Specification-Based Intrusion Detection in WLANs

Rupinder Gill, Jason Smith and Andrew Clark
Information Security Institute, Queensland University of Technology
GPO Box 2434, Brisbane, 4001, QLD, Australia
{rs.gill, j4.smith, a.clark}@qut.edu.au

Abstract

Wireless networking technologies based on the IEEE 802.11 series of standards fail to authenticate management frames and network card addresses and suffer from serious vulnerabilities that may lead to denial of service, session hijacking, and address masquerading attacks. In this paper, we describe and implement a specification-based intrusion detection system for IEEE 802.11 wireless infrastructure networks, which not only provides attack detection, but also implements policy compliance monitoring. The specification used by our intrusion detection system is derived from network protocol state transition models and site security policy constraints. We also perform an experimental and comparative analysis of the technique to assess its effectiveness. The results indicate that the approach is superior at successfully detecting a greater variety of attacks than other existing approaches.

1. Introduction

The notion of monitoring computer systems and networks for malicious activity is long-standing and nowhere is the requirement for preventative approaches to security to be supplemented by a monitoring and detection capability more exigent than in wireless local area networks (WLANs). The broadcast nature of the physical (PHY) layer in wireless networks makes gaining access to the medium a trivial undertaking. Flawed legacy encryption schemes such as wired equivalence privacy (WEP), the forgeabilty of management frames and the spoofability of Media Access Control (MAC) addresses and other frame contents combine to make attacks like eavesdropping, session hijacking and denial of service (DoS) a real threat for WLANs.

While recent enhancements to the IEEE 802.11 standard [8] undoubtedly improve the level of security that preventative techniques can bring to bear on wireless network deployments, two factors must be considered. Firstly, transitional security deployment modes that support the concurrent use of the obsoleted and the current security primitives, leave the network vulnerable to attacks that target the legacy algorithms. Secondly, while a large number of authentication protocols are supported via the extensible authentication protocol (EAP) framework adopted by IEEE 802.11i, few are suitable for use in wireless environments [14]. Both of these factors serve to increase the configuration burden associated with deploying secure wireless networks and motivate the need for techniques to monitor for policy compliance as well as intrusions when wireless networks are deployed in security sensitive environments.

Intrusion detection systems (IDSs) monitor either for evidence of intrusions or for deviations from expected behaviour. When the events of interest to an IDS define the undesirable behaviour, or intrusions, the system is said to be a *misuse-based* IDS. When the events of interest to an IDS are the expected or normal behaviours of the monitored system, with intrusions defined as deviation of the monitored behaviour from this baseline, the system is said to be an *anomaly-based* IDS. Misuse-based IDSs use signatures for attack detection, while anomaly-based systems rely on identifying attacks by detecting deviations from learned *normal* behaviour. Misuse-based systems enjoy the reputation of having a low false positive rate, while anomaly-based systems can generate a large number of false positives. However, unlike anomaly-based systems, misuse-based systems cannot detect novel attacks or attacks for which no signatures are available.

More recently (and as discussed in related work in Section 3), the construction of anomaly-based systems that define models of correct behaviour from explicit policy declarations [4], referred to as *specifications*[1], in-

[1] While some authors will refer to specification-based intrusion detection as a separate category of detection technique to misuse and anomaly detection, we view the specification-based approach as a useful technique for constructing anomaly-based IDSs.

stead of observing and learning correct behaviour during a training phase, represent a promising direction for improving the utility of anomaly-based IDSs and reducing their false positive/false negative rate.

1.1. Our Contribution

In this paper we describe the construction of an IDS, using the specification-based approach, for IEEE 802.11 infrastructure wireless LANs. The correct model of behaviour used by our IDS is formed from a specification that is derived by combining a model of the underlying protocol state machines with the constraints imposed by the security policy of the system. We report on our implementation and demonstrate its effectiveness via experimentation and comparative analysis. In summary, the contributions of this work include:

1. the description of an approach for constructing an anomaly-based IDS, in which the underlying model of correct behaviour is derived from an extensible specification;
2. the application of the anomaly-based IDS to IEEE 802.11 infrastructure wireless LANs; and
3. an experimental and comparative analysis of the technique to assess its effectiveness.

The remainder of the paper is structured as follows. Section 2 describes infrastructure wireless LANs and details the specific threats to them. Related work is covered in Section 3. Our approach to implementing a specification-based IDS and application of this approach to infrastructure wireless LANs is presented in Section 4. Experimental analysis and comparison of our technique with other wireless IDSs is discussed in Section 5. Finally, conclusions and directions for future work are presented in Section 6.

2. Security of Infrastructure WLANs

The technique we describe in this paper has been applied to detect a variety of attacks that are possible in infrastructure WLANs. An *infrastructure* WLAN is one in which the wireless nodes (STAs) communicate via an access point (AP) which is connected to a wider fixed (or infrastructure) network. Our technique focuses on modelling the protocols used between the STAs and the AP.

