# Substantiating Anomalies In Wireless Networks Using Group Outlier Scores

Elankayer Sithirasenan, Vallipuram Muthukkumarasamy
School of Information and Communication Technology, Griffith University, Gold Coast, Australia
Email: {e.sithirasenan, v.muthu}@griffith.edu.au

*Abstract*— **Huge amounts of network traces can be collected from today's busy computer networks. Analyzing these traces could pave the way to detect unusual conditions and/or other anomalies. Presently, due to the lack of effective substantiating mechanisms intrusion detection systems often exhibit numerous false positives or negatives. The efficiency of a network intrusion detection system (NIDS) depends very much on detecting and effectively validating the detected anomalies. Furthermore, most NIDSs do not have proven mechanisms that will easily accommodate legitimate dynamic changes. Achieving dynamic adaptation in real time has been a long standing desire for effective intrusion detection and prevention. Real time detection of outliers is a feasible option to substantiate anomalies in large data sets, leading to effective intrusion detection and prevention. In this context we propose and investigate a novel mechanism to detect intruders and to classify security threats using group outliers. Our system monitors for timing and/or behavioral anomalies and uses outlier based techniques to substantiate the anomaly. In this paper we introduce the concept of Group Outlier Score (GOS) and its use in substantiating security threats in wireless networks. We have tested the concept on our experimental wireless networking environment. The analysis of the results reveals that with a threshold value of 1.2 for GOS our system demonstrates optimum performance.**

*Index Terms*— **Security, Outlier Detection, Intrusion Detection, Wireless Networks.**

## I. Introduction

Due to continuing advances in communication technology, the use of computer networks has progressed exponentially. More and more hosts are connecting to wired or wireless computer networks resulting in large amounts of data being collected for network security related analysis. To get the most out of this data, effective analysis methods are required to extract non-trivial, valid, and useful information. Considerable research work has been carried out towards improving knowledge discovery in databases (KDD) in order to meet these demands.

In several applications, such as network intrusion detection, sensor networks, stock market analysis, health monitoring systems, etc., the problem of detecting rare events, abnormal behavior, and exceptions is very important. Methods for finding such outliers in large data sets are drawing increasing attention of researchers. The leading approaches to outlier detection can be classified as distance based [1], [2], depth-based [3], clustering [4], density-based [5] or discovery-driven [6].

In the context of wireless networks, the behavior of wireless hosts may often change due to the nature of wireless environment, mobility of the user and the variations in user requirements. As shown in Figure 1 anomaly filters detect anomalies that are often outside the theoretical or practical behavior region of a protocol. Anomalies outside the theoretical and/or practical behavior regions are common and are usually easy to detect and substantiate [7]. On the other hand, anomalies within the theoretical or practical behavior regions are rare and lead to false negative reporting. Detecting these anomalies is challenging and requires analyzing the protocol behavior from different perspectives. Our proposed system substatiates such anomalies by analyzing it from different view points. A wireless host which behaves normally from one view point may behave differently from another view point. Hence, the challenge is to differentiate the legitimate changes of the wireless host from potential security threats.

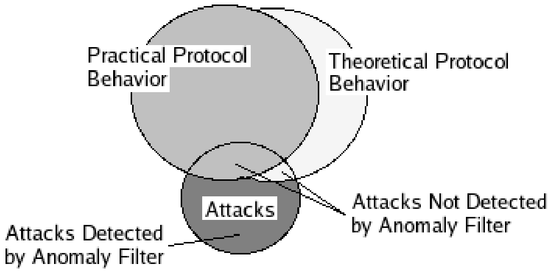
Practical Protocol Behavior
Theoretical Protocol Behavior
Attacks
Attacks Not Detected by Anomaly Filter
Attacks Detected by Anomaly Filter

Figure 1. Attacks Not Detected By Anomaly Filter

In this paper we discuss the possible use of assorted data views for substantiating the legitimacy of security threats in wireless computer networks. Our proposed method which tries to differentiate between legitimate and illegitimate events in the network environment is developed using outlier based data association techniques. The main features of our approach are:

- We build a repository of network traces on a Beowulf cluster as a partial data cube compared to methods which consider building and manipulating the entire data cube.
- We update aggregates in real time as opposed to methods which calculate summary values during

This work was supported in part by Institute for Intelligent and Integrated Systems (IIIS), Griffith University, Australia.
Manuscript received August 28, 2010; revised December 7, 2010.

fixed time intervals.

