Detection of MITM Attack in LAN Environment using Payload Matching

Dawood Al Abri
Department of Electrical and Computer Engineering
Sultan Qaboos University
Muscat, Sultanate of Oman
Email: alabrid@squ.edu.om

***Abstract*— Man-in-the-Middle (MITM) attack enables an attacker to monitor the communication exchange between two parties by directing the traffic between them to pass through the attacker's machine. Most existing schemes for discovering MITM attack focus on detecting the mechanism used to direct the traffic through the attacker machine. This paper presents a new detection scheme that is based on matching the payload of frames exchanged in the network. The proposed scheme is independent of the mechanism used to launch the MITM attack. Experimental result shows that the proposed scheme can achieve excellent detection performance with proper choice of the scheme's tuning parameters.**

Keywords—MITM; detection; attack, security, traffic analysis, ARP poisoning

I. Introduction

The widespread use of networks in everyday life transactions lures more attackers to exploit the inherited weakness in most networking protocols to accomplish malicious goals such as obtaining sensitive information. Man-in-the-Middle (MITM) is an attack where the traffic between the two communicating parties is directed in such a way that it passes through the attacker's machine. This enables the attacker to accomplish many malicious goals such as denial of service by not forwarding the traffic between the two parties and passive monitoring of traffic. In the last case, it is very unlikely that the communicating parties can detect the passive monitoring since information exchange between them goes as expected and without modification from their point of view.

Most documented detection schemes rely on monitoring traffic for the presence of malicious packets that can be used to divert the traffic in a way that suggests an MITM attack. Although they are effective if the attacker uses that particular way of diverting the traffic, they will fail to detect MITM if the attacker utilize a different approach to accomplish the attack. In this paper, we present a detection scheme that focuses on detecting MITM independent of the scheme used to divert the traffic. The new scheme relies on looking for a match between the payloads of different frames transmitted in the network.

The remainder of this paper is organized as follows. Section II reviews the MITM attack. In section III, related work is discussed. The proposed scheme is presented in section IV. Section V presents the experimental results. Finally, section VI concludes the paper.

II. Man in the Middle Attack

The section explains briefly the Man in the Middle (MITM) attack. In MITM, the traffic is directed in such way that it passes through the attacker's machine, which enables him to view the data being exchanged among other things. One popular way to accomplish this attack is to exploit the Address Resolution Protocol (ARP) to propagate wrong mapping between the Internet Protocol (IP) address and Media Access Control (MAC) address of the target machine. This attack is known as the ARP poisoning. In typical setting, nodes in Local Area Network (LAN) use ARP to create a mapping between an IP address and the corresponding MAC address of other nodes they need to communicate with. For example, in attack-free setting as in Fig. 1 (a), each of the nodes A and B, using ARP, obtains the correct MAC address that corresponds to the other node's IP address. This information is stored in an ARP table and utilized whenever a node wants to send something to the other one.

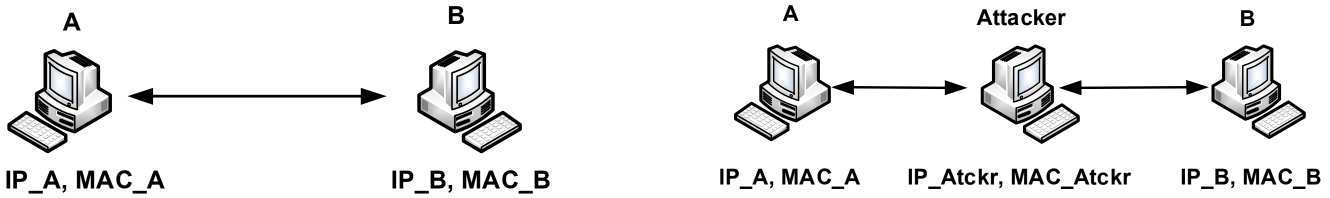
In the case of ARP poisoning, the attacker sends bogus ARP packets to fill the ARP tables of targets machines (A and B) with wrong mapping information. As shown in Fig. 1(b), the ARP entry of both nodes has the IP address of the other node pointing to the attacker MAC address. So, whenever a new frame is constructed for the other node, the frame will have the MAC address of the attacker as the destination address, and hence the traffic will be directed by the LAN switch to the attacker machine. The attacker will simply relay the traffic received from one node to the other node.

