JAKU: Comp. IT., **Vol. 1,** No. 1, pp: 55-81 (2012 A.D. / 1434 A.H.)

DOI: 10.4197 / Comp 1-1.3

Intrusion Detection System for Misbehaving WLAN Users

Sami M. Alesawi, Reda M. Salama and **Abdulrahman H. Altalhi**

Department of Information Technology, King Abdulaziz University
Jeddah, Saudi Arabia

salesawi,rkhalia,ahaltalhi@kau.edu.sa

Abstract. Wireless networks in the lives of people at work, at home, and in public places, plays a decisive role. Given the widespread demand for wireless networks, providers deploy wireless local area networks (WLAN) to provide access to broadband Internet Within the range of a public wireless LAN hotspot such as in airports or hotels, users can access their e-mails and browse the Internet either for free or, most often, against a fee. However, as the number of users rises, so does the risk that users may misbehave. Misbehaving users can to a large extent increase their share of bandwidth at the expense of other paid users by slightly modifying the driver of the network adapter. As the use of such networks grows, users will demand secure yet efficient, low-latency communications. Intruders' detection is one of the key techniques that can be used to protect a network against outsiders. Many Intrusion Detection Systems (IDSs) have been designed for wired networks. Unfortunately, most of these IDSs do not give the expected results when used with wireless networks and are especially poor at addressing the Media Access Control (MAC) layer problem. In this sheet, we present the design and implementation of an IDS tool that is chosen for WLANs and addresses misbehavior at the MAC layer properly.

Keywords: IDS, WLAN, MAC layer, Misbehavior

I. Introduction

As long as the revenue from hotspot use is increasing, the widespread deployment of IEEE 802.11 hotspots will continue to grow. However, the commercial use of these networks has already revealed a set of associated security-related problems[10]. One major challenge that requires more exploration by the research community is Media Access Control (MAC) layer selfish behavior, which has not been given adequate attention from the research community. The focus has usually been on the malicious behavior of users in the Wireless Local Area Network (WLAN) field[16] and has thereby captured only one side of the problem.

The IEEE 802.11 standards for wireless networks generate inherent general vulnerabilities that are not easily preventable. Moreover, the MAC layer protocol in such networks has its own vulnerabilities. The architectural design of wireless networks in distributed and separated user communities' raises issues of compliance with standard network protocol rules. In addition, users are clustered in communities that are defined on the basis of proximity, a common service or some other common interest. Because such communities can operate without a central supervising entity, no notion of trust can be presupposed. Furthermore, what made wireless network adapters and devices easily programmable was the increasing level of sophistication of the design of protocol components, together with the requirement that protocols be flexible and readily reconfigurable. As a result, it is feasible for a network user (node) to tamper with software and firmware, modify its wireless interface and network parameters and ultimately abuse the protocol. This situation is called protocol misbehavior. The goals of misbehaving stations range from the exploitation of available network resources for the user's own benefit to network disruption.

A misbehaving station in a wireless network can intentionally misuse the MAC protocol to gain bandwidth at the expense of other stations. This misbehavior can be done by changing the network driver, either by manipulating some of the standard parameters or by departing from standard communication procedures. The many benefits for the misbehaving station are as follows[18]:

- It can result in significant bandwidth gains because it directly affects the wireless medium. Therefore, it is more efficient than misbehavior at the network and transport layers.

- It is hidden from and independent of the upper layers and hence cannot be detected by any mechanism designed for those layers. However, it can be combined with upper layer misbehavior to increase the impact.

- It is always usable because all wireless stations use the same IEEE 802.11 MAC protocol.

The solution to the problem is to provide a reliable means of detecting such instances of misbehavior that would activate network defense and response mechanisms and isolate the misbehaving station.

This paper presents an IDS alarm tool for selfish behavior in IEEE 802.11 MAC layers in WLANs[11, 7]. It will provide a general bound for worst-case attack scenarios in wireless networks when one or many intelligent adversaries exist. We will also preview modeling approaches and indicate how to extend the study of such behavior for detection systems.

The organization of this paper is as follows:

- The next section discusses existing work in the areas of IEEE 802.11 MAC misbehavior and the detection of such attacks.

- Section III presents an overview of (i) the IEEE 802.11 MAC layer in wireless networks and (ii) TCP congestion control at the transport layer, which is essential to better understand misbehavior.

- In Section IV, we will explore MAC layer selfish behavior, focusing on techniques of misbehavior used by cheaters.

- In Section V, we present in detail the measures that can be used to address selfish behavior in a way that is transparent to the operation of the network, unlike other proposed techniques that require modifications to the existing standard.

- Section VI will describe the implementation of the IDS alarm tool used to detect MAC cheating.

- Conclusions will be presented in Section VII.

II. Related Studies

Even if protocol misbehavior has been studied in various scenarios for most communication layers and using several mathematical frameworks, the problem of Media Access Control (MAC) layer misbehavior is relatively new and unexplored in the literature. Various solutions to routing layer misbehavior have been proposed for ad hoc

BSS (Basic Service Set) networks that are part of WLANs. However, the problem that we consider in this paper is how to develop the proper solution to MAC layer misbehavior, which is too different for those proposed solutions to apply here. This section presents those solutions in brief.

Kyasanur and Vaidya[20] address MAC layer misbehavior using detection and correction mechanisms. Their main aim is to let the receiver assign and send back-off values to the sender in the Clear to Send (CTS) and Acknowledge (ACK) frames and then use them to detect potential misbehavior. The latter involves the use of a correction scheme that adds a penalty to the next back-off that is a function of the observed misbehavior. This solution is efficient in addressing MAC layer misbehavior but has the following disadvantage:

- First, it requires modifications to the IEEE 802.11 MAC protocol that are incompatible with the current standard. Such an approach is not practically feasible and indicates the incompatibility of the solution with the current standard.

- It gives control to the receiver over the sender by making the receiver assign back-off values to the sender in both the detection and the correction schemes. Hence, the proposed approach creates the possibility of new types of misbehavior in the MAC layer, including misbehaving receivers and collusion between the sender and the receiver.

- It creates communication and computation overhead if used with WLANs. The first is due to the addition of new frame header fields, and the second is due to the detection and correction schemes used to compute back-off values and, in some cases, penalties for each individual frame of the sending station.

- This technique considers only stations with backlogged UDP traffic in attempting to detect misbehaving. If the misbehaving station generates traffic with an interframe delay, the latter may result in greater measured back-off than the assigned amount and hence allow the cheater to escape detection.

Konorski[19] considers an ad hoc network in which all stations hear each other and proposes a misbehavior-resilient back-off algorithm based on game theory. Because it requires a new back-off mechanism different from the current standard, this solution is not practical for use with current hotspots. It is worth noting that neither of the previous two studies implements the proposed algorithm in a real-world example.

In[18], the focus was on MAC layer misbehavior in wireless hotspot communities. The authors propose a sequence of conditions used to test the extent to which the protocol parameters have been manipulated. The advantage of the scheme is its simplicity and ease of implementation. This study was an important source of inspiration for this paper.

Intrusion detection systems[6] are also relevant to MAC layer issues, although they handle security flaws rather than protocol misbehavior. A commercial example of these systems is AirDefense Guard, in which distributed sensors placed near APs monitor the wireless medium and send reports to a central server.