- We substantiate the legitimacy of abnormal conditions using GOS and use different characteristics of the network to accurately classify security threats.
- We rebuild only selected views of the data cube to accommodate changes to the network environment rather than rebuilding the full cube.

Our experiments demonstrates online real time manipulation of multidimensional data sets represented as a partial data cube suitable for applications that requires real time response.

Using partial data cubes for fast querying and populating the aggregate values in real time makes our detection system viable and reliable for a range of critical applications. Our outlier detection algorithm makes at most three drill-down queries on the partial data cube to substantiate the legitimacy of the security threats. Thus, our method demonstrates online real time manipulation of multidimensional data sets represented as a partial data cube suitable for applications that requires real time response. To the best of our knowledge, this is the first work to successfully use group outliers for substantiating abnormal conditions in computer networks.

This paper is organized as follows. In Section II we give a brief overview of related work on using data mining for intrusion detection and outlier detection. The concept of partial data cube construction with some basic observations and properties are presented in Section III. In Section IV we define uncertainty and develop mathematical expressions for substantiating anomalies. The experimental results are presented in Section V and analyzed in Section VI. Finally, Section VII concludes the paper.

## II. Related Work

The recent development in data mining has made available a wide variety of algorithms, drawn from the fields of statistics, pattern recognition, machine learning and database for the purpose of detecting irregularities in data sets. Storing of large amounts of network traces, and analyzing them in real time has distinguished data mining as a promising mechanism to incorporate for intrusion detection [8]–[10].

Most data mining based intrusion detection systems detect unusual events or anomalous behavior in networks effectively. However, the major issue is the validation of the detected illegitimate events. Almost all of the work found in the literature present techniques to detect abnormal or rare events in networks. However, having found an anomalous or rare event how do we classify it as legitimate or illegitimate is the major challenge yet to be addressed. In this view, discovery of outliers to extract a few data objects with abnormal behavior patterns, which are more interesting than common patterns in some cases, can be of practical significance in intrusion detection systems, credit fraud detection, etc. Hence, we are using outlier mining techniques to address the problem of validating the legitimacy of abnormal conditions.

Much of the work in outlier detection has been approached from a statistical point of view and is primarily concerned with one or very few attributes. However, because the network data has many dimensions, the use of clustering for anomaly detection has been investigated. Clustering is an unsupervised machine learning technique for finding patterns in unlabeled data with many dimensions. $K$-means clustering is used to find natural groupings of similar alarm records. Records that are far from any of these clusters indicate unusual activity that may be part of a new attack [11].

Several methods for detecting outliers in multivariate data without apriori assumption of the distribution have been proposed. Knorr and Ng [1] proposed the distance-based approach where an object in a data set $P$ is a distance-based outlier if at least a fraction $b$ of the objects in $P$ is further than $r$ from it. This outlier definition is based on a single, global criterion determined by the parameters $r$ and $b$. However, this can lead to problems when the data set has both dense and sparse regions.

The depth-based approach computes different layers of $k$-$d$ convex hulls [3]. Objects in the outer layer are detected as outliers. However, it is a well-known fact that the algorithms employed suffer from the dimensionality curse and cannot cope with large $k$. In the case of clustering, many algorithms detect outliers as by-products [4]. Their main objective is clustering; hence they are not optimized for outlier detection. Furthermore, in most cases, the outlier definition or detection criteria are implicit and cannot easily be inferred from the clustering procedures.