Besides ARP poisoning, attacker may utilize other mechanisms to accomplish MITM attack such as:

- Placing itself as a gateway machine. For example, attacker could setup a bogus DHCP server and then send IP configuration information that includes itself as the gateway.
- Poisoning DNS entry for certain DNS records (e.g. proxy) to point to the attacker. This way traffic will be sent directly to the attacker which then forwards it to the correct destination.
- Using *ICMP redirect* to instruct the victim to direct the traffic to the attacker machine as a better path to the destination.

Although the above list is by no mean comprehensive, it shows that there are several ways to accomplish the MITM.

978-1-4799-7800-7/15/$31.00 ©2015 IEEE



ARP entry for B in A's table:	***ARP entry for A in B's table:***
IP_B, MAC_B	***IP_A, MAC_A***

(a) Normal

ARP entry for B in A's table:	***ARP entry for A in B's table:***
IP_B, MAC_Atcker	***IP_A, MAC_Atcker***

(b) ARP poisoning

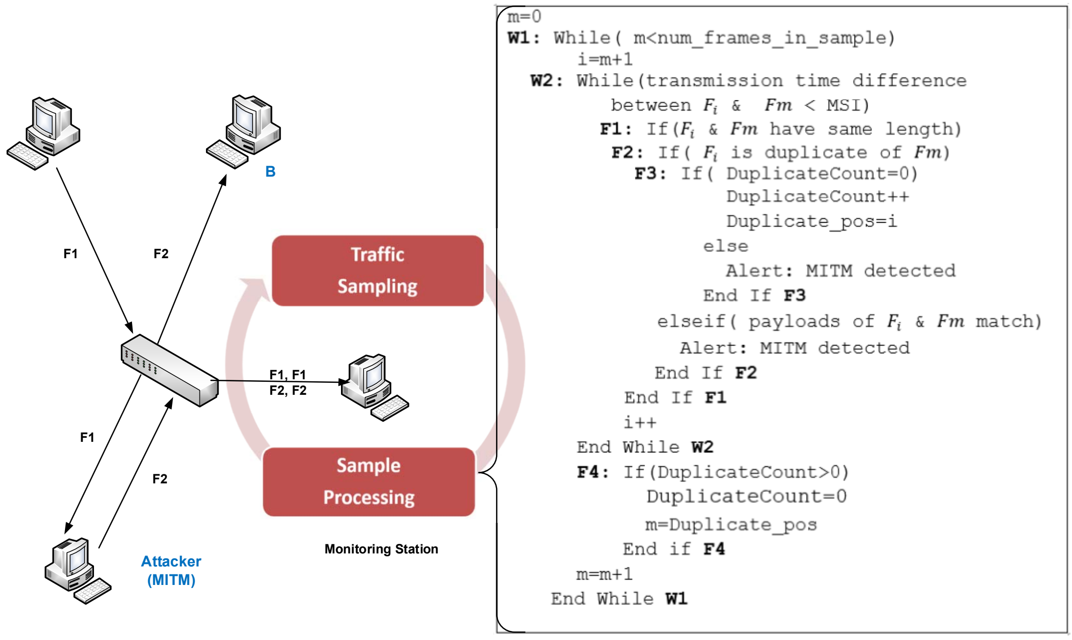
Fig. 1 Illustration of the use ARP poisoning to accomplish MITM. (a) Normal operation of ARP. (b) ARP poisoning case