III. Background

This section presents a survey of a few important background topics, including the following:

(a) How the MAC layer works according to the IEEE 802.11 standard,

(b) How the TCP (transmission control protocol) at the transport layer works to solve the problem of traffic congestion, and

(c) The difference between the ACK of the MAC layer and the ACK of the TCP belonging to the transport layer. The above subjects must be focused on to develop a better understanding of MAC misbehavior.

A. IEEE Standard MAC Layer

The IEEE 802.11 standard is the most common standard for wireless LANs[8]. This standard specifies a frequent medium access control (MAC) and several physical layer standards for wireless LANs. The MAC layer in the 802.11 standard is the brain of the wireless network. It is responsible for controlling access to the shared air medium by directing the 802.11 physical layer to sense the medium and transmit and receive 802.11 frames. However, the MAC layer uses a coordination function to control access to the medium as in Fig. 1.

Distributed Coordination Function (DCF): The stations use a shared radio channel to send frames. The frame transmission can begin once a station gains access to the medium. The Distributed Coordination Function (DCF) is the fundamental access method used to access the medium under the IEEE 802.11 standard; this is mandatory and based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol (the other part of the CSMA Family: CSMA/CD Protocol is used for wired Ethernet [14]). This coordination function is applied in all stations for use with both ad-hoc and infrastructure network configurations. Under the DCF, IEEE 802.11 stations access the medium and attempt to send frames when there is no other station transmitting. If one of the stations is sending a frame, the other stations should wait until the channel is free.

A Network Allocation Vector (NAV) is the counter resident at each station that indicates the amount of time that the previous frame needed to be sent. This number is used to reserve the medium for the sending station. In order to restrict the MAC layer checks the value of its NAV (Fig. 2). The NAV must be 0 before a station can attempt to send a frame. Prior to transmitting a frame, a station calculates the amount of time necessary to send the frame based on its length and data rate. The station places a value representing this time in the duration field in the header of the frame. When stations receive the frame, they examine this duration field value and use it to set their corresponding NAVs[15].

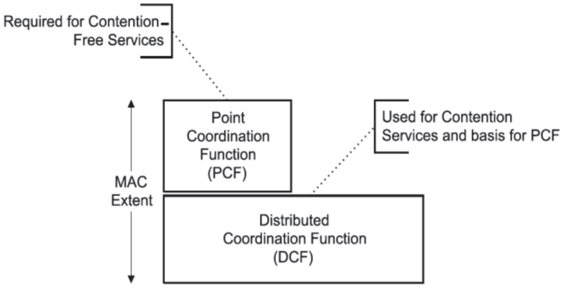

Required for Contention-Free Services
Point Coordination Function (PCF)
Used for Contention Services and basis for PCF
MAC Extent
Distributed Coordination Function (DCF)

Fig. 1. MAC Architecture[8].

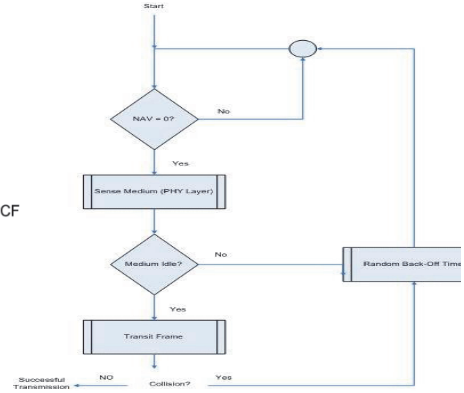


Fig. 2. DCF Offers a Distributed Form of Medium Access[15].

One important aspect of the DCF is that a station must use its random back-off timer[13] to detect that a medium is busy. Therefore, when the channel is in use, the station must wait a random period of time before attempting to access the medium again. This prohibits multiple stations from sending data at the same time. The random delay (referred to as the contention window or "CW") causes stations to wait different periods of time and makes it possible to avoid all stations' sensing the medium at exactly the same time and finding the channel idle, which would cause them to transmit data at the same time. The back-off timer is decreased only when the medium is idle; it is frozen when the medium is busy. After a busy period, the decreasing of the back-off timer resumes only after the medium has been free longer than a Distributed Interframe Space (DIFS). The DIFS is the IEEE 802.11 standard idle time used by stations that need to start a new transmission. A station initiates a transmission when the back-off timer reaches zero. The back-off interval is chosen as follows (1):

$$Integer(22+i * random()) * Slot\_Time. \tag{1}$$

Where *i* is the number of consecutive times a station attempts to send its frames, *random()* is a uniform random variable between 0 and 1, *Integer(x)* represents the largest integer <= x, and *Slot_Time* is a time period.

After each unsuccessful transmission, the contention window takes the next value in the series until it reaches CWmax (Fig. 3). The back-off timer significantly reduces the number of collisions and corresponding retransmissions, especially when the number of active users increases.

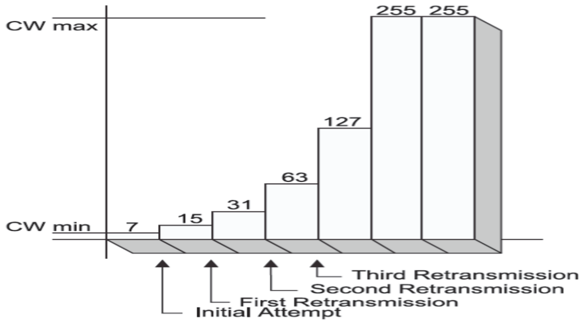


Fig. 3. Exponential increase of CW[8].

In radio-based LANs, a transmitting station cannot listen for a collision while sending data, mainly because the station cannot have its receiver on while transmitting a frame. As a result, the receiving station needs to send an acknowledgement (ACK) if it detects no errors in the received frame after a period of time called the Short Interframe Space (SIFS). The SIFS is the IEEE 802.11 standard short interval used to split transmissions belonging to a single dialogue (RTS-CTS or DATA-ACK). This value is fixed and is calculated in such a way that the transmitting station is able to switch back into the receiving mode decode the incoming frame (2).

$$DIFS = 2 * Slot\_Time + SIFS. \quad (2)$$

If the sending station does not receive an ACK after the specified period of time (SIFS), the sending station assumes that there was a collision (or RF interference) and that it must retransmit the frame (Fig. 4)

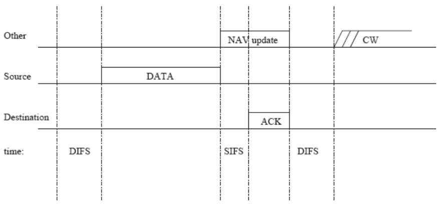


Fig. 4. Basic access CSMA/CA protocol[8].

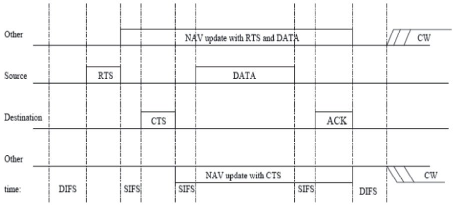


Fig. 5. RTS/CTS exchange in CSMA/CA[8].

The method may be refined under various circumstances to further minimize conflict. Here, the transmitting and receiving stations exchange short control frames [“request to send” (RTS) and “clear to send” (CTS) frames] after determining that the medium is idle and after any deferrals or back-offs prior to data transmission (Fig. 5).

B. TCP Congestion Control[1]

Congestion begins to occur[17], when the amount of network traffic becomes greater than the network can handle. Most of the management of the network congestion is carried out by the TCP because the practical answer to the problem of congestion is to decrease the data rate. TCP tries to achieve the goal of reducing congestion by dynamically controlling the window size. Due to the robustness of modern transmission lines, TCP algorithms monitor timeouts for signs of trouble under the assumption that timeouts are caused by congestion.