The density-based approach proposed by Breunig, et al. [5] relies on the Local Outlier Factor (LOF) of each object, which depends on the local density of its neighborhood. The neighborhood is defined by the distance to the MinPts-th nearest neighbor. In typical use, objects with a high LOF are flagged as outliers. Jin, et al. [12] proposed an algorithm to efficiently discover top-n outliers using clusters, for a particular value of MinPts. LOF does not suffer from the local density problem. However, selecting MinPts is non-trivial. In order to detect outlying clusters, MinPts has to be as large as the size of these clusters, and computation cost is directly related to MinPts. Furthermore, the method exhibits some unexpected sensitivity on the choice of MinPts. Aggarwal et al. [13] claim that both distance-based and local outliers do not work well for high dimensional dataset since the data are sparse, and outliers should be defined in sub-space projections. They proposed an evolutionary algorithm to find the outliers.

All of the above outlier detection methods are based on individual outliers, and the association of outliers has not been considered. In contrast, the “discovery-driven” method proposed by Sarawagi, et al. [6] aims at finding exceptions in data cube cells. In this method of data exploration an analyst’s search for anomalies is guided by pre-computed indicators of exceptions at various levels of detail in a data cube. They define a cell as an exception if the aggregate of the cell differs significantly from its anticipated value. The anticipated value is calculated by a formula and they suggest an additive or multiplicative form. They also give a formula to estimate the standard deviation. When the difference between the cell value and its anticipated value is greater than 2.5 standard deviation, the cell is considered an exception.

Similar to this work, Lin and Brown [14], [15] also focus on OLAP cube cells in their analysis. They define a function on OLAP cube cells to measure the extremeness of the OLAP cell, which is called the outlier score. Instead of defining outlier for individual record, they consider to build the outlier measure for a group of data points. These data points are "similar" on some attributes and are "different" on other attributes. If these common characteristics are quite "unusual", or in other words, they are "outliers", these data points are well separated from other points. Hence, they claim that this "weird" characteristics strongly suggest that these data points are generated by a particular mechanism, and could be associated. This method combines both outlier detection in data mining and concepts of OLAP. They also describe a real world example.

Although both the above methods focus on detecting exceptions, they have not considered using it for substantiating the exceptions for real time decision making. Further, for real time operations which requires very fast responses as in security or health related applications the use of OLAP cubes are yet to be investigated. Moreover, maintaining large data sets and updating them in real time may be intricate and impractical.

In this view our outlier detection technique includes several novel features. Firstly, our data repository is built as a partial data cube, because in our case we don't have to materialize all of the views. Further, we update the data cube on-the-fly enabling a continuous learning strategy. Finally, when it becomes necessary to rebuild the partial data cube we re-construct only those views that needs to be updated. Materializing selected views and using surrogate views for querying the data cube offer the query performance necessary for applications such as network intrusion prevention and health monitoring systems, which require real time response [16]. Moreover, many applications demand substantiating the legitimacy of events in real time. In this respect the use of group outliers enables us to substantiate the legitimacy of events from different view points. This feature will be vital in network intrusion detection systems and the like to reduce the number of false alarms. In the next section we briefly discuss wireless networks and the association process.

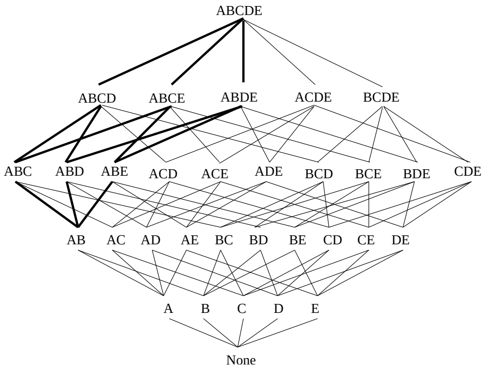

Figure 2. Data Cube Views

TABLE I.
DATA CUBE ATTRIBUTES

| Reference | Attribute | Type |
|---|---|---|
| A | Source ID | Dimension |
| B | Destination ID | Dimension |
| C | Event ID | Dimension |
| D | Time of Day | Dimension |
| E | Channel ID | Dimension |
| F | Protocol ID | Dimension |
| G | Cipher ID | Dimension |
| - | Event count | Measure |
| - | GOS | Measure |

