected

The second scenario is the attack scenario. The attacker keep on listening on its interface in promiscuous mode for a ARP request from victim-1 to victim-2. As soon as it get that it sends a Fake ARP response. The sensor node in the range of the attacker node captures this ARP Reply from attacker and sends a probe request to the AP with the source IP-MAC of ARP reply. On receiving the probe request AP broadcasts an ARP request for the IP-MAC pair. Based on the ARP reply, the AP sends either a positive or a negative reply to sensor node. Sensor node then finds out whether it is a IP-Spoofing or SMITM attack or it is a normal scenario. Figure 5 shows the ARP Cache of node 3, node 4 and the AP. Fig 6 shows when SMITM is detected.

Based on the positions of attacker, victims and the sensor nodes there can be three cases possible,

1) Attacker and both the victims are static and are in the coverage of same sensor node.
2) Attacker is static and in the coverage of a sensor nodes, however both the victims are in the coverage of a different sensor nodes.
3) Attacker is mobile. Attacker conducts ARP poisoning over victim-1 under the coverage of a sensor, and then moves to the coverage of another sensor where it conducts ARP poisoning over victim-2.

It has been observed from simulation traces, that the proposed WIDS system can detect the attacks when the attacker is static. However, the system fails when the attacker is mobile. At that time when the attacker is mobile, there is a requirement for cooperation between the sensors.

IV. Conclusion

In this paper a WIDS is proposed to detect SMITM and IP-spoofing attack in a BSS. The proposed scheme can also detect conventional ARP poisoning using MITM attack. A set of sensor nodes are placed inside the BSS that overhear ongoing communications between the AP and the stations. Based on the overhear ARP request and response messages, the sensor nodes send a probe to the AP to gather additional information. The attacks are detected based on the information obtained by the sensor nodes and the AP. The simulation results show that the proposed scheme works correctly when the attacker is static. However, for mobile attacker, there is a requirement for cooperation between the sensor nodes to detect ARP poisoning correctly.

References

[1] "IEEE standard for information technology telecommunications and information exchange between systems local and metropolitan area networks specific requirements," *IEEE Standard*, 2007.
[2] A. Lashkari, M. Mansoor, and A. Danesh, "Wired equivalent privacy (WEP) versus wi-fi protected access (WPA)," in *Proceedings of the 2009 International Conference on Signal Processing Systems*, 2009, pp. 445–449.
[3] M. Mathews and R. Hunt, "Evolution of wireless LAN security architecture to IEEE 802.11i (WPA2)," in *Proceedings of the Fourth IASTED Asian Conference on Communication Systems and Networks*, 2007, pp. 292–297.
[4] D. Bae, J. Kim, S. Park, and O. Song, "Design and implementation of IEEE 802.11i architecture for next generation WLAN," in *Proceedings of the First SKLOIS conference on Information Security and Cryptology*, 2005, pp. 346–357.
[5] F. De Rango, D. C. Lentini, and S. Marano, "Static and dynamic 4-way handshake solutions to avoid denial of service attack in Wi-Fi protected access and IEEE 802.11i," *EURASIP J. Wirel. Commun. Netw.*, vol. 2006, no. 2, Apr. 2006.
[6] S. E. Frankel, B. Eydt, L. Owens, and K. A. Scarfone, "SP 800-97. establishing wireless robust security networks: A guide to IEEE 802.11i," Tech. Rep., 2007.
[7] A. Herzberg and H. Shulman, "Stealth-MITM DoS attacks on secure channels," *CoRR*, vol. abs/0910.3511, 2009.
[8] M. S. Ahmad, "WPA TOO!" in *Proceedings of the DEFCON 18*, 2010.
[9] H. Hwang, G. Jung, K. Sohn, and S. Park, "A study on MITM (man in the middle) vulnerability in wireless network using 802.1X and EAP," in *Proceedings of the 2008 International Conference on Information Science and Security*, 2008, pp. 164–170.