[1] This section is a rewrite of the section “6.5.10 TCP Congestion Control” of [17], pp:571-581.

In TCP, a suitable window size must be chosen when a connection is being established between a sender and a receiver. Based on its buffer size, the receiver can specify a window. Problems due to buffer overflow at the receiver will not arise if the sender adheres to this window size. However, the sender may face some troubles due to network congestion. Figure 6 and Figure 7 show this problem hydraulically. Figure 6 presents a small receiving container waiting at the end of a wide pipeline. There will be no losing of water as long as the sender does not send more water than the container can hold. Figure 7 shows that the pipeline capacity, not the container capacity, is the restraining factor. If water comes in excessively and rapidly on the sender side, it will accumulate and some will be lost.

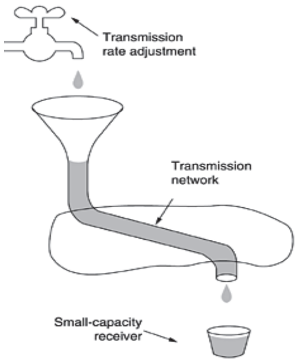


Fig. 6. A fast network and a low-capacity receiver[17].

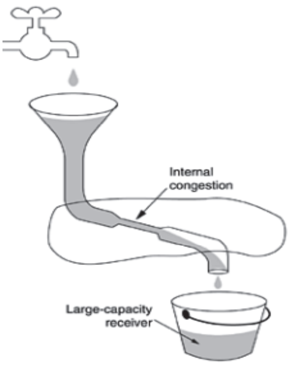


Fig. 7. A slow network feeding a high-capacity receiver[17].

In TCP, to address these problems, two windows must be maintained by each sender: the window granted by the receiver and the congestion window. The number of bytes that may be sent is determined by the minimum of the two windows. TCP then uses the slow start algorithm to utilize theses windows and control the amounted of data to be injected into the network.

IV. Possible Techniques Used by Cheaters

In this section, the focus will be on techniques used to address greedy MAC behavior, which occurs when the cheater changes the operation of the IEEE 802.11 protocol either by not obeying communication procedures or by changing the parameters defined in the standard[18].

Several studies have shown that the traffic flowing through public wireless LANs is primarily TCP traffic (around 90%) and that it is mainly downlink. Therefore, it is important to differentiate between misbehavior techniques according to the type of traffic targeted by misbehaving users, either TCP or UDP traffic. We will describe in the following paragraph greedy attacks on uplink traffic (of both TCP and UDP types) and downlink TCP traffic. Cheater attacks against downlink UDP traffic are much more difficult to perpetrate and will not be considered in this paper.

A. Uplink Traffic

Uplink attacks are easy to establish because a cheater does not require great effort to cheat. To increase the contention windows a greedy station can selectively scramble frames sent by other stations. The potentially targeted frames with scrambling are CTS, ACK and DATA frames.

- When targeting the CTS frames, the cheater hears an RTS frame sent by another station to the access point (or to any other station in the case of an ad-hoc network) and intentionally causes the collision and loss of the corresponding CTS frame to prevent the frame exchange sequence. As a result, the channel becomes idle after the CTS corruption, the station whose CTS has been jammed doubles its contention window, and the cheater has a greater chance of being able to send his data.

- Although jamming ACK and DATA frames does not decrease the data frame transmission time, it doubles the contention window of the ACK destination (*i.e.,* the DATA source) station and consequently causes the latter to select larger back-offs. As in the previous case, the cheater increases his chances of gaining access to the channel.

A greedy station can manipulate protocol parameters to increase bandwidth share:

- When the channel is idle, the station can transmit after the SIFS but before the DIFS.

- When sending RTS or DATA frames, the station can artificially increase the duration value (in the frame headers). Therefore, because the stations in range set their NAVs using that value, they will refrain from attempting to use that channel during that time.

- The station can reduce the back-off time by choosing a small fixed contention window. The back-off will, therefore, always be chosen from this small window.

A cheater may also combine several of the above techniques or adaptively change his misbehavior to avoid detection.

B. Downlink Traffic

In the downlink traffic context, the cheater will attempt to increase the share of traffic sent to him through the AP (or through any ad-hoc station that pretends to be an AP), thus increasing the number of packets assigned to it in the AP's queue. To achieve this goal, the cheater will target the protocols responsible for filling this queue. We can distinguish between two types of sources (*e.g*., Web servers) that send traffic to wireless stations through the AP:

- UDP sources: Because UDP requires no acknowledgement from the receiver and hence cannot be affected by channel conditions, attacking the UDP traffic is pointless.

- TCP source: Unlike in the above case, the TCP traffic rate reacts to the channel conditions using congestion windows and acknowledgements from the receiver. Hence, an attack can be mounted on TCP traffic by exploiting the congestion avoidance mechanism and reducing the source rate until the flow is eventually shut down.

The cheater spends more "effort" to increase his/her share of the bandwidth and from the AP to detect the misbehavior. Because of the closed-loop nature of TCP flows, their impact extends beyond the local area (the hotspot and associated nodes) to affect remote servers. Consider the topology in Fig. 8 and the following typical scenario: two mobile nodes, Bc and Ac, are connected to the Internet via the AP. Ac and Bc download large files from two remote servers, As and Bs, respectively. Both are downloading with the use of FTP/TCP. To increase its downloading data rate, the cheater (Ac) can use the following two techniques to reduce Bs's data rate, therefore freeing more bandwidth for itself at the AP (or at any common bottleneck between the servers and the AP):

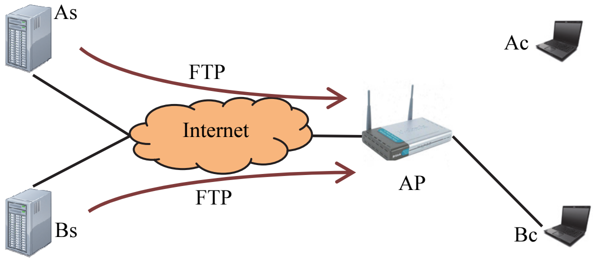


Fig. 8. Stations Ac (cheater) and Bc downloading from servers As and Bs[18].

The Ac jams the TCP-ACKs from Bc to the AP, so they never reach the server Bs. As the TCP-ACKs are lost (jammed), Bs decreases its sending data rate using TCP congestion control and kills the connection. At the AP, Bc's share of the bandwidth decreases, which increases the data rate from As to Ac.

When the above technique is used, the AP can still hear the collisions/jamming and may detect Ac's actions based on the number of retransmissions by Bc. Another option for Ac involves jamming the AP frames that are destined to Bc, therefore reducing Bs's data rate without detection by the AP. However, Ac's packets share the same queue as Bc's packets at the AP. When jammed frames are repeatedly retransmitted by the AP, Ac's packets become delayed in the queue, and Ac's data rate (from As) decreases as well. To prevent retransmissions by the AP and queuing delays, Ac can send forged MAC-ACKs on behalf of Bc for the jammed packets. This also reduces the data rate from Bs. Furthermore, Ac can jam only a portion of the AP's frames to Bc, which allows Ac to save its battery power and makes detection even harder.