## III. THE DATA CUBE

A popular model for On-line Analytical Processing (OLAP) applications is the multidimensional database also known as the data cube [17]. A data cube consists of two kinds of attributes: dimensions and measures. The set of dimensions may consist of elements like IP addresses, port addresses, event identities etc. that together form a key. Measures are typically numeric elements like packet size, duration etc. Data cube queries represent an important class of OLAP queries in decision support systems. The pre-computation of the different views of a data cube (i.e., the forming of aggregates for every combination of GROUP BY attributes) is critical to improving the response time of the queries [18]. However, in many cases not all views are needed for decision making, therefore it is advantageous to use only selected views. Such cubes with only selected views are referred to as partial data cubes [19].

Figure 2 shows a data cube with thirty two views made up of five attributes. In an actual application although the number of attributes can be many, the required number of views may be very few depending on the requirements. For example, if we assume that attribute "A" and "B" represent the identities of APs and STAs respectively, then for data association analysis between the APs and STAs we will consider only those views that consists of attributes "A" and "B", i.e. views "ABCE", "ABCF", "ABEF", "ABC", "ABE", "ABF" and "AB". Furthermore, in some cases we may also need to establish associations with child views to further strengthen our validation (see views connected by bold lines in Figure 2). In this manner, for intrusion detection and classification purposes, we selected the appropriate attributes and views bearing in mind the various security threats on the wireless networking environment.

Table I presents the list of attributes that are stored in the data cube for our analysis. Since we are concerned with wireless traces during RSNA [20], we store only those attributes that are necessary for this purpose. In this respect the identities (Source and Destination IDs) of the two communicating hosts, the current message (event) passed, the time during the message is passed, the channel

TABLE II.
SELECTED VIEWS

| | | | | |
|---|---|---|---|---|
| ABCDFG | BCDFG | ABCFG | BDF | BDFG |
| ABDFG | AF | BF | BEF | ABDE |
| ADE | BCFG | AC | ABEF | ACE |
| AE | ACDEFG | ACD | ABEG | BDG |
| BCEFG | AB | BDEF | ACEFG | ABF |
| ABDEF | ABCE | ADG | ACDE | BDEG |
| ABC | BE | ABCDEFG | AEF | ABDEG |
| ABCEFG | BEG | BD | ADF | AD |
| AG | ABD | BCE | AEG | BC |
| ADEF | ABDG | ABDF | ABG | ACFG |
| BDE | BG | BCDEFG | ABE | ABCDE |
| ADEG | A | BCDE | BCD | B |
| ACDFG | ABCD | | | |

TABLE III.
VIEWS ASSOCIATED WITH SECURITY THREATS

| No. | Threat | Attributes |
|---|---|---|
| 1 | Replay attack | ABCD, ABC, ABD, BCD, ACD, ABCF, ABF, ACF, BCF |
| 2 | Masquerading and Malicious AP | BCFG, BCF, BCG, BFG, CFG<br>ACFG, ACF, ACG, AFG |
| 3 | Session Hijack | ABCE, ABC, ABE, BCE, ACE<br>ABCF, ABF, ACF, BCF |
| 4 | Man-In-The-Middle (MitM) | same as above |
| 5 | Denial of Service (DoS) | ABCD, ABC, AB |

on which the message is passed, the protocol used and the cipher used are stored in the data cube.