III. Related Work

One approach used to deduce the presence of MITM attack is to detect the mechanism used to launch the attack, particularly the popular ARP poisoning. Two methods of detecting ARP poisoning is reported in[1]: the first relies on reverse poisoning of ARP table of any node that sends an ARP reply and then performing a probing step to see if that node will rely back a test packet sent to it or not. If it relays back the test packet, then it presumed to be an attacking machine. This method fails to detect ARP poisoning attack by tools that utilize their own ARP table rather than the host local ARP table. The second method described in [1] alters the CAM table of switch to monitor if a test packet sent by a monitoring host is sent back to it. The drawback of this method is that it may disturb normal traffic if a large number of test packets are sent to increase the effectiveness of detection. Moreover, under heavy traffic condition, many switches effectively broadcast the traffic rather than forward it, which renders this detection method as ineffective. Several ARP poisoning detection algorithms for hubbed and switched environments are described in [2]. Of particular interest here is the duplicate packet detection algorithm described in [2] which uses a similar idea to the work presented in this paper. The goal of the scheme in [2] is to detect ARP spoofing in hubbed environment where the entire content of IP packet modulo the Ethernet addresses, the IP TTL field and the IP header checksum is hashed and used as index in state table. If new packet produces a hash that is already present in the table, then an alert about relaying of a packet is generated. The hash is removed from the table after two seconds. The scheme presented in [2] focuses on ARP poisoning (spoofing) detection whereas the scheme presented in this paper focuses on detection of MITM whether it accomplished via ARP poisoning or other means. Moreover, the scheme proposed here has three configurable parameters to tune its performance which is not the case with scheme in [2]. A good analysis and comparison of schemes to detect and prevent ARP poisoning can be found in [3].

Another approach employed by researchers to detect attacks is to observe anomalies in the network. Typically, in this type of detection, a reference for the normal network behavior is constructed and then any divergence from normal operation (e.g. spike in traffic level) is signaled as an attack. In general, most intrusion detection systems (IDSs) follow this approach. One example of such IDS approach is [4] where the author uses discrete event system (DES) to build a model for the network under normal condition and under attack condition in order detect ARP related attacks. The proposed approach to detect MITM in [4] is to monitor if two responses have source and destination IP addresses flipped but with the same source MAC. It is essentially detecting the scheme used to launch the MITM (ARP poisoning) which will fail if the attacker uses a different approach to initiate the attack (e.g. DNS poisoning). In [5], authors proposed a scheme to detect MITM based on the delay measurement. In the proposed method, if the mean delay of connection is abnormally longer than suggested by previous data, MITM is assumed present. This method may suffer whenever the delay measurements exhibit large variation due to unstable network condition or if the attacker can forward the packets very fast so that the delay measurements with or without the attacker are undistinguishable. The authors in [6] present a low cost embedded IDS that has reactive and proactive modes. In the reactive mode, it monitors the ARP message to detect suspicious messages or unsolicited ARP replies. In the proactive mode, it periodically refreshes the ARP information of all active hosts to ensure they have correct binding between IP and MAC. The use of low cost hardware limits the capability of the proposed system to low input load for the reactive mode. In [7], authors present a scheme, called DTRAB, to detect the presence of various attacks on encrypted protocols. DTRAB relies on extracting traffic features, building normal traffic profile, and then generating an alert whenever there is a deviation from the normal case. As acknowledged in [7], DTRAB may have difficulty with IPSec-based end-to-end secure tunnel. The authors in [8] describes a scheme to detect and protect against MITM by allowing web application to vouch for the authenticity of their certificate based on previously shared secrets.

To the best of the author's knowledge, there is no other work that attempts to detect the MITM attack in switched LAN environment by matching the payload content of the frames transmitted. Notice that the proposed approach is different from traditional IDS where the payload may be examined to detect abnormality. Here, selective portion of payload from different transmitted frames are compared to detect a match which will be used as a sign for the presence of an MITM