V. Components of System Alarm IDS Tool

There are several approaches that might be used to counter misbehavior[7, 18,11], as presented in section IV. The proposed IDS alarm tool is one of suggested solutions. In accordance with the number of possible attacks, the system alarm has a modular architecture, as depicted in Fig. 9. The system alarm needs to be used only on the AP within an infrastructure network as a network-based IDS or in any one of the communicating stations within an ad-hoc network as a host-based IDS. The system alarm periodically collects traffic traces from the active user stations during short intervals of time called monitoring periods.

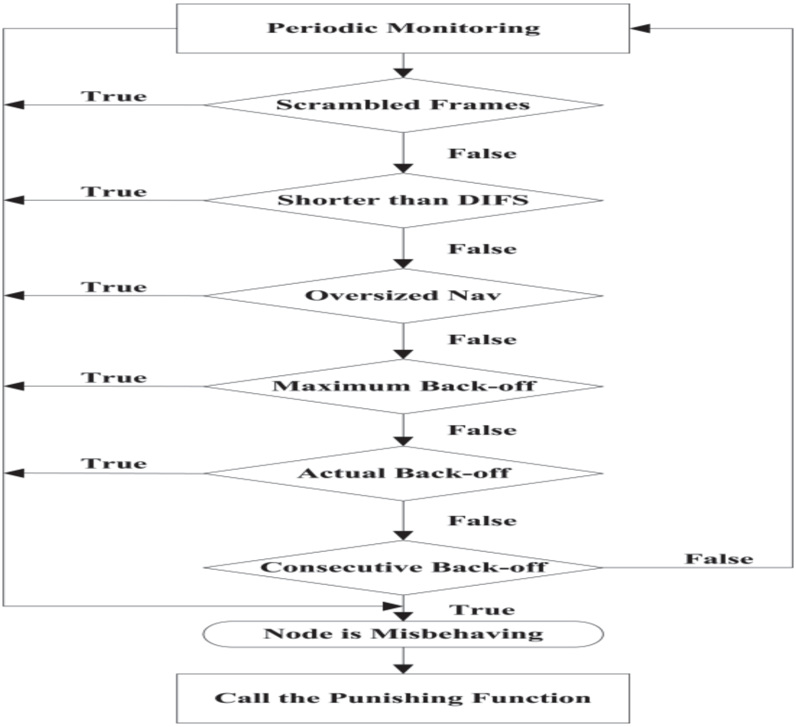


Fig. 9. The Modular Architecture of System Alarm.

A. Detection of the Intended Collisions

The goal of this test is to detect the misbehavior techniques that rely on frame scrambling, the first attacks described in the uplink and downlink in section IV.

Scrambled Frames Detection: to gain a significant share of the common wireless bandwidth using CTS/ACK/DATA scrambling, the cheater must scramble a relatively large percentage of the CTS, ACK, or DATA frames sent by other stations. As a result, the average number of retransmissions will be less than that of other stations and can be detected using (3). (For this test as for the following ones, if the inequality holds, it means that a greedy attack is probably taking place).

$$num\_rtx(M_i) < \Phi \times E_{j \neq i}[num\_rtx(M_j)]. \qquad (3)$$

Where num_rtx(M) is the number of times station M retransmitted the last frame successfully received by the AP. Φ is a tolerance parameter with a value between 0 and 1; it is applied to the average number of retransmissions from all “other” stations, $E_{j\neq i}$.

The IDS alarm tool can detect retransmission by observing repeated sequence numbers in the headers of RTS or DATA frames when the corresponding CTS or ACK frames are scrambled, respectively. Regarding the DATA frames, one might argue that the AP will not be able to identify retransmissions because the frames are scrambled. However, the cheaters cannot scramble the headers of these frames; otherwise, they cannot know whether a given frame is destined for them.

The attacker's identity can be derived from the number of retransmissions, because we assume a rational attacker who jams others frames only when necessary. The attacker cannot change this number to cheat because the sender (the AP) will respond to a wrong sequence number by discarding the frame (if the number is not larger than the last recorded one) or sending a frame out of order (if the number is larger than the last recorded value), depending on the specific wireless card. We also assume that the attacker cannot change the MAC address of his station because an authentication mechanism (*e.g.*, WPA or IEEE 802.11i) makes it impossible to use arbitrary MAC addresses.

A potential cause of false positives (which lead a system alarm to detect station misbehavior when there is none) when this test is used might be poor channel conditions that lead to frame loss and retransmission. The Φ tolerance parameter must be chosen carefully to avoid this pitfall.

B. Detection of Manipulated Protocol Parameters

The following paragraphs will address the misbehavior techniques used to alter protocol parameters.

Shorter than DIFS Detection: The AP can monitor the idle period after the last ACK and identify any station that transmits information before the required DIFS period. After having observed this misbehavior repeatedly for several frames from the same station, the AP can make a reliable decision using the "shorter than DIFS" test of (4).

$$Idle\_time\_after\_ACK(M_i) < DIFS. \qquad (4)$$

Oversized NAV Detection: By measuring the actual duration of a transmission (including the DATA, ACK, and optional RTS/CTS) and comparing it with the duration field value in the RTS or DATA frame header, the AP can identify a station that regularly sets the duration (and therefore the NAV of the listening stations) at very large values. In (5), tolerance parameter A (greater than 1) ensures that the AP does not mistakenly accuse well-behaved stations.

$$A \times actual\_duration(M_i) < duration(M_i). \qquad (5)$$

C. Back-off Manipulation

In the following, we will address the misbehavior techniques associated with back-off manipulation, which are the easiest to implement and the hardest to detect.

Maximum Back-off Detection: because the IEEE 802.11 protocol selects back-offs randomly from the range [0, CW - 1] (where CW depends on the number of retransmissions), the maximum selected back-off $\max_{bkf}(S_i)$ over a set of frames sent by a given station should be greater than or equal to CW_{min}-1 if the number of samples is sufficiently large. In (6), this property is used to examine stations whose maximum back-off over a set of samples is smaller than a threshold value $threshold_{maxbkf}$.

Clearly, a tradeoff exists between the number of samples and the threshold; if the IDS system increases the threshold (its largest value is CW_{min}), it must increase the number of sampled back-offs to derive more distinct values and thus avoid false positives. For the designed IDS alarm tool, a threshold equal to $CW_{min/2}$ is used; thus, the test works if the reduced contention window is within the range [0, $CW_{min/2}$ -1].

$$max_{bkf}(Si) < threshold_{maxbkf}. \qquad (6)$$

Regrettably, detection may be easily avoided by a clever cheater. The cheater can make the monitor observe in every sample at least one back-off value larger than or equal to the threshold, and channel conditions can also yield a similar result, thus making the check fail. Therefore, the maximum back-off check is only auxiliary to the following two tests.

Actual Back-off Detection: This test involves measuring the actual back-off as in (7). The main test procedures can be summarized as follows:

- If between two transmissions from station M there are no collisions, we assume that M spent all of its idle time backing off (although it may be simply part of the M's interframe delay if it is transmitting at low data rates). Then the back-off can be estimated by computing the sum as illustrated in Fig. 10.