The attribute “Event ID” can take values 0 to 56, representing the 57 different messages exchanged during an RSNA [21]–[24]. The attribute Time of Day represents one of twenty four time periods during the day. It starts from value 0 (for time period midnight - 1.00 am) and continues up to value 23 (for time period 11.00 pm - midnight). Channel ID is from 1 to 11 for IEEE 802.11b/g networks and 36 to 161 (36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161) for IEEE 802.11a networks. Protocol ID ranges from 0 to 5, representing the six protocols; IEEE802.11, EAP, TLS, PEAP, LEAP and EAPOL considered in our study. The Cipher ID is 0 for AES/CCMP, 1 for TKIP and 2 for WEP. In addition to these attributes we can also store information related to the network identities and traffic related information if desired. The examples considered in our study are mainly focusing on wireless networks that adopt IEEE 802.11i security mechanism.

### A. Data Views Vs Security Threats

Having decided on the attributes, next, we established the views that are necessary to substantiate the security threats. Table II shows all of the views considered in our analysis. As mentioned earlier, almost all of the views selected include either or both attributes A and B. Using these views we can query the data cube on any of the associations related to access points and/or hosts. However, since we are interested in substantiating security threats it is important to establish a relationship between these views and security threats.

Table III shows the relationship between data cube views and some common security threats. This table was established considering each threat and identifying the attributes that are associated with the threat. Having identified the attributes we then categorized the views that are vital for substantiating the security threat. For example in the case of Threat 1 - Replay Attack, we need to track the source and destination of every message in addition to the event identity. A replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. This is carried out either by the originator or by an adversary who intercepts the data and retransmits it, possibly as part of a masquerade attack by IP packet substitution (such as stream cipher attack). Hence, attributes A, B and C are important to analyze this type of attack. However, attributes D (Time of Day) would be useful in the case of a delayed attack. Similarly, attribute F (Protocol ID) is also useful since the adversary may replay the packet on a different protocol as part of a masquerade attack. Therefore, to substantiate this threat we consider all views that are associated with these attributes.

In the case of Threat 2 - Masquerading and Malicious AP, an adversary uses a legitimate hosts’ identity to masquerade as a legitimate host. Here, we consider attributes B (Destination ID) and C (Event ID), since we need to track messages directed towards the Malicious host. For this type of threats attributes F (Protocol ID) and G (Cipher ID) are also significant since the masquerader may want to force a legitimate host to downgrade its capabilities in order to secure access. Hence, views BCFG, BCF and BCG will be important. The same attributes can be monitored with the source host as well. Therefore views ACFG, ACF and ACG will also be useful to substantiate this attack.

Threat 3 - Session Hijack is a more advanced attack, where the association between a legitimate station and the access point is hijacked by an illegitimate user. In this case the illegitimate user will force a channel change with the access point/station and masquerade as a legitimate access point/station. Hence, in this kind of a threat we need to track the source, destination, event, channel and protocol of the messages exchanged. By tracking the protocol we establish whether the illegitimate session establishes a different kind of association. Therefore, for this type of a threat, views associated with attributes A, B, C, E and F are are considered. As for the readers interest all views associated with these attributes are shown with bold lines in Figure 2.

Threat 4 - Man-In-The-Middle attack is not very much significant in the context of effective confidentiality measures. However, if a MitM attack turns into a session hijack attack then our detection mechanism can be of use. Hence, we can make use of the same attributes as for session hijack attack. Furthermore, if a MitM is actively participating in the communication between the two legitimate hosts, then a timing anomaly detector will be able to detect some form of timing anomaly. But, it will again be the intrusion prevention module that will have to substantiate the anomaly.

For Threat 5 - Denial-of-Service attack we consider the events associated with the source and the destination. Hence, attributes A, B and C will be considered to substantiate this attack.

The rationale for using data cubes is (i) we can readily utilize the advantages of OLAP, such as systematic storage of historical data and fast querying (ii) by linking the threat to the data cube views we can provide a classification about the type of threat and the possible solutions and (iii) the scaling of the entire system to meet the ever growing needs of a computer network.