Numerous security weaknesses existed in early implementations of IEEE 802.11 wireless network components (some of these weaknesses are inherent in the protocols and algorithms specified in the 802.11 standard). The more recent 802.11i enhancements to the MAC layer protocols address many of the original weaknesses. The IEEE 802.11i standard introduces the notion of a *robust security network association* (RSNA) which allows mutual authentication, introduces key management protocols and new data encryption and integrity protocols. RSNA uses 802.1X and EAP for authentication and access control. However, 802.1X and EAP were not designed specifically for wireless environments and many of the authentication protocols supported by EAP are unsuitable for use in WLANs [14]. The 802.11i standard allows RSNA and *pre-RSNA* (i.e., WEP and the original 802.11 authentication) to co-exist in what is referred to as a *transitional security network* (TSN). This means that an STA may be configured to connect to both RSNA and pre-RSNA networks. In this case, a *security roll-back attack* may be employed by an adversary to trick the STA into using pre-RSNA by impersonating association frames from an RSNA-configured AP [7]. Another significant problem that remains, even in 802.11i networks, is that the management frames used by the MAC layer are not authenticated. Neither the original IEEE 802.11 standards, nor the recent IEEE 802.11i standard specify mechanisms for protecting the integrity of management frames, leaving IEEE 802.11 based WLANs vulnerable to management frame spoofing and the associated DoS attacks that such spoofing permits [1]. Even the EAP frames used for authentication in 802.11i networks are unprotected and can be easily used as a means to launch similar attacks against wireless LANs [12].

In this paper we demonstrate that by using a specification based upon the state model of the 802.11, 802.1X and EAP protocols we are able to detect a significant number of the DoS attacks which arise due to unauthenticated management and EAP frames. In addition, we show that by monitoring protocol executions to ensure that certain constraints are met (for example, constraints on the algorithms or authentication protocols used) it is possible to detect violations of organisational security policy.

3. Related Work

Our work combines state transition modelling with constraints derived from a security policy to construct a specification of correct behaviour that can be used in an anomaly-based IDS. The relevance of previous work in the areas of state transition modelling, specification-based intrusion detection, and wireless intrusion detection to our current work is now discussed.

Ilgun et al. [9] proposed the use of state transition analysis for intrusion detection. The work currently presented obviously draws on the foundational ideas presented by Ilgun et al. [9] but instead of using state transitions to model attacks, we use state transitions as the foundation for a specification-based anomaly detection scheme. Specification-based intrusion detection was first suggested by Ko [10] and required the definition of desirable application behaviour (with respect to a site security policy) and the subsequent monitoring of the execution of the application for violations of the specification. Sekar et al. [13] generated a specification-based model of the internet protocol (IP) state machine and combined this with more traditional statistical machine learning techniques for anomaly detection. Our work is similar, in that we use the network protocol specification as the starting point to simplify the generation of our state transition specification, but differs in that we have no need to superimpose statistically based techniques to achieve accurate detection of intrusions. Tseng et al. [2] applied the specification-based approach to detecting intrusions targetting routing protocols in ad hoc networks. The *specification* of the correct behaviours expected are created from a finite state machine model of the routing protocol. While this work serves as a potent stimulant for our work, which applies the specification technique to infrastructure, rather than ad hoc networks, our work differs in that it does not require the use of such strong assumptions. For example, there is no assumption in our work that MAC addresses cannot be forged and in fact, our approach allows us to detect MAC address spoofing attacks.

Current approaches for detecting address spoofing attacks include: the monitoring of MAC frame sequence numbers [15] and verification of MAC addresses against lists of valid users or valid wireless network card vendors [3]. As both MAC addresses and frame sequence numbers can be arbitrarily changed, such approaches are insufficiently robust. While Guo and Chiueh [6] describe how monitoring patterns of sequence number changes can improve the robustness of sequence number monitoring for attack detection, the technique remains unsuitable for policy compliance monitoring.

Techniques based on monitoring the physical characteristics, as viewed by the receiver of radio transmissions, have also been proposed and implemented. Two parameters appear to be useful, including the received signal strength indication (RSSI), which provides a numeric indication of the strength of a received signal, and round trip time (RTT) measurements. Approaches based on RSSI monitoring have been reported by Lim et al. [11] and Gill et al. [5]. Gill et al. also report preliminary success on implementing RTT monitoring [5]. A significant advantage of using physical layer parameters in an IDS is that they are much more difficult to accurately predict, and therefore fabricate (as an attacker may wish to do), given the dynamic nature of the wireless environment. The use of such parameters, however, requires considerable fine-tuning of the deployment environment so that appropriate thresholds can be selected in order to minimise false positives and reduce the likelihood of false negatives.

A significant observation that can be made regarding existing wireless intrusion detection approaches is that all the techniques, unreliable and ineffective as they may be, are focused on attack detection only. None of the techniques surveyed support policy compliance monitoring, an essential and significant contribution of the work presented in this paper.

4. Specification-Based Approach

In this section, we describe how a specification is constructed for our IDS and how such a specification can be used by a wireless sensor to monitor a network for both attacks and compliance with the network security policy.

At a high level, a passive wireless sensor monitors the radio frequency (RF) spectrum and constructs a state transition model for each STA and associated AP that it senses. The sensor is configured with a specification, which encapsulates both the expected state transitions and the constraints of the network security policy. Each frame received by the sensor is evaluated against the specification. If this evaluation reveals that a security constraint is violated, or unexpected state transitions occured, an alert is raised by the sensor.