- If a collision occurs, it may be more difficult to determine the identities of the senders of the colliding frames, which are, therefore, the stations who are measured actual back-off should be updated. For the sake of simplicity, collisions are simply not taken into account; in the case of collisions, neither the current back-off nor the next one is measured for any station.

$$B_{ac}\,[M_i\,] < \alpha_{ac} \times B_{acnom}. \tag{7}$$

Where B_{ac} [M_i] denotes the average actual back-off of station M_i (as observed by the AP). B_{acnom} is the nominal back-off value, which is equal to the average back-off of the AP, assuming that it has enough traffic to compute this value. The α_{ac} ($0 < \alpha_{ac} \leq 1$) parameter is configurable according to the desired true positive (correct detection) and false positive (wrong detection) percentages (for example, a value of αac = 90% may be used in our proposed tool).

Because it collects no data during collisions, the actual back-off test measures back-offs that are selected only from the [0, CW_{min}-1] range. This test fails to detect misbehavior if the cheater creates interframe delays (*e.g.*, for a TCP source using congestion control). In fact, the test measures these delays instead of back-offs because it adds up the idle periods between transmissions from the same source. The solution to this problem is provided by the consecutive back-off test.

In Figure 10, transmissions from M are interleaved with one or more transmissions from other nodes (including the AP). In addition to including the DATA frame, the transmissions also include all of the control frames (*e.g.*, RTS, CTS, and ACK), as well as the interleaving idle periods of the SIFS and DIFS. The measured value is the sum of all idle intervals (not including the interframe spaces) between two transmissions from M[7].

Consecutive back-off: Figure 11 presents the consecutive back-off test. The test on (8) works for sources with interframe delays. In practice, this includes most TCP sources (the delay is typically due to the congestion control of TCP). The actual back-off test for these sources does not yield the correct values (as explained in the previous paragraph) and consequently cannot be used to detect potential cheating.

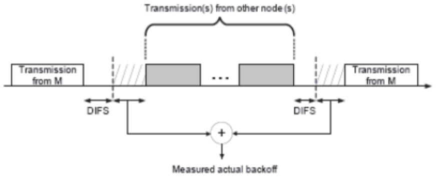


Fig. 10. Measurement of the actual back-off[7].

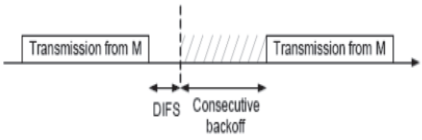


Fig. 11. Measurement of consecutive back-off taken only between consecutive non-interleaved transmissions from M[7].

Let us consider a station, M, sending TCP traffic. We assume that there is sufficient traffic from other sources on common channels that between two frames sent by M and separated by a transport layer delay, there should be at least one interleaving frame from another station. For this reason, if the AP observes two consecutive non-interleaved frames from M, it can consider the idle time between them as only a back-off in addition to the mandatory DIFS. These consecutive frames are the result of channel contention that may force M to queue packets at the MAC layer, even if they are separated by a delay at the upper layers. In this situation, M will benefit from the back-off because it will free its MAC layer queue. Therefore, the IDS alarm tool can collect significant samples from the back-off values chosen by M; we call these samples consecutive back-offs.

The above assumption regarding the traffic level is realistic. Actually, if the traffic on the channel is sufficiently low to invalidate this assumption (*i.e.*, if M can send consecutive non-interleaved frames separated by a delay in addition to the back-off and DIFS), cheating will be pointless as reducing the back-off does not affect delays at the upper layer. Misbehavior detection will not be necessary in such a case.

$$B_{co}[M_i] < \alpha_{co} \times B_{conom}. \quad (8)$$

As in the previous test, the average of the collected values $B_{co}[Mi]$ is compared to a fraction α_{co} of the nominal value B_{conom} ($0 < \alpha_{co} \leq 1$; $\alpha_{co} = 90\%$ may be used in the IDS alarm tool). The latter is the average consecutive back-off of the AP if sufficient data are available.

VI. System Alarm IDS Implementation

We have implemented the IDS alarm tool to prove the need for the proposed detection system and its efficiency. The discussion of the IDS system has been divided into three categories, each one of which has a particular importance to our understanding of the developed IDS system. The first will define the desired types of data, the second describes how to attain these data and the third addresses how to analyze these data to detect the probability of misbehavior. These three categories will be presented below.

A. The Needed Data

It is necessary to describe the types of data desired and differentiate them from the wide variety of available collected data before starting the data collection stage. This ambiguity degrades system performance and makes it difficult to know where to begin the analysis stage. Therefore, discarding the unwanted data is necessary to ensure good results. This section examines in detail only three of the presented techniques, which is more than sufficient for the purposes of this paper. Of the previously described techniques, the three selected for use in the implementation phase are as follows:

- DIFS misbehavior
- Max Back-off misbehavior
- Scrambled frames misbehavior

First, it is necessary to decide which types of data are required to detect the use of these three chosen techniques. All of the different types of frames (DATA, ACK, RTS and CTS) are divided into frame headers plus the included data to be delivered (for the DATA frame only) and the Frame Check Sequence (FCS). The frame headers include a variety of information in many fields with size differences measured by octets of 8 bits each (Fig. 12).

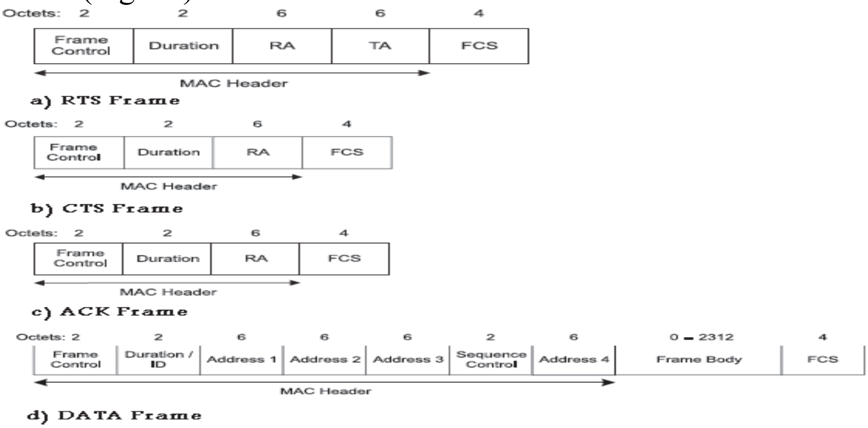


Fig. 12. Types of Frame Format[8].

B. The Needed Data for DIFS Misbehavior

Inequality (4) is used to discover shorter than DIFS misbehavior. In (4), there are two sides; the first side is:

1. The idle time after the acknowledgement of a previous transmission by any transmitting station and will be derived from multiple fields found in the frame header of the involved transmitted frames.

On the second side:

2. DIFS is the defined standard idle time used in transmissions. The address field for the transmitting station is needed to distinguish the transmitting stations from each other and identify the misbehaving one. The second desired field will be used to determine the type of frame: the Frame Control field, which identifies the function of the frame.

There are three types of frames: control, data, and management. A management frame is used mainly to request connection or disconnection to/from the WLAN network. With these frames, the IDS system will be used to differentiate the misbehaving station from the stations that have already joined the network. Therefore, the management frame type will not be used in the system implementation process. The control frames will be the ACK, RTS and CTS frames, whereas the data frame is DATA. All of these types of frames (DATA, ACK, RTS and CTS) need to be available to the IDS system. The frame-receiving time is required to calculate the idle time spent before starting the new transmission. Additionally, the bit rate is needed to calculate the transmission time associated with the frames. The bit rate can be established from the preamble frames added at the physical layer. In addition, the recording of the transmission order in the time scale is definitely important because this information is used to decide which transmissions come first and which ones come later.