## IV. Outlier Detection

In statistics, an outlier is a single observation "far away" from the rest of the data. Often statistics derived from data sets that include outliers are misleading. For example, if we are calculating the average temperature of some objects in a room, and most are between $20-25^{o}$C, but an oven is at $350^{o}$C, the median of the data may be 23 and the average temperature may be 55. In this case, the median is more likely to reflect the temperature of a randomly sampled object than the mean. Therefore, outliers may be indicative of data points that belong to a different population than the rest of the sample set. In this view, outliers can be of significant use in intrusion detection applications for detecting abnormal conditions. Hence, before we explore the use of outliers for intrusion detection, we formally define the concepts and notations used in the rest of the paper.

### *A. Axioms*

In this section we formally define the concepts and notations used in the rest of this paper.

$A_1, A_2, \ldots, A_m$ are the $m$ attributes we consider in our study, and $D_1, D_2, \ldots, D_m$ are their domains respectively. Let $z^{(i)}$ be the $i$th incident, and $z^{(i)}.A_j$ be the value of the $j$th attribute of incident $i$. $z^{(i)}$ can be represented as $z^{(i)} = (z_1^{(i)}, z_2^{(i)}, \ldots, z_m^{(i)})$, where $z_k^{(i)} = z^{(i)}.A_k \in D_k, k \in \{1, \ldots, m\}$. $Z$ is the set of all incidents.

Definition 1. The Cell
The concept of the cell and other concepts used in this paper are similar to the concepts used in the field of OLAP [17], [25].

$Cell$ $c$ is a vector of the values of attributes with dimension $t$, where $t \leq m$. Therefore, a cell is a subset of the Cartesian product of $D_1 \times D_2 \times \ldots \times D_m$. A cell can be represented as $c = (c_{i_1}, c_{i_2}, \ldots, c_{i_t})$, where $i_1, \ldots, i_t \in \{1, \ldots, m\}$, and $c_{i_s} \in D_{i_s}$. In order to standardize the definition of a cell, for each $D_i$, we add a "wildcard" element "*". Now we allow $D'_i = D_i \cup \{*\}$. For cell $c = (c_{i_1}, c_{i_2}, \ldots, c_{i_t})$, we can represent it as $c = (c_1, c_2, \ldots, c_m)$, where $c_j \in D'_j$, and $c_j = *$ if and only if $j \notin \{i_1, i_2, \ldots, i_t\}$. $c_j = *$ means that we do not care about the value on the $j$th attribute. $C$ denotes the set of all cells. Since each incident can also be treated as a cell, we define a function $Cell : Z \rightarrow C$. if $z = (z_1, z_2, \ldots, z_m)$, $Cell(z) = (z_1, z_2, \ldots, z_m)$.

Definition 2. Contains
Cell $c = (c_{i_1}, c_{i_2}, \ldots, c_{i_t})$ contains incident $z$ if and only if $z.A_j = c_j$, $j \in \{i_1, \ldots, i_t\}$. With the "wildcard" element *, we can also say that cell $c = (c_1, c_2, \ldots, c_m)$ contains incident $z$ if and only if $z.A_j = c_j$ or $c_j = *$, $j = 1, 2, \ldots, m$. We also say cell $c' = (c'_1, c'_2, \ldots, c'_m)$ contains cell $c = (c_1, c_2, \ldots, c_m)$ if and only if $c'_j = c_j$ or $c'_j = *$, $j = 1, 2, \ldots, m$.

Definition 3. Cell contents
We define function $content$ as $content(c) : C \rightarrow 2^Z$, which returns all the incidents that cell $c$ contains. $content(c) = \{z | cell\ c\ contains\ z\}$.

Definition 4. Count
Function $count$ is defined in a natural way over the non-negative integers. $count(c)$ is the number of incidents that cell $c$ contains. $count(c) = |content(c)|$.

Definition 5. Parent Cell
Cell $c' = (c'_1, c'_2, \ldots, c'_m)$ is the $parent\ cell$ of cell $c$ on the $k$th attribute when $c'_k = *$ and $c'_j = c_j$, for $j \neq k$. Function $parent(c, k)$ returns $parent\ cell$ of cell $c$ on the $k$th attribute.