To motivate and explain our approach in concrete detail, we consider the example of a security sensitive WLAN deployment within an organisation. Given the sensitivity of the deployment and following a risk assessment, an organisational security policy for the use of WLANs is established. In summary, this policy requires that: the wireless network implement a robust security network (RSN), utilising the advanced encryption standard (AES) for link layer integrity and confidentiality protection; port-based network access control is implemented using 802.1X; and the STAs must mutually authenticate with the network using EAP-TLS.

The specification used by our IDS consists of two major components. The first is the state transition model, which describes the expected states a legitimate, policy compliant STA and AP would transition

through when establishing a security association (SA). The second component models the constraints imposed by the detail of the security policy.

4.1. The State Transition Model

In the context of our working example, the state transition model must include components that model the protocols involved in establishing a SA, that is the 802.11, 802.1X and EAP state machines. An overview of the expected exchanges used to establish an SA between an STA and an AP, along with the resulting state transition diagram based on those exchanges, is shown in Figure 1.

The state transition model is much less complex than the contributing state machines. The reasons for this are that the transition model only has to include security relevant states that are passively observable. A significant number of internal, or security irrelevant states do not need to be modelled and are unavailable to a passive observer. A beneficial side-effect of this is that only a limited amount of state must be maintained by the passive monitor thereby reducing the resource requirements of the monitor.

Figure 1 shows the expected incremental order in which the legitimate STA and the AP should exchange frames between each other during SA establishment and the order in which their state transitions should occur. A legitimate STA is expected to transition through state 0 through to state 9 in a strict sequence (see Figure 1) to establish a SA with an AP. Table 1 and Figure 1 show all the frames originating from the STA or the AP that cause transitions in the state transition model and their target states. The states are expected to be traversed incrementally and last state of the STA cannot be the same as its current state with the exception of state 9. This is the state when the SA is complete and the STA and the AP engage in data communication.

4.2. Detecting State Transition Violations

All STA-AP associations should strictly transition through the sequence of states specified by the state transition model (see Figure 1). Any anomalous transitions in the observed state transition model can be used to detect violations of the model. There can be three kinds of anomalous transition: (a) a negative state shift, which occurs when the STA transitions from a higher state to a lower state; (b) a positive state shift, which occurs when the STA bypasses an incremental state; and (c) a zero state shift, which occurs when

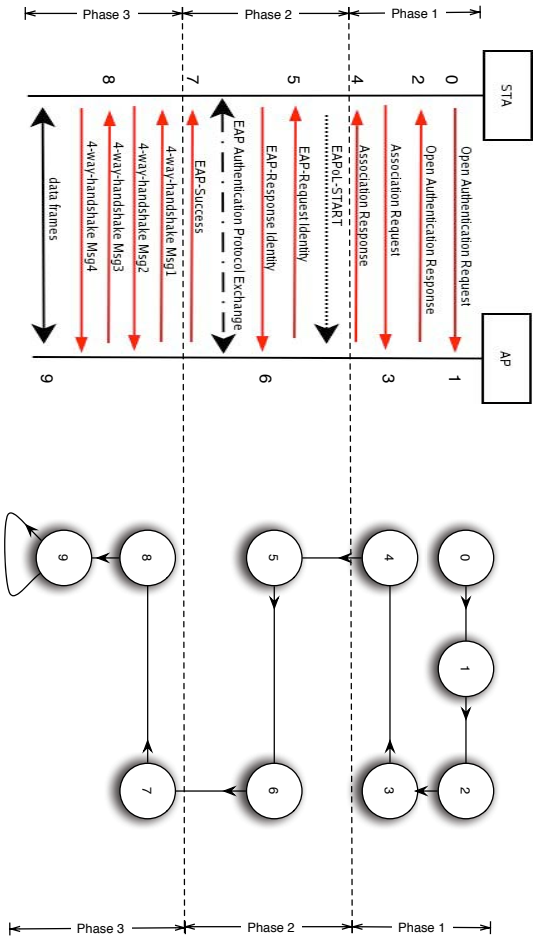

Figure 1. State Transition Model

Frame Type	Classification	Target Transition State	State Number
Authentication Request	+ve	*WAIT_AP_OPEN_AUTH_SUCCESS*	1
Authentication Response (success)	+ve	*AUTHENTICATED_UNASSOCIATED*	2
Association Request	+ve	*WAIT_AP_ASSOC_RESPONSE*	3
Association Response (success)	+ve	*AUTHENTICATED_ASSOCIATED*	4
ReAssociation Request	+ve	*WAIT_AP_ASSOC_RESPONSE*	3
ReAssociation Response (success)	+ve	*AUTHENTICATED_ASSOCIATED*	4
EAP-Request Identity	+ve	*802.1X_INIT*	5
EAP-Response Identity	+ve	*WAIT_EAP_AUTH*	6
EAP-Success	+ve	*4_WAY_HANDSHAKE_INIT*	7
EAPoL-Key (message 3)	+ve	*KEYDONE_PORTCLOSED*	8
Data	+ve	*DATA_TX_RX*	9
Deauthentication	-ve	*UNAUTHENTICATED_UNASSOCIATED*	0
Disassociation	-ve	*AUTHENTICATED_UNASSOCIATED*	1
EAPoL-Start	-ve/+ve	*802.1X_INIT*	5
EAPoL-Logoff	-ve	*802.1X_INIT*	5
EAP-Failure	-ve	*802.1X_INIT*	5

Table 1. Frame Type Transitions

the STA does not change its current state from its last state.