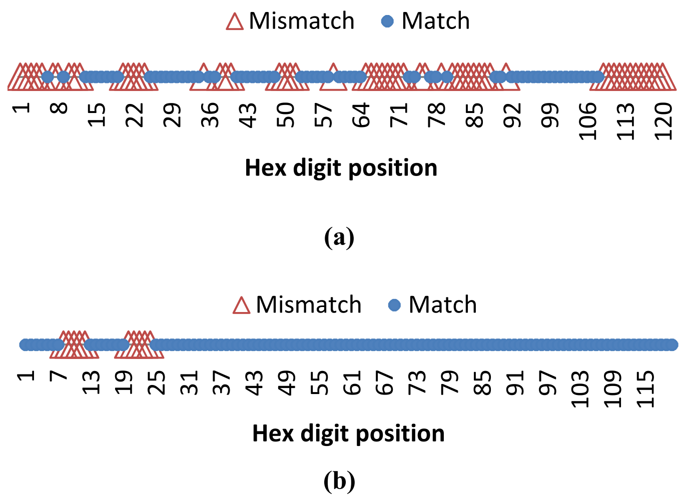
```
m=0
W1: While( m<num_frames_in_sample)
      i=m+1
  W2: While(transmission time difference
         between $F_i$ & $Fm$ < MSI)
        F1: If($F_i$ & $Fm$ have same length)
         F2: If( $F_i$ is duplicate of $Fm$)
           F3: If( DuplicateCount=0)
                   DuplicateCount++
                   Duplicate_pos=i
                 else
                   Alert: MITM detected
                 End If F3
             elseif( payloads of $F_i$ & $Fm$ match)
               Alert: MITM detected
             End If F2
          End If F1
          i++
      End While W2
      F4: If(DuplicateCount>0)
            DuplicateCount=0
            m=Duplicate_pos
          End if F4
      m=m+1
   End While W1
```

Fig. 2 The proposed detection scheme setup.

attack as we shall explain in more detail in Section IV. There is no need in the proposed scheme to have a reference model for the network under normal or abnormal operation. Moreover, the proposed scheme is not tied to a particular mechanism of initiating the MITM but it attempts to detect the presence of MITM regardless of the scheme used to launch the attack.

IV. The Proposed Detection Scheme

As mentioned previously, there are many ways to launch the MITM attack. Rather than focusing on detecting the specific mechanism used to launch the MITM, the scheme proposed here exploits the expected similarity of traffic generated by the victim node and the traffic relayed again by the attacker to detect the presence of MITM regardless of the mechanism used to launch it. The key idea of the proposed scheme can be illustrated using the setup shown in Fig. 2. Here, we assume that an attacker manages, via some mean, to launch a MITM attack against A and B. Under such scenario, a frame **F1** sent from A to B will reach first the attacker which then forwards it as new **F2**, which maybe a slightly modified version of **F1** (e.g. change destination MAC address from the attacker MAC address to that of B in case of ARP poisoning). We refer to these two frames (i.e. the frame received from one victim and the corresponding frame relayed by the attacker) as an *MITM frame pair* (or simply *MITM pair*). Therefore, if both frames are compared carefully, we would expect to see high similarity between them especially in passive attack. Even if certain fields are modified as mentioned previously, the similarity would persist over a good portion of the frames' lengths. This point is illustrated in Fig. 3 where pairs of frames are taken from a captured traffic stream (in hexadecimal format) and the first 120 hex digits (60bytes) are compared. In Fig. 3(a), the pair consist of two unrelated frames and it is clear that there is a lot of difference between the digits throughout the comparison length. On the other hand, when MITM frame pair is used for comparison (Fig. 3(b)), there is a perfect match starting from the 25th hex digits till the end of the frame.



(a)

(b)

Fig. 3 Similarities between the first 120 hex digits in captured hex traffic stream of: (a) two unrelated frames (b) MITM pair.

The proposed scheme is based on detecting the similarity in traffic to discover the presence of an MITM attack. A station is used to monitor the traffic that goes through the LAN (see Fig. 2). The monitoring station performs the following two steps:

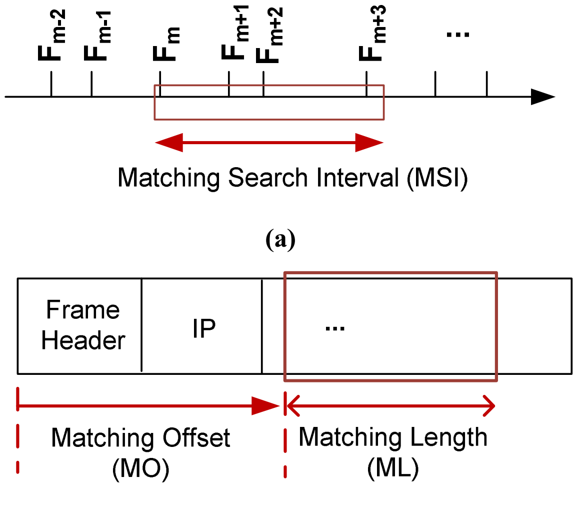
- **Traffic Sampling**: here a sample of all traffic that passes through the network is gathered by the monitoring station, for example, by utilizing switch port mirroring capability that is available in wide range of switches. The sample duration should be chosen to be long enough to enable reliable detection without requiring excessive processing power. In the limiting case, continuous sampling can used to ensure maximum detection reliability at the expense of high computational power.