Definition 6. Neighborhood
$P$ is called the $neighborhood$ of cell $c$ on the $k$th attribute when $P$ is a set of cells that takes the same values as cell $c$ in all attributes but $k$, and does not take the wildcard value $*$ on the $k$th attribute, i.e., $P = \{c^{(1)}, c^{(2)}, \ldots, c^{(|P|)}\}$ where $c_l^{(i)} = c_l^{(j)}$ for all $l \neq k$, and $c_k^{(i)} \neq *$ for all $i = 1, 2, \ldots, |P|$. Function $neighbor(c, k)$ returns the neighborhood of cell $c$ on attribute $k$. Neighborhood can also be defined in another way: the neighborhood of cell $c$ on attribute $k$ is a set of all cells whose parent on the $k$th attribute are same as cell $c$.

The above six definitions are obtained directly from the OLAP area with some changes to the words used as discussed in [14]. Having introduced a common set of notations for the data cube terminology, we now derive the necessary mathematical expressions to establish the concept of Mutual Outliers.

Definition 7. Relative Frequency

$$freq(c, k) = \frac{count(c)}{\sum_{c' \in neighbor(c,k)} count(c')} \tag{1}$$

We call $freq(c, k)$ given by Equation (1) as $relative frequency$ of cell $c$ with respect to attribute $k$. The relative frequency can also be defined as:

$$freq(c, k) = \frac{count(c)}{count(parent(c, k))} \tag{2}$$

Definition 8. Uncertainty Function
Next, we define function $U$ to measure the uncertainty of a neighborhood. This uncertainty measure is defined in terms of the relative frequencies of the neighborhood. If we use $P = \{c^{(1)}, c^{(2)}, \ldots, c^{(|P|)}\}$ to denote the neighborhood of cell $c$ on attribute $k$, then $U : R^{|P|} \rightarrow R^{+}$, where

$$U(c, k) = U(freq(c^{(1)}, k), freq(c^{(2)}, k), \ldots, freq(c^{(|P|)}, k)).$$

$U$ is symmetric for all cells $c^{(1)}, c^{(2)}, \ldots, c^{(|P|)}$. $U$ takes a smaller value if the "uncertainty" in the neighborhood is low. One candidate uncertainty function that satisfies the above properties is the entropy: $H(X) = -\sum p_i \log(p_i)$. Now, we have $U(c, k) = H(c, k)$, where

$$H(c, k) = -\sum_{c' \in neighbor(c,k)} freq(c', k) \log(freq(c', k) \tag{3}$$

This is also the expression for entropy, conditional on the neighborhood. When $freq = 0$, we define $0. \log(0) = 0$, as in information theory.

The uncertainty function provides an estimate as to how much abnormal an incident is with respect to its neighbors. Hence, using this function we can establish a score value to determine the extent to which an anomaly has occurred. We name this score value as the group outlier score and we define it in the next section.

### *B. Group Outlier Score*

Function $G$ is used to measure the abnormality of an incident and we define it as the Group Outlier Score (GOS). It is the ratio between the information ensued by an incident and that of its neighbors. The more abnormal an incident is, the higher the value the GOS gets. Function $G$ is defined as:

$$G(c, k) = -\Big(\frac{\log(freq(c, k))}{H(c, k)}\Big) \tag{4}$$

When $H(c, k) = 0$, we say $\frac{log(freq(c,k))}{H(c,k)} = 0$. We also verify that this function satisfies the following properties:

1) If $c^{(1)}$ and $c^{(2)}$ are two one dimension cells, and both of them take non-* values on the same attribute, then $G(c^{(1)}, k) \geq G(c^{(2)}, k)$, iff $count(c^{(1)} \leq count(c(2))$.
2) Assume that $c^{(1)}$ and $c^{(2)}$ are two one-dimension cells, and they take non-* values on two different attributes, say $i$ and $j$ respectively. If $freq(c^{(1)}) = freq(c^{(2)})$, then $G(c^{(1)}, i) \geq G(c^{(2)}, j)$ holds iff $H(c^{(1)}, i) \leq H(c^{(2)}, j)$.
3) $G(c^{(1)}, k) \geq G(c^{(2)}, k)$ always holds if $\exists k, c^{(2)} = parent(c^{(1)}, k)$.