4.2.1 Negative Shifts

Negative state shifts occur when the STA, rather than moving sequentially through states 0 to 9, transitions to a state smaller than its last state. A negative shift is usually a symptom of a DoS attack where management or EAP frames are used to cause a negative state transition. Although all of these frames can be spoofed to launch a denial-of-service attack, they do serve a legitimate resource management function in WLANs. Hence a negative shift does not necessarily imply a DoS attack. To accommodate legitimate negative state shifts, the sensor uses an index of suspicion for every STA. This index is incremented for every observed negative shift for that particular STA and when this index exceeds a threshold value, the sensor raises an alert to that effect. Hence the index of suspicion is used to reduce the number of false positives and to flag the occurrence of excessive negative shifts as a likely DoS condition.

4.2.2 Positive Shifts

Positive state shifts occur when the STA, rather than moving sequentially through states 0 to 9, transitions to a state greater than its last state by a value of more than 1. Usually a positive shift is a consequence of frame loss, but in the presence of a preceding negative shift, a positive shift in the STA's state can be an indication of a spoofing, session hijacking or man in the middle (MITM) attack. Session hijacking attacks usually consist of two steps: the adversary forces the legitimate STA to disconnect from the network, usually via a DoS attack; and then assumes the MAC address of the victim STA to communicate with the network. This attack will cause two shifts in the STA's observed state transition model: a negative shift when the DoS attack is launched; and a positive shift when the adversary sends data frames spoofing the legitimate STA's MAC address.

Frame loss is very common in IEEE 802.11 networks. This might cause a positive shift in the observed state transition model. For example, if the sensor does not receive all transmissions, it may perceive a STA's state to be different from the STA's real state. To accommodate the effects of frame loss, the sensor uses an index of suspicion for keeping track of all positive state shifts observed for a particular STA. If the number of positive shifts exceeds the predefined index of suspicion threshold for a particular STA, the sensor alerts this condition as excessive frame loss.

4.2.3 Zero Shifts

Zero shifts occur when the STA's current state and last state are the same. In the state transition model (see Figure 1), the STA is expected to enter only one state repeatedly, namely state 9, after it has successfully established security association. If a STA repeatedly remains in a state other than 9, this could be indicative of a misconfiguration or a DoS flooding attack. However, a zero shift condition could also occur if the monitor receives retransmissions of traffic to/from the STA. To minimize the rate of false positives generated from this condition, the sensor maintains an index of suspicion where an alert is raised only when the number of zero state shift events for a particular STA exceeds the predetermined threshold for zero state shifts.

4.3. Unexpected Frames

Table 1 and Figure 1 show all the frames that result in state transitions (in the state transition model) and their target states. These frames can be classified into two categories: frames that lead to a +ve (positive) transition and frames that cause a -ve (negative) state transition. Each frame has a target transition state that it would cause the sensor's observed state transition model to transition to. However, depending on the classification (+ve/-ve) of a frame and the current state of the STA in the state transition model, the frame might not lead to any transition in the state transition model. We refer to such frames as *unexpected frames*. Real world STAs and APs respond only to certain frames in certain states. The *unexpected frames* represent all frames that would be ignored by the STA/AP in its current state.

The logic to determine if a frame is an *unexpected frame* for the current state is now explained. For a +ve frame, a transition is only processed if the current state of the STA is smaller than or equal to the target transition state for that frame. Similarly a -ve frame leads to a state transition only if the current state of the STA in the state transition model is greater than or equal to the target transition state for that frame. If an *unexpected frame* is detected, an alarm is raised.

4.4. Attack Classification

When attacks are detected they are classified using the type-subtype of the last frame that caused the alarm and examining the address fields of the frame. For instance, if a *Deauthentication Broadcast Flood* is launched, a zero state transition tolerance threshold exceeded alarm is raised. This alarm indicates a flooding attack. The type-subtype of frames being injected

would be *deauthentication* and the destination address field would be broadcast. This information is used to classify the attack as a *Deauthentication Broadcast Flood* attack. Also in order to classify *unexpected frame* flooding attacks, a counter is used which is incremented for every *unexpected frame* received. When this counter exceeds a pre-configured threshold, an alarm is raised to indicate that an *unexpected frame flooding* attack was detected.

4.5. Security Policy Constraints

While the state transition model serves as a good starting point for a detection system it must be further refined. In order to detect intrusive actions that do not violate the state transition model but do violate the security policy, additional constraints must be incorporated into the specification. These constraints are derived from the network security policy requirements. In the case of our example, these constraints are on the capabilities that the network should advertise (via the robust security network information element (RSN IE)), the supported encryption algorithms and the required authentication method. In summary these constraints are:

- **RSN mode of operation**: To monitor for compliance with this constraint, the wireless sensor will inspect the contents of the advertised AP capabilities contained in the RSN IE to verify that required parameters are present and that prohibited parameters are absent. The list of permitted and prohibited modes of operation are provided to the sensor via runtime configuration directives.
- **802.1X**: The use of port-based network access control can be evaluated from the state transition model. A network not supporting 802.1X, will not enter states 5 and 6 of the state transition model (as shown in Figure 1), but will move directly from state 4 to state 9, which will result in an alert being raised if port-based access control is required by the policy.
- **AES link layer encryption**: As with the requirement for RSN, the monitor evaluates the RSN IE and checks that AES is the only supported pairwise cipher suite advertised for use. Again, the list of permitted and prohibited encryption algorithms are provided to the sensor via runtime configuration directives.
- **EAP-TLS for authentication**: For stations in state 6, the monitor checks that any EAP Request frames destined for the monitored STA contain an EAP-Type of EAP-TLS. A more sophisticated monitoring capability could be implemented by adding EAP method specific states to the state transition model, to ensure that the EAP method was executing as expected. Such an approach would be of limited use when tunnelled EAP methods[2] are employed.

For each STA, the sensor checks the security constraints whenever it transitions between states. Any violation of these security constraints is raised as an alarm as a violation of the site security policy.

5. Experimentation

A number of experiments were carried out with the following goals: 1) to establish the feasibility and reliability of the proposed specification based intrusion detection technique; 2) to perform a comparative analysis to demonstrate the effectiveness of the proposed technique against other wireless intrusion detection techniques; 3) to establish the impact of the sensor location on reliability and effectiveness of the proposed technique; and 4) to demonstrate the policy compliance monitoring capabilities of the technique.

5.1. Methodology and Setup

The experimental setup involved six nodes: an STA, an AP, an attacker and three sensors. For the purposes of the experiments, a Robust Secure Network (RSN) was set up using *hostapd* software running on a Linux laptop as the AP. A Windows XP PC was used as the STA. Three Linux laptops were used as passive sensors and a laptop running Linux was used as the attacking station. An open source wireless IDS, *Snort-Wireless*[3] was used for comparative analysis with its default settings. A custom *Snort-Wireless* preprocessor (*State-Transition-Processor*) was used on the sensors to process all WLAN events passively observed by the sensor and use the intrusion detection techniques presented in Section 4 to detect violations of the site security policy. A custom tool *zaildar* that supports injection of arbitrary frames into a WLAN was developed for these experiments.

To satisfy the goals of the experiments, as identified earlier, the experiments were divided into three sets: Set1, Set2 and Set3. In Set1 experiments a number of DoS attacks were launched against established

[2]Tunnelled EAP methods are executed over encrypted links and are not passively observable.

[3]http://snort-wireless.org

associations by spoofing the MAC addresses of the legitimate AP or STA. A data exchange was established between the legitimate STA and AP, after establishment of a SA, so that effects of the DoS attacks were obvious. In Set2, MAC spoofing based attacks were launched against the WLAN infrastructure by random STAs where no SA existed between the random STAs and the AP. All attacks involving masquerading the STA source MAC that were launched in Set1 were also launched in the Set2 experiments. Set3 experiments were carried out to establish the false positive rate of the proposed technique in the absence of any attack traffic and to demonstrate the ability of the proposed technique to provide policy compliance monitoring. In these experiments, after establishment of SA, a data exchange was started between the STA and the AP and the sensors were used to monitor this normal WLAN activity. Then for policy compliance testing, the AP was reconfigured to implement an *Open Network* instead of a RSN and the sensors were used to monitor another data exchange between the STA and the AP. The *Open Network* does not require any authentication and merely requires the STA to be in state 4 to be able to perform data exchange with the AP.

In order to assess the proposed technique's ability to deal with varying levels of frame loss, three sensors were located at increasingly greater distance from the AP and the STA. It was expected that as the distance between the sensor and the monitored entities increases, the rate of false positives would increase for that sensor due to frame loss. To test this theory, the AP, the STA, the attacker and a sensor (Sensor1) were placed in one lab room, in close proximity to each other. The second sensor (Sensor2) was placed in another lab room about 20 meters away from the position of the AP and the third sensor (Sensor3) was placed further away about 50 meters away from it in yet another room. The sensors were used to obtain traffic captures, during the running of the attacks, from their respective locations. The traffic capture (on all three sensors) was terminated after every attack and saved in a separate file. This ensured that each traffic capture file contained only one attack, effectively labeling the attack captures. Traffic captures from each sensor were then processed by the *State-Transition-Processor* on the sensor in offline mode. The experiments were carried out in a busy RF environment where numerous actively used WLANS were known to operate. Hence, the sensors captured noise as well as relevant attack traffic.

In the interest of comparative analysis, *Snort-Wireless* was used with its default settings with the *macspoof*, the *authflood* and the *deauthflood* preprocessors enabled. In the remainder of the paper this setting of *Snort-Wireless* is referred to as *Snort-Wireless-Default*. A comparative study was performed on detection capabilities of *Snort-Wireless-Default* and our custom *Snort-Wireless* preprocessor i.e. *State-Transition-Processor*.

5.2. Choosing Thresholds

Due to legitimate use of various management frames in WLANs and the existence of unexpected state transitions and frames (for benign reasons such as frame loss and restransmissions) thresholds were implemented in order to effectively manage the rate of false negatives and false positives for the *State-Transition-Processor*. In this paper, we refer to these as *indices of suspicion* and their values were determined by a combination of empirical analysis of WLAN traffic and what was considered to be most reasonable for that particular threshold. A threshold value of 5 was used for each of the negative state shift, the zero state shift and the unexpected frame tolerance thresholds. While positive state shift tolerance threshold was set to 2. Post-hoc analysis of the experiment captures also confirmed the effectiveness of these thresholds in keeping false negatives and false positives to a minimum.