- **Sample Processing**: here frames within the traffic sample are processed to detect the presence of MITM pair which will serve as an indicator for the presence of MIM attack. The processing involve the following steps:
    - **Pairing the frames:** each frame will be paired with all the possible candidate frames that lie within time interval that we call *matching search interval* (MSI) as illustrated in Fig. 4(a). The reason for introducing MSI is to reduce the overall computational power required as compared to pairing each frame with all other frames. The MSI should be chosen to be large enough so that a frame and its MITM pair (if it exists) lie within MSI. In typical MITM attack, the header information of the two frames of MITM pair will be different as some fields of the header (typically layer 2 header) need to be modified in order to direct the traffic to desired destination (attacker or the other victim). The remaining of the frame remains usually unmodified unless the attacker is carrying an active attack. Based on these observations, some initial screening is done to eliminate frames that are very unlikely to constitute MITM pairs. First, exact match frames are eliminated because, as seen from Fig. 2, any frame will appear twice in the captured traffic: one from input port and other copy from output port. One special case that needs to be considered here is when the attacker is performing the MITM attack by manipulating the forwarding mechanism of the switch (e.g. port stealing). Here, the switch is tricked to forward the packets for victim machine to the attacker which then relays them without actually modifying their content. In this case, the same frame is expected to appear four times in the traffic: original frame from victim to switch, the copy sent by the switch to the attacker, the copy relayed by the attacker to the switch, and the copy delivered by the switch to other victim machine. Second, pairing is only done between frames of the same size as the modification is done to fields in the headers not the payload which means that size of the two frames of an MITM pair should be the same.
    - **Matching payload of the frame pairs**: The paired frames will be tested for matching of content. The matching starts at an *offset* from the start of frame to skip the parts that are likely to be modified by the attacker. From specified matching offset (MO), a selected portion of the payload is used for comparison. The size of the selected portion is determined by the matching length (ML) parameter as shown in Fig. 4(b). This parameter can be used to tradeoff the accuracy of matching with processing overhead.

V. Experimental Results

A. Experimental Setup

To investigate the performance of the proposed scheme, a series of experiments were conducted in the lab. All the computers used in these experiments have the following specification: Intel core i7-4770 CPU @ 3.4 GHz, 8 GB RAM running 64-bit Windows 7. The switch is Cisco Catalyst 2960G. The setup is similar to that shown in Fig. 2. The two victim computers were used as an ftp server and client. A total of 30 files have been created and stored in the ftp server. Each file has size of 1 MB and contains random data. The ftp client downloaded each file while the attack is going on and a traffic trace was recorded in monitoring station for each file downloaded using WinDump [9]. These 30 traces files were used to study the performance of proposed scheme offline via Python scripts developed for this purpose. The results presented below are the average from these 30 different traces. Three types of MITM attacks were considered: attack initiated by ARP-Poisoning, attack initiated by ICMP-redirect, and attack where that attacker acted as proxy. The first two attacks were launched using Ettercap [10] and for the proxy attack, Squid [11] were used. For ICMP-redirect, a Linux client machine was used since ICMP-redirect does not work very well with Windows machine. Unless otherwise stated, the value of MSI is 2000ms.