C. The Needed Data for the Max Back-off Technique

Inequality (6) showed how to detect this type of misbehavior. All of the abovementioned types of information required to detect DIFS misbehavior are also required to detect the use of the Max Back-off technique. The threshold value is the parameter defined by the network administrator and should be chosen carefully to avoid false positives.

D. The Needed Data for the Scrambled Frames Technique

Inequality (6) is used to detect the different scrambling techniques. The Mac Address field is required to identify transmitting stations, and it is also necessary to increment the counter of the retransmission. Every time a retransmission occurs, this can be detected from the sub-control Retry field in the Frame Control field.

E. Obtaining the Needed Data

To study the performance of the proposed IDS alarm tool, we have developed a traffic generation system that generates wireless network traffic. Both types of regular traffic and the traffic including the misbehaving station's traffic are included in this generation system. The three chosen misbehavior techniques are definitely covered by this software. The system was developed using the Visual Basic 6 code because of its ability to be used with Microsoft Access databases. Using Visual Basic 6 has allowed us to design an automated system that extends from traffic generation to data collection and then save the data in a Microsoft Access database. This, in turn, has facilitated the third stage of identifying misbehavior.

This software will make it possible to examine misbehavior in detail using the three techniques. All types of scrambling misbehavior using the CTS, DATA and ACK scrambling methods will be covered in this simulation, as well DIFS misbehavior and the Max Back-off technique. The basic model used in this simulation is a maximum-length frame. It is assumed that the host operating system limits the outgoing frame size to 1,500 bytes. IEEE 802.11 permits the use of much larger frame sizes, but this has not traditionally been true of client products. Most access points connect to existing networks via the Ethernet, and therefore, the payload size is limited to the maximum Ethernet payload size. (This simple precaution is required to obtain Wi-Fi certification.). The ability to change this size was considered in the design of this simulator software. In addition to the payload data, there are 36 additional bytes of data added in the encapsulation process. The IEEE 802.11 MAC header adds 28 bytes of data for various control and management functions and to detect and address errors. An additional eight bytes are added by the Sub-Network Attachment Point (SNAP) encapsulation header to identify the network layer protocol[8]. The SNAP is an IEEE-defined layer 2 encapsulation header format for Ethernet and similar network packets. It is used at the logical link control layer of the IEEE 802.11 communications protocol stack as a trailer for the IEEE-defined LLC header in so-called "LLC/SNAP encapsulation". The total size of the MAC payload for the TCP data segment is 1,536 bytes (12,288 bits).

The IEEE 802.11g standard is significantly faster than IEEE 802.11b, and in a network consisting only of IEEE 802.11g clients, it is even slightly faster than IEEE 802.11a. However, to ensure backwards compatibility with legacy 802.11b clients, the IEEE 802.11g mechanisms will change to those of the IEEE 802.11b if there is at least one IEEE 802.11b client. In addition, most of the 802.11 cards in existence are designed based on the IEEE 802.11b standard. For these reasons, we chose the IEEE 802.11b standard with a data frame size of 1500 bytes and with the optional RTS/CTS as bases for the traffic generation system simulator.

The IEEE 802.11b calculations will be done first, at which point we will compare the other results to those for 802.11b. The basic timing numbers for IEEE 802.11b are as follows:

- SIFS = 10 µs
- Slot time = 20 µs
- DIFS = 2 x Slot time + SIFS = 50 µs

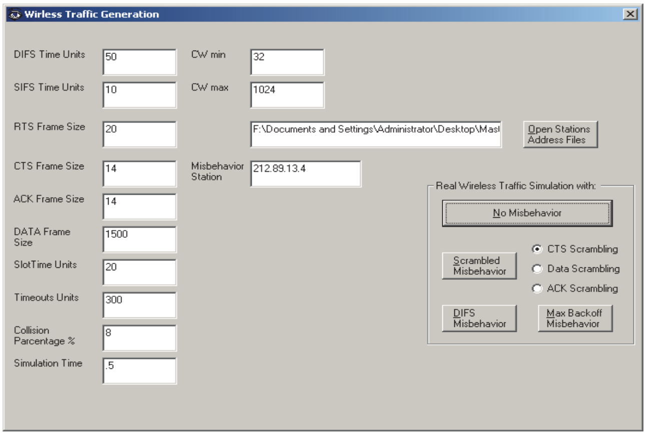
The IEEE 802.11b physical layer requires that a preamble be pretended to every frame before it is transmitted. That preamble may be either the traditional "long" preamble, which requires 192 µs for transmission, or an optional "short" preamble, which requires only 96 µs[8]. It is obligatory that the system support the use of the long preamble, which is the default setting on most devices. Therefore, in the interest of reducing the number of calculations included in this paper, only calculations using the long preamble will be presented.

IEEE 802.11b running at the maximum speed divides data up into 8-bit symbols. There are 1,536 8-bit blocks in the DATA segment. Encoding the MAC frames is easy. IEEE 802.11b divides the MAC frame into a series of 8-bit "symbols" and then transmits 1.375 million symbols per second. We sum the individual components of the transmission to obtain the total duration (Table I).

Table 1. Time Calculation of the Frame Exchange.

Type of Transmission	Required Time
DIFS	=50 µs
RTS	= 192 µs + 20/1.375 MBps = 192 + 15 µs = 207 µs
SIFS	10 µs
CTS	= 192 µs + 14/1.375 MBps = 192 µs + 11 µs = 203 µs
SIFS	10 µs
IEEE 802.11 Data	= 192 µs + 1536/1.375 MBps = 192 µs + 1,118 µs = 1,310 µs
SIFS	= 10 µs
IEEE 802.11 ACK	= 192 µs + 14/1.375 MBps = 192 µs + 11 µs = 203 µs
Frame exchange total	= 2,003 µs

The table results show that each transmission requires 2,003 µs assuming no back-off (the minimum back-off time is equal to zero). All of the above calculations were necessary to understand the use of the generation system simulator algorithm[21, 22], which will be described next (Fig. 13):



Wirless Traffic Generation

DIFS Time Units: 50
SIFS Time Units: 10
RTS Frame Size: 20
CTS Frame Size: 14
ACK Frame Size: 14
DATA Frame Size: 1500
SlotTime Units: 20
Timeouts Units: 300
Collision Parcentage %: 8
Simulation Time: .5
CW min: 32
CW max: 1024
F:\Documents and Settings\Administrator\Desktop\Masl
Misbehavior Station: 212.89.13.4

Real Wireless Traffic Simulation with:
● CTS Scrambling
○ Data Scrambling
○ ACK Scrambling

Fig. 13. Wireless Generation Traffic System.

First, the algorithm defined a public queue of stations with Maxqueue equal to 10 stations. The station class consists of the following:

- Address: the station address.
- Retrans: the number of frame retransmissions for any station.
- BitRates: the bit rate of any station transmission, chosen randomly according to the random algorithm.
- Bkoff: chosen randomly and used to determine the chosen back-off used in a transmission.

The queue class consists of the following:

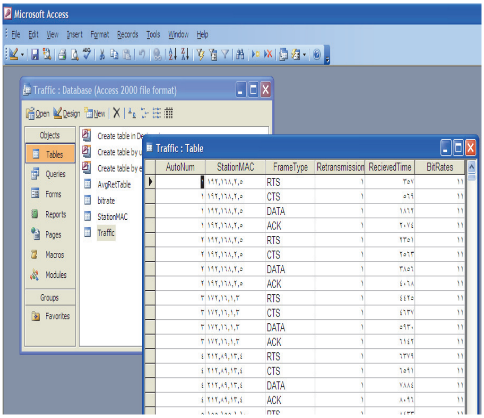
- Queued (Maxqueue +1): an array of Station Classes.
- Front: indicates if the queue is full or is empty.
- Rear: indicates if the queue is full or empty.
- Lower: the lower limit of the queue.
- Upper: the upper limit of the queue.