5.3. Results and Observations

This section presents the results and observations of executing *State-Transition-Processor* and *Snort-Wireless-Default* on captures obtained from Set1, Set2 and Set3 experiments. The *macspoof* preprocessor of *Snort-Wireless-Default* uses the flawed technique of using the patterns of sequence number changes to detect intrusions. As expected, it generated numerous false alerts for every MAC address in every capture file, irrespective of the attack. This indicated that this preprocessor was too noisy to provide any real results for our experiments and was ignored in the evaluation of the results for all our experiments.

Tables 2 and 3 show the results of executing *State-Transition-Processor* and *Snort-Wireless-Default* over captures obtained from the three sensors for each attack launched in Set1. The first column describes the attack type and the target state transition for the STA. The *Alarms* column describes the type of alarm generated and the state transition sequence that triggered it. The following keys are used to describe the type of attacks: *Zero* = zero state transition tolerance threshold exceeded, *Pos* = positive state transition tolerance threshold exceeded, *Neg* = negative state transition tolerance threshold exceeded, *noSA* = Data

frame detected from a MAC without SA, *nonCompliantNetwork*= RSN IE does not match the required IE, *Spoofing* = Positive state shift detected immediately proceeding a negative state shift or a -ve frame (see Section 4.2), *unexpected frame* = unexpected frame received for current state, *unexpected frame flood* = unexpected frame tolerance threshold exceeded. *noSA* is a specialized case of an *unexpected frame*. In the remainder of the paper, these alarm keys are used to refer to the alarms. The middle column represents success or failure in detecting the attack for the capture obtained from that sensor. The $\surd$ symbol is used to represent successful detection, the $\times$ symbol is used to represent detection failure. All the attacks in Set1 were correctly classified by the *State-Transition-Processor*. However, *Snort-Wireless-Default* only detected and classified the deauthentication flooding attacks and the *Authentication Flood* attack. It has no capability of detecting the other attacks in Table 2.

Attack	Sensor1	Sensor2	Sensor3	Alarms
Deauthentication Broadcast Flood DoS (9→0)	√	√	√	Zero (0→0), Neg (0,1,2...→0) Spoofing , noSA (0→9)
Targeted Deauthentication DoS (9→0)	√	√	√	Zero (0→0)
Client Initiated Targeted Deauthentication DoS (9→0)	√	√	√	Zero (0→0), Neg (0,1,2...→0) Spoofing , noSA (0→9)
Disassociation Broadcast Flood DoS Attack (9→2)	√	√	√	Zero (2→2), Neg (2,3,4...→2)
Targeted Disassociation DoS (9→2)	√	√	√	Zero (2→2), Neg (2,3,4...→2)
Client Initiated Targeted Disassociation DoS (9→2)	√	√	√	Zero (2→2), Spoofing , noSA (2→9)
Association Flood	√	√	√	unexpected frame flood, unexpected frame
Authentication Flood	√	√	√	unexpected frame flood, unexpected frame
EAP-Success Flood	√	√	√	unexpected frame flood, unexpected frame
EAPoL-Start Flood DoS (9→5)	√	√	√	Zero (5→5), Spoofing , noSA (5→9)
EAP-Failure Flood DoS (9→5)	√	√	√	Zero (0→0), Neg (0,1,2...→0) unexpected frame flood, unexpected frame
EAPoL-Logoff Flood DoS (9→5)	√	√	√	Zero (5→5), Neg (5,6,7...→5) Spoofing , noSA (5→9)
EAP-Request Identity Flood DoS	√	√	√	unexpected frame flood, unexpected frame

Table 2. *State-Transition-Processor* **attack detection and alarms for Set1 experiments**

For all the attacks launched in Set2, the *State-Transition-Processor* raised type *Zero* alarms on processing captures obtained from all the sensors. An exception was the *Authentication Flood* attack, which generated *unexpected frame* and *unexpected frame flood* alarms. Other exceptions were *Association Flood* and *EAP-Failure Flood* attacks which resulted in *unexpected frame*, *unexpected frame flood* and *Zero* alarms. All attacks were classified correctly by the *State-Transition-Processor* for all captures. On the other hand, *Snort-Wireless-Default* had the same detection rate (on all three sensors) as in Set1 experiments.

For the first part of Set3 experiments, where a normal RSN data transfer was monitored, the *State-Transition-Processor* did not raise any alarms for captures obtained from Sensor1 and Sensor2. However, the capture from Sensor3 led to generation of false alarms of type *Pos* and *noSA*. For *Open Network* traffic monitoring, the *State-Transition-Processor* raised alarms of type *nonCompliantNetwork* and *noSA* for captures obtained from all sensors. The capture from Sensor3 also led to generation of false alarms of type *Pos*. In both cases, *Snort-Wireless-Default* raised no alarms apart from the *macspoof* alarms which were ignored as noise.