(a)

(b)

Fig. 4 Illustration of parameters used in the detection scheme. (a) Matching search interval. (b) Matching offset and length

B. Performance Metrics

The proposed scheme is essentially attempting to find a frame coming from a victim and the corresponding frame that the attacker relays to the other victim (i.e. MITM frame pair). Ideally, the scheme should be able to identify correctly an MITM pair without producing any false positive. To quantify this, let P be the total number of MITM pairs in the sample under consideration (positive condition) and N be the total number of non-MITM pairs out of the total frame pairs processed in the same sample (negative condition). Based on this, we use the following statistical metrics to quantify the performance of the scheme:

- True positive rate (TPR) or sensitivity: the percentage of correctly indentified MITM pairs, i.e. true positive (TP), out of the total number of MITM pairs in the traffic sample ($TPR = TP/P$).
- True negative rate (TNR) or specificity: the percentage of correctly identified non-MITM pairs, i.e. true negative (TN), out of the total number of non-MITM pairs in the traffic sample ($TNR = TN/N$)

For perfect detection scheme, both of these metrics should be 1, i.e. MITM and non-MITM pairs are classified correctly.

C. Numerical Results and Discussion

The TPR chart for the ARP-poisoning-based MITM attack is shown in Fig. 5. The scheme does not detect any MITM frame pairs when the offset is small (MO=10 bytes) and have perfect hit rate for high values of offset. For low values of offset, the comparison starts within the Ethernet header where the address fields in the header are modified by the attacker to divert the LAN traffic to his station and hence a mismatch is expected. The deeper the comparison starts, the more likely that we are dealing with higher layer payload data that is very likely will be the same (especially in passive attack) which result in perfect match and hence the jump in performance.

Fig. 6 shows the specificity of the scheme for various values of matching offset and length for the ARP-based MITM attack. Notice that for low value of offset (MO=10bytes), the scheme have a perfect specificity. We can understand this by noting that for this particular value of offset, the scheme did not detect any MITM pair (Fig. 5) which means that the scheme basically returns negative results for all pairs (i.e. it considers all pairs as non-MITM frame pairs). It follows then that any non-MITM frame pair will be classified as such (i.e. $TN = N$) and hence the perfect specificity. This highlights the need for both TPR and TNR to characterize the performance of the scheme accurately. For large values of offset, the specificity increases with the increase of the matching length. This can be explained as follows. If we take a sequence of n bits from both frames and assume that the each bits is equally likely to take 0 or 1, then the probability that the n-bit sequence from both frames match is the same as the probability that n-bit sequence takes a specific pattern (to match the other frame bits pattern) which is 2^{-n}. Hence, for small matching length, the probability of match is higher than for longer length, i.e. a non-MITM pair may be classified as MITM pair and hence reduces the number of true negative (i.e. reduces TNR).

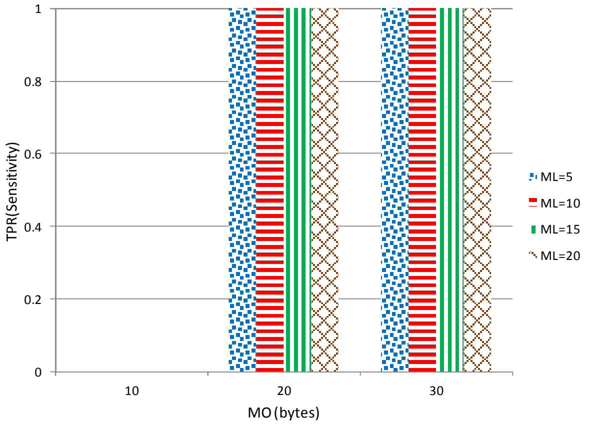


Fig. 5 The TPR (sensitivity) of the proposed scheme for different values of matching offset and length (both in bytes) for ARP-poisoning-based MITM attack.

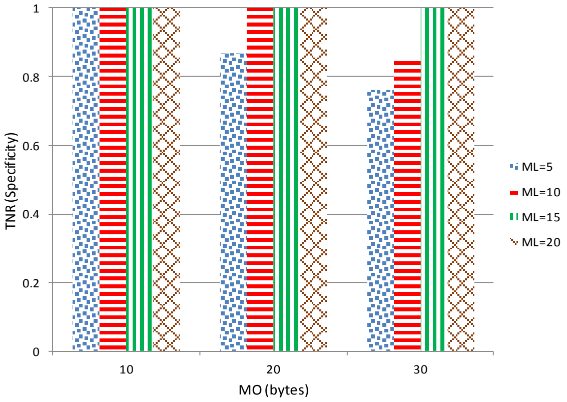


Fig. 6 The TNR (specificity) of the proposed scheme for different values of matching offset and length (both in bytes) for ARP-poisoning-based MITM attack.