The simulation used the queue array as a FIFO link list queue, so the transmissions are processed first-come, first-served unless a collision occurs. If a collision occurs, then the current station transmission parameters will be changed (*e.g.*, the back-off parameter), and the information to be transmitted will be put at the end of the queue. For the random process, the Poisson distribution is used because it simulates real-life action. Every time the simulation runs, it is possible that collisions will occur in a random manner. The default parameter chosen for collisions is a 5% probability. For any particular type of misbehavior, the parameters governing the traffic to be generated will be changed during the simulation process, and the resulting database will reflect the appropriate traffic (Fig. 14).

F. Detection of Misbehavior

When the appropriate thresholds are selected for the identification of the misbehaving station as described in section V, one can simply run the detection algorithm by clicking the “analyze” button (Fig. 15). Then the algorithm is calculated by opening the generated traffic database, checking every station using the approved tests (the DIFS, maximum back-off, and CTS/DATA/ACK scrambling misbehavior tests) and conducting the required calculations for time and the other factors according to the previous equations in section 5 used to discover the misbehaving station.

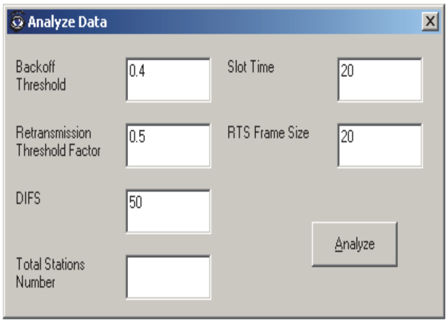
If the process reveals any type of misbehavior, then an alert is generated. The Alert window will appear, showing the address of the misbehaving station and the particular misbehavior (Fig. 16-18).


Microsoft Access
File Edit View Insert Format Records Tools Window Help
Traffic : Database (Access 2000 file format)
Open Design New
Objects
Tables
Queries
Forms
Reports
Pages
Macros
Modules
Groups
Favorites
Create table in Design view
Create table by using wizard
Create table by entering data
AvgRetTable
bitrate
StationMAC
Traffic

Traffic : Table

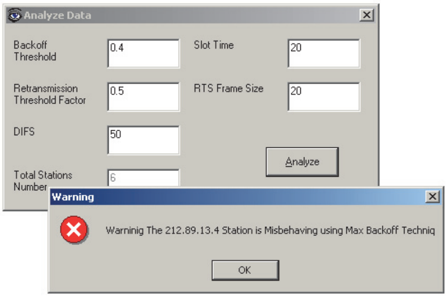
AutoNum	StationMAC	FrameType	Retransmission	RecievedTime	BitRates
١	١٩٢,١٦٨,٢,٥	RTS	١	٣٥٧	١١
١	١٩٢,١٦٨,٢,٥	CTS	١	٥١٩	١١
١	١٩٢,١٦٨,٢,٥	DATA	١	١٨٦٢	١١
١	١٩٢,١٦٨,٢,٥	ACK	١	٢٠٧٤	١١
٢	١٩٢,١٦٨,٢,٥	RTS	١	٢٣٥١	١١
٢	١٩٢,١٦٨,٢,٥	CTS	١	٢٥٦٣	١١
٢	١٩٢,١٦٨,٢,٥	DATA	١	٣٨٥٦	١١
٢	١٩٢,١٦٨,٢,٥	ACK	١	٤٠٦٨	١١
٣	١٧٢,١٦,١,٣	RTS	١	٤٤٢٥	١١
٣	١٧٢,١٦,١,٣	CTS	١	٤٦٣٧	١١
٣	١٧٢,١٦,١,٣	DATA	١	٥٩٣٠	١١
٣	١٧٢,١٦,١,٣	ACK	١	٦١٤٢	١١
٤	٢١٢,٨٩,١٣,٤	RTS	١	٦٣٧٩	١١
٤	٢١٢,٨٩,١٣,٤	CTS	١	٦٥٩١	١١
٤	٢١٢,٨٩,١٣,٤	DATA	١	٧٨٨٤	١١
٤	٢١٢,٨٩,١٣,٤	ACK	١	٨٠٩٦	١١

Fig. 14. Generation of the Wireless Traffic.


Analyze Data

Backoff Threshold: 0.4
Slot Time: 20
Retransmission Threshold Factor: 0.5
RTS Frame Size: 20
DIFS: 50
Total Stations Number:

Fig. 15. Misbehavior Detection System.

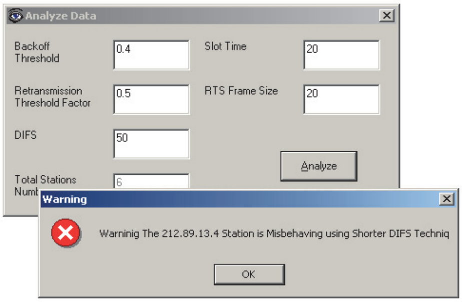

Analyze Data

Backoff Threshold: 0.4
Slot Time: 20
Retransmission Threshold Factor: 0.5
RTS Frame Size: 20
DIFS: 50
Total Stations Number: 6

Warning

Warninig The 212.89.13.4 Station is Misbehaving using Max Backoff Techniq

Fig. 16. Max Back-off Misbehavior Detection.

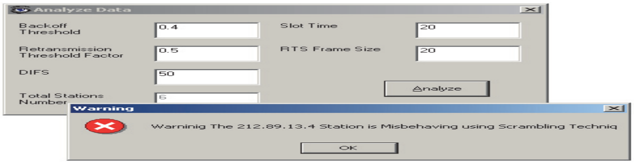

Analyze Data

Backoff Threshold: 0.4
Slot Time: 20
Retransmission Threshold Factor: 0.5
RTS Frame Size: 20
DIFS: 50
Total Stations Numb: 6

Warning

Warninig The 212.89.13.4 Station is Misbehaving using Shorter DIFS Techniq

Fig. 17. DIFS Misbehavior Detection.


Analyze Data

Backoff Threshold: 0.4
Slot Time: 20
Retransmission Threshold Factor: 0.5
RTS Frame Size: 20
DIFS: 50
Total Stations Number: 6

Warning

Warninig The 212.89.13.4 Station is Misbehaving using Scrambling Techniq

Fig. 18. Scrambling Misbehavior Detection.

VII. Conclusion

In summary, to build a highly secure wireless network, we need to deploy intrusion detection techniques (the second wall of defense after intrusion prevention measures like firewalls, encryption and authentication mechanisms[10] in this wireless environment and not just in fixed wired networks. An IDS is not a replacement for a firewall; it is just one layer of security. Although some firewalls have intrusion detection capabilities, they are typically able to detect fewer attacks than a full-fledged IDS. This paper presented a tool based on statistical analysis for intrusion detection in mobile networks as a proposed way of identifying misbehaving users that benefit from the weaknesses of the MAC layer protocol. The key features of the IDS alarm tool are its full compliance with existing standards and its ability to identify cheaters. It does not need any modification to the MAC protocol but instead trigger the occurrence of the misbehave usage of the WLAN.