Attack	Sensor1	Sensor2	Sensor3	Alarms
Deauthentication Broadcast Flood DoS	√	√	×	deauthFlood
Targeted Deauthentication DoS	√	√	√	deauthFlood
Client Initiated Targeted Deauthentication DoS	√	√	√	deauthFlood
Authentication Flood	√	√	√	authFlood

Table 3. *Snort-Wireless-Default* **attack detection and alarms for Set1 experiments**

5.4. Analysis

The *State-Transition-Processor* was able to detect all the attacks launched in experimentation Set1 and Set2 correctly by monitoring the state changes in the state transition model described in Section 4.1. It also successfully detected policy violations in Set3 by applying site security constraints and demonstrated a low rate of false positives.

5.4.1 Attack Detection

The *State-Transition-Processor* was not only successful in detecting all attacks in Set1 and Set2, it also classified the detected attacks correctly in all cases by simply using the type-subtype of the last frame causing an increment in one of the indices of suspicion and examining the address fields of the frame. In Set1 and Set2 experiments, the alarm type of *Zero* was correctly raised in all flooding attacks and in instances where there were numerous unsuccessful attempts by the STA to establish SA with the AP, the alarm type of *Neg* was also correctly raised indicating a DoS attack. Whenever a data frame was detected to/from a MAC address that had not completed its SA, a *noSA* alarm was raised. MAC address spoofing attacks (alarm type *Spoofing*) were also correctly detected whenever a positive state shift was detected immediately following a negative state shift or a -ve frame. In these experiments an alarm type of *Spoofing* was raised whenever the sensor detected a data frame to/from the entity that had just received a -ve frame or had undergone a negative shift. As described in Section 4.2, the frames that

would not normally be processed by a real STA/AP in a particular state were flagged by raising the *unexpected frame* alarm type. The *unexpected frame flood* alarm type was raised whenever a threshold number of the unexpected frames was detected.

There were some unexpected frame transitions that occurred in the Set1 and Set2 experiments. For instance, in Set1 experiments, the *EAP-Failure* attack led to the generation of additional alarms of type *unexpected frame* and *unexpected frame flood*. This was caused because on receiving an *EAP-Failure* frame, the STA sends a deauthentication frame to the AP, hence transitioning the observed state to 0. Hence every subsequent *EAP-Failure* frame led to the generation of an *unexpected frame* alarm as the STA was in state 0. A complete list can be seen in Table 2. However none of these unexpected transitions affected the successful classification/identification of the real attack.

In all experiments the use of indices of suspicion in the *State-Transition-Processor* assisted in eliminating the majority of sources of false positives and multiple alerts for the same attack, while maintaining a null false negative rate. In Set3 experiments, the captures obtained from Sensor3 led to the generation of a number of false positives. As expected, the most common false positives generated were alarms of type *Pos* and *noSA* as these alarms are directly related to frame loss. The rate of false positives appears to be directly proportional to the rate of frame loss and Sensor3, being the furthest away from the AP and the STA, experienced the most frame losses. No false positives were generated by Sensor1 or Sensor2. By correlating the alarms generated by different sensors, most of the false positives can easily be eliminated. All false positives generated in Set3 experiments were eliminated when results from the three sensors were correlated. In the correlation process, perhaps more weight can be placed on the alarms generated by the sensors located in close proximity of the AP and the STA[4].

In Set3 experiments, the *State-Transition-Processor* correctly detected violation of the site security policy by simply using the state transition model and related constraints (provided via a config file). The IE of the new WLAN did not match the expected IE and the state transition model did not expect data transfer to occur before state 8 is reached. In *Open Networks* data transfer can occur after state 4. This led to generation of the *nonCompliantNetwork* and *noSA* alarms.

[4]This scheme will have to be assisted by some flavor of location aware infrastructure

5.4.2 Comparative Analysis

In the experiments, *Snort-Wireless-Default* was only able to detect the authentication flood attacks and the various deauthentication flood attacks and was unsuccessful in detecting attacks violating the site security policy; whereas *State-Transition-Processor* successfully detected all the launched attacks (see Table 3 and Section 5.3). A noticeable drawback in *Snort-Wireless-Default* is that it does not provide any DoS attack detection apart from flooding attacks, which is further limited to just authentication and deauthentication flood attacks. Also its *deauthflood* and *authflood* preprocessors require the injected frames to arrive at a certain rate (which is configurable) for successful detection of flooding attacks. For instance *Snort-Wireless-Default* did not detect the deauthentication broadcast flood attack in the traffic capture obtained from Sensor3 since, due to frame loss, Sensor3's capture file did not represent the attack incoming at a high rate (see Table 3). *State-Transition-Processor* has no such preconditions and is capable of detecting flooding attacks without any dependence on the rate of frame injection.