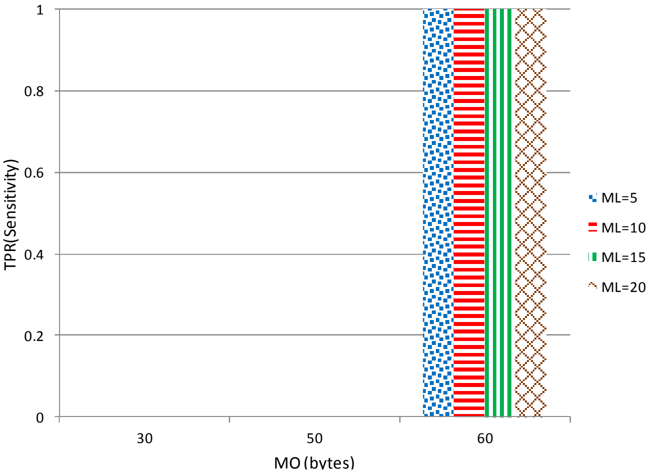
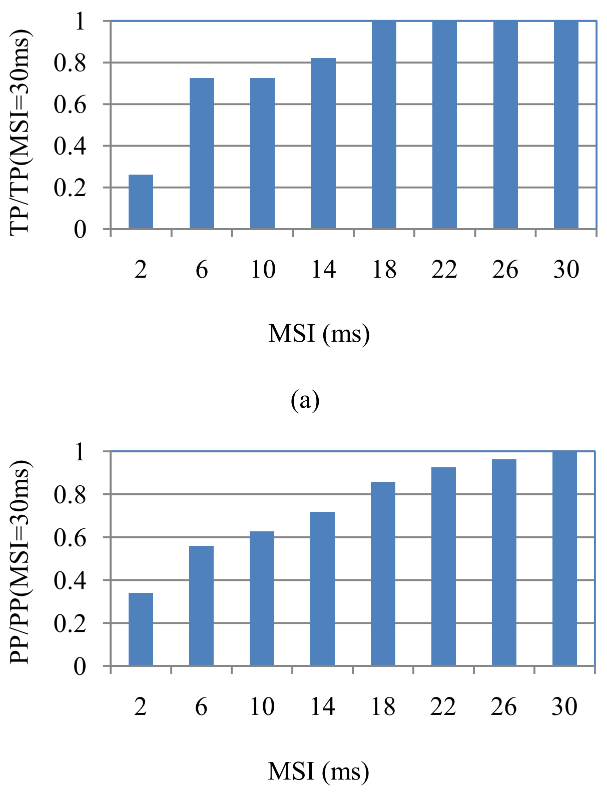


Fig. 7 The TPR (sensitivity) of the proposed scheme for different values of matching offset and length (both in bytes) for proxy MITM attack.

For the attack initiated by ICMP redirect, the results are very similar to that of ARP-poisoning attack which can be explains as follows. In the ICMP redirect attack, the victim is tricked into believing that the attacker presents a better path to the destination and hence the victim sets the Ethernet destination to point to the attacker. The attacker then simply relays this to the other victim by changing the destination address accordingly. Therefore, the difference between the two frames of the MITM pair will be observed over the Ethernet header which the same as in the ARP-poisoning-based attack which explains the similar results.

The TPR of the proxy attack is shown in Fig. 7. Here, it is clear that the matching offset should be larger compared to the previous two attacks to achieve reliable detection performance. This is because the proxy terminates the connection in one side and initiates it on behalf of the client on the other side and hence the higher layers' headers (network and transport) are affected which requires increasing the offset to ensure that the matching is done beyond the part that has been changed. The result for TNR shows that the scheme achieves specificity of almost 1 for all combination of MO and ML considered for TPR (Fig. 7).



(a)

(b)

Fig. 8 Impact of MSI for the ARP poisoning attack (MO=30bytes, ML=20bytes) on total number of: (a) true positive detected (b) pair processed. Results are normalized by the values obtained for MSI=30ms.