References

[1] **Lippmann, R.** ***et al***. "Evaluating intrusion detection systems: the 1998 darpa off-line intrusion detection evaluation." In *Proceedings of DARPA Information Survivability Conference & Exposition II*. Hilton Head, SC, USA, Januray 25-27, 2000. 12-26.

[2] **Haines, J., Rossey, L., Lippmann, R.** and **Cunningham, R.** "Extending the DARPA off-line intrusion detection evaluations." In *Proceedings of DARPA Information Survivability Conference & Exposition II*. Anaheim, CA, USA, June 12-14, 2001. 35-45.

[3] **Siraj, A., Bridges, S.** and **Vaughn, R.** "Fuzzy intrusion detection." In *Proceedings of Joint 9th IFSA World Congress and 20th NAFIPS International Conference*. Vancouver, British Columbia, Canada, July 28-28, 2001. 2165-2170.

[4] **Bernardes, M.** and **Santos Moreira, E.** "Implementation of an intrusion detection system based on mobile agents." In *Proceedings of International Symposium on Software Engineering for Parallel and Distributed Systems*. Limerick , Ireland, June 10-11, 2000. 158 - 164.

[5] **Helmer, G., Wong, J., Honavar, V., Miller, L.** and **Wang, Y.** "Lightweight agents for intrusion detection." *The Journal of Systems and Software*, **67**, 2003: 109–122.

[6] **Zhang, Y.** and **Lee, W.** "Intrusion detection in wireless ad-hoc networks." *In Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom '00)*. Boston, MA, USA, August 6-11, 2000: 275-283.

[7] **Raya, M., Hubaux, J.** and **Aad, I.** "DOMINO: A system to detect greedy behavior in IEEE 802.11 hotspots." In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys '04)*. Boston, MA, USA, June 6-9, 2004: 84-97.

[8] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Draft Standard for Information Technology - Telecommunications and Information Exchange Between Systems - IEEE Draft P802.11-REVmb/D8.0 - Local and Metropolitan Area Networks - Specific Requirements - Part 11, IEEE, March 29, 2011: 1-2766.

[9] **Balachandran, A., Voelker, G., Bahl, P.** and **Rangan, P.** "Characterizing user behavior and network performance in a public wireless LAN." In *Proceedings of The International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '02)*. Marina del Rey, CA, USA , June 15 - 19, 2002: 195-205.

[10] **Stallings, W.** *Cryptography and Network Security: Principles and Practice*. 5th edition. Upper Saddle River, NJ, USA: Prentice Hall, 2011.

[11] **Cardenas, A., Radosavac, S.** and **Baras, J.** "Detection and prevention of MAC layer misbehavior in ad hoc networks." In *Proceedings of the 2nd ACM workshop on Security of*

ad hoc and sensor networks (SASN '04). Washington, DC, USA, October 25-29, 2004: 17-22.

[12] **Santamaria, A.** and **Lopez-Hernandez, F.** *Wireless LAN Standards and Applications*. Boston, MA, USA: Artech House, 2001.

[13] **Prasad, R.** and **Munoz, L.** *WLANs and WPANs Towards 4G Wireless*. Boston, MA, USA: Artech House, 2003.

[14] **Casad, J.** *Sams Teach Yourself TCP/IP in 24 Hours*. 4th Edition. Indianapolis, IN, USA: SAMS, 2008.

[15] **Geier, J.** *Wireless networks first-step*. 1st edition. Indianapolis, IN, USA: Cisco Press, 2004.

[16] **Edney, J.** and **Arbaugh,** W. *Real 802.11 security: Wi-Fi protected access and 802.11i*. Boston, MA, USA: Addison-Wesley Professional, 2004.

[17] **Tanenbaum, A.** and **Wetherall, D.** *Computer Networks*. 5th edition. Upper Saddle River, NJ, USA: Prentice Hall, 2010.

[18] **Raya, M., Aad, I., Hubaux, J.-P.** and **El Fawal, A.** "DOMINO: Detecting MAC Layer Greedy Behavior in IEEE 802.11 Hotspots." *IEEE Transactions on Mobile Computing (IEEE)*, **5**, (12), 2006: 1691 - 1705.

[19] **Konorski, J.** "Multiple access in ad hoc wireless LANs with noncooperative stations." *NETWORKING* (Springer) Vol. 2345/2006 of LNCS (2006): 1141-1146.

[20] **Kyasanur, P.** and **Vaidya, N.** "Detection and handling of MAC layer misbehavior in wireless networks." In *Proceedings of International Conference on Dependable Systems and Networks. San Francisco*, CA, USA, June 22-25, 2003: 173 - 182.

[21] **Wurzinger, P., Bilge, L., Holz, T., Goebel, J., Kruegel, C.** and **Kirda, E.** "Automatically generating models for botnet detection." In *Proceedings of the 14th European conference on Research in computer security (ESORICS'09)*. Saint Malo, France: Springer-Verlag, September 21-25, 2009: 232-249.

[22] **Yen, T.-F.** and **Reiter, M.** "Traffic Aggregation for Malware Detection." In *Proceedings of the 5th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA '08)*. Paris, France: Springer-Verlag, July 10-11, 2008: 207-227.

نظام لكشف التسلل لإساءة تصرف مستخدمي الشبكات المحلية اللاسلكية

سامي العيساوي، ورضا سلامة، وعبدالرحمن الطلحي

قسم تقنية المعلومات، جامعة الملك عبدالعزيز، جدة، المملكة العربية السعودية

المستخلص. تلعب الشبكات اللاسلكية في حياة الناس سواء في العمل، في المنزل، وفي الأماكن العامة، دورًا حاسمًا. ونظرًا للإقبال الواسع للشبكات اللاسلكية، يسعى مقدموا خدمة الشبكات اللاسلكية المحلية (WLAN) لتوفير الوصول إلى شبكة الإنترنت عريضة النطاق داخل نطاق نقطة ساخنة من جمهور الشبكة المحلية اللاسلكية كما هو الحال في المطارات أو الفنادق، وبالتالي يمكن للمستخدمين الوصول إلى البريد الإلكتروني وتصفح الإنترنت إما مجانًا، في معظم الأحيان، أو مقابل رسوم. ومع ذلك، فكما يرتفع عدد المستخدمين، كذلك ترتفع المخاطرة من إساءة التصرف من المستخدمين. ويمكن للمسيئي التصرف لحد كبير زيادة حصتهم من عرض النطاق الترددي على حساب غيرهم من المستخدمين الدافعين مقابل الخدمة عن طريق تعديل طفيف على محركات محول الشبكة. وكما ينمو استخدام هذه الشبكات، يطالب المستخدمون بوسائل اتصال آمنة وفعالة وسريعة التجاوب. لذلك فإن عملية الكشف عن المتسللين هي واحدة من التقنيات الرئيسية التي يمكن استخدامها لحماية الشبكة من المتسللين. وقد تم تصميم العديد من أنظمة كشف التسلل لشبكات الاتصال السلكية. ولسوء الحظ، فإن معظم هذه الأنظمة لا تعطي النتائج المتوقعة عند استخدامها مع الشبكات اللاسلكية ونتائجها ضعيفة خصوصا في التصدي لمشكلة طبقة (MAC). وفي هذا البحث، يتم عرض تصميم وتنفيذ أداة IDS لشبكات WLAN، والتي تعالج سوء تصرف في طبقة MAC بشكل صحيح.