Besides flooding attacks, *State-Transition-Processor* is also capable of detecting spoofing/MITM/session hijacking attacks by checking for a positive state shift immediately following a negative state shift or a -ve frame. This detection mechanism is much more reliable and robust than the one used by *Snort-Wireless-Default* i.e. detecting gaps in sequence numbers of frames (*macspoof* preprocessor). *State-Transition-Processor* can even detect stealthy DoS attacks such as when an attacker monitors a legitimate STA passively and injects a DoS attack frame into the WLAN every time the STA reaches half way through establishing SA with the AP. However, *Snort-Wireless-Default* has no means of detecting similar attacks. *Snort-Wireless-Default* is also not capable of detecting random injection attacks aimed at consuming network bandwidth such as the injection of data frames with random STA source addresses and injection of random frames which will have no impact on the state of the STA (i.e. *unexpected frames*). The *State-Transition-Processor* would detect this as anomalous activity. *Snort-Wireless-Default* requires a separate preprocessor to detect each kind of attack, while *State-Transition-Processor* can detect attacks without requiring any additional algorithms or preprocessors [5]. *State-Transition-Processor* does not require constant code updates/patches to detect new attacks and has a very small footprint and hence can be easily deployed on a large scale without the need of

[5]The authors however acknowledge the flexibility provided by the design of *Snort-Wireless* which enables pre-processors to be readily plugged in for enhanced functionality

extensive computing power.

6. Conclusions And Future Work

In this work we have presented an approach for constructing an IDS for infrastructure WLANs using a specification-based approach. This system implements both attack detection and policy compliance monitoring, a unique contribution of this work. The specification used by the system comprised a state transition model and set of constraints. Construction of the state transition model was based on the underlying network protocol specifications (IEEE 802.11, 802.1X, and EAP). Constraints were derived from an example security policy.

The IDS constructed using these techniques was implemented as a *Snort-Wireless* preprocessor and its accuracy and sensitivity were evaluated. The technique proved capable of passively detecting all known attacks and effective at monitoring for policy compliance. Experimental analysis demonstrated that the technique, when combined with the use of *indices of suspicion* was able to detect all intrusions with a minimal number of false positives (i.e. the technique is accurate) and no attacks went undetected (i.e. the technique is sufficiently sensitive). The system meets the requirement of being maintainable. As an implementation of an anomaly based approach, the system does not require updates as new attacks, or variants of attacks emerge. The use of runtime configuration parameters permits the constraints used by the system to be easily updated as the network security policy changes.

In future work, we aim to evaluate the attack resistance of the approach to confirm that it is able to reliably report intrusions in the presence of attackers that are attempting to exhaust the resources of the monitor, by instantiating a large amount of state on the monitor for example. We also aim to study more complicated and complex algorithms to determine the values for *indices of suspicion*. We would also like to study the effects of tuning these thresholds on the false positives and false negatives.

References

[1] J. Bellardo and S. Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *Proceedings of the USENIX Security Symposium. Washington D.C., USA*, 2003.

[2] C.-Y. Chin-Yang Tseng, B. P., C. Ko, R. Limprasittiporn, J. Rowe, and K. N. Levitt. A specification-based intrusion detection system for AODV. In S. Setia and V. Swarup, editors, *SASN*, pages 125–134. ACM, 2003.

[3] M. Chirumamilla and B. Ramamurthy. Agent based intrusion detection and response system for wireless LANs. In *IEEE ICC '03. Volume: 1 , 11-15 May*, pages 492–496, 2003.

[4] H. Debar and J. Viinikka. Intrusion detection: Introduction to intrusion detection and security information management. In *FOSAD 2004/2005*, 2005.

[5] R. Gill, J. Smith, and A. Clark. Experiences in Passively Detecting Session Hijacking Attacks in IEEE 802.11 Networks. In R. Safavi-Naini, C. Steketee, and W. Susilo, editors, *Fourth Australasian Information Security Workshop (Network Security) (AISW 2006)*, volume 54 of *CRPIT*, pages 221–230, Hobart, Australia, 2006. ACS.

[6] F. Guo and T. Chiueh. Sequence number-based MAC address spoof detection. In A. Valdes and D. Zamboni, editors, *RAID*, volume 3858 of *LNCS*, pages 309 – 329. Springer, 2005.

[7] C. He and J. C. Mitchell. Security analysis and improvements for IEEE 802.11i. In *Proceedings of the 12th Annual Network and Distributed System Security Symposium*, Feb 2005.

[8] IEEE Std 802.11i–2004. *WLAN Security Standard*. IEEE Standards Association, 23rd July 2004.

[9] K. Ilgun, R. A. Kemmerer, and P. A. Porras. State transition analysis: A rule-based intrusion detection approach. *IEEE Trans. Software Eng.*, 21(3):181–199, 1995.

[10] C. C. W. Ko. *Execution Monitoring of Security-Critical Programs in a Distributed System: A Specification-based Approach*. PhD thesis, U.C. Davis, 1996.

[11] Y.-X. Lim, T. Yer, J. Levine, and H. Owen. Wireless intrusion detection and response. In *Information Assurance Workshop. IEEE Systems, Man and Cybernetics Society, 18-20 June*, pages 68–75, 2003.

[12] A. Mishra and W. Arbaugh. An Initial Security Analysis of the IEEE 802.1X Standard. Technical report, 2003.

[13] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and Z. S. Specification-based anomaly detection: a new approach for detecting network intrusions. In V. Atluri, editor, *ACM CCS*, pages 265 – 274. ACM, 2002.

[14] D. Stanley, J. Walker, and B. Aboba. Extensible authentication protocol (EAP) method requirements for wireless LANs. Technical report, IETF, 2005. RFC 4017.

[15] J. Wright. Detecting wireless LAN MAC address spoofing, 2003. White paper.