The impact of the matching search interval (MSI) on the number of correctly identified MITM pairs (i.e., true positive (TP)) and number of frame pairs processed (PP) is depicted in Fig. 8 for the ARP-poisoning-based MITM attack. The results are normalized by values obtained for MSI=30ms. Here, MO and ML are set to large values to ensure that they lead to an excellent detection performance and hence it enables us to focus on how the variation of MSI impacts the performance. It is clear that the larger the value of MSI, the larger number of MITM pairs identified (Fig. 8(a)). This comes at the expense of having to examine larger number of frames over large time interval. On the other hand, when MSI small, the scheme is restricted to examine frames that exists within a small time interval and hence it more likely to miss an MITM pairs that are separated by more than MSI. For all of these values of MSI, the specificity remains 1 since the classification ability of scheme to detect non-MITM pairs is not affected by whether the two frames of MITM pair are inside the MSI interval or not. It worth noting here that in practical setting, the scheme does not have to detect *every* MITM pair in the traffic stream; it only needs to detect *one* MITM pair to alert the network administrator to the presence of MITM attack.

VI. Conclusions

This paper introduces a scheme to detect MITM attack in switched LAN based on matching payload of frames transmitted within the LAN. The advantage of the proposed scheme is that it focuses on detecting the MITM attack based the symptom of traffic generated in such situation rather than attempting to detect the specific technique that used to launch the attack. The performance of the proposed scheme and impact of its various parameters is studied experimentally. The results show that we can achieve an excellent detection of MITM attack by using large values for both the matching offset (MO) and the matching length (ML). The large value of MO ensures that the matching process starts within the higher layers payload. Unlike the lower layer headers (which is likely to be modified by the attacker), the higher layers payload is likely to be the same in an MITM pair and hence increases the probability of match. Similarly, higher values of ML reduces the misclassification of non-MITM pair as an MITM pair and hence improves the specificity of the scheme.

A drawback of the proposed scheme is that the computational and memory requirements may become high in heavy traffic conditions. Future work will focus on further testing and optimizing the scheme for such operating conditions.

References

[1] K. Kalajdzic and A. Patel, "Active Detection and Prevention of Sophisticated ARP-Poisoning Man-in-the-Middle Attacks on Switched Ethernet LANs," in *WDFIA 2011*, pp. 81-92.

[2] M. Carnut and J. Gondim, "ARP spoofing detection on switched Ethernet networks: A feasibility study," in *Proc. 5th Simposio Seguranca em Informatica*, 2003.

[3] C. L. Abad and R. I. Bonilla, "An analysis on the schemes for detecting and preventing ARP cache poisoning attacks," in *ICDCSW'07*, pp. 60–60.

[4] N. Hubballi, S. Biswas, S. Roopa, R. Ratti, and S. Nandi, "LAN attack detection using discrete event systems," *ISA Trans.*, vol. 50, no. 1, pp. 119–130, 2011.

[5] V. A. Vallivaara, M. Sailio, and K. Halunen, "Detecting Man-in-the-middle Attacks on Non-mobile Systems," in *Proc. 4th ACM Conf. on Data and Application Security and Privacy*, New York, NY, USA, 2014, pp. 131–134.

[6] J. Belenguer and C. M. T. Calafate, "A low-cost embedded IDS to monitor and prevent Man-in-the-Middle attacks on wired LAN environments.," in *SECURWARE*, 2007, pp. 122–127.

[7] Z. M. Fadlullah, T. Taleb, A. V. Vasilakos, M. Guizani, and N. Kato, "DTRAB: combating against attacks on encrypted protocols through traffic-feature analysis," *IEEEACM Trans. Netw. TON*, vol. 18, no. 4, pp. 1234–1247, 2010.

[8] I. Dacosta, M. Ahamad, and P. Traynor, "Trust no one else: Detecting MITM attacks against SSL/TLS without third-parties," in *Computer Security–ESORICS 2012*, Springer, 2012, pp. 199–216.

[9] https://www.winpcap.org/windump/

[10] http://ettercap.github.io/ettercap/

[11] http://www.squid-cache.org/