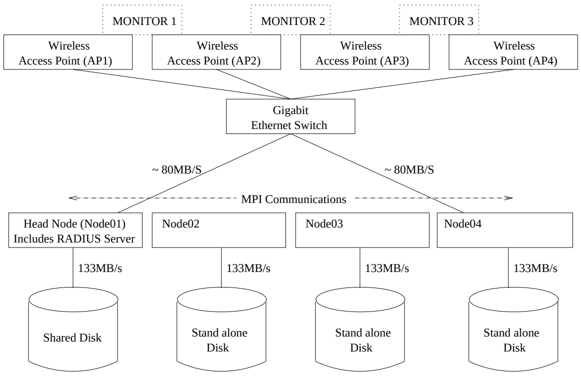

Figure 3. The Experimental Setup

From the above declarations it could be deduced that in an ideal case the GOS value of an event with no abnormality as one. This is becuase the information ensued by all normal events must be the same. Hence, in practical situations we can categorize events with GOS values away by a set threshold value from the ideal value as outliers. In the next section we describe our proposed early warning system with an example illustrating the above concept of substantiation.

## V. Experiments And Results

Our experimental setup (Figure 3) consists of a small Beowulf cluster and several 802.11i enabled APs. The APs are configured to communicate with IEEE 802.11i enabled STAs. The Beowulf cluster has four nodes including the head node. The head node is a Pentium 4 machine with 1GB internal memory and 120GB secondary storage. The back nodes are Pentium 4 machines with 1GB internal memory and 40GB secondary storage. The head node includes a Dlink AG530 wireless adapter configured to capture wireless network traffic in promiscuous mode. The captured packets are passed to the event engine for further processing. The head node also runs the RADIUS server software and acts as an authentication server for the access points. Our present setup has only one monitoring device. However, in a distributed system there will be a number of monitors talking to the central intrusion prevention module. Thus our system could be easily enhanced to handle a distributed environment.

Table IV lists a sample of the raw wireless network traces captured from our experimental setup. In these traces STA1 represents the MAC address of a wireless station and AP1 represents the MAC address of an access point. Since our aim is to investigate anomalies during the RSNA we concentrate only on the management frames. The above traces relate to the RSNA between STA1 and AP1 with EAP-TLS authentication.

In our EWS, when timing and/or behavioral anomalies are detected, it is vital to verify the legitimacy of such anomalies. The intrusion prevention module plays a major role by validating and substantiating the anomalies

TABLE IV.
RSNA EVENTS

| |
|---|
| 1 16.51404: $STA1- > AP1$ Association Request |
| 2 16.51458: $AP1- > STA1$ Association Response |
| 3 16.51508: $AP1- > STA1$ EAP Request, Identity |
| 4 16.51628: $STA1- > AP1$ EAP Response, Identity |
| 5 16.53508: $AP1- > STA1$ EAP Request, PEAP |
| 6 16.53588: $STA1- > AP1$ EAP Response, Nak |
| 7 16.56430: $AP1- > STA1$ EAP Request, EAP-TLS |
| 8 16.56894: $STA1- > AP1$ TLS Client Hello |
| 9 16.59350: $AP1- > STA1$ EAP Request, EAP-TLS |
| 10 16.59435: $STA1- > AP1$ EAP Response, EAP-TLS |
| 11 16.63310: $AP1- > STA1$ TLS Server Hello .. |
| 12 16.66166: $STA1- > AP1$ EAP Response, EAP-TLS |
| 13 16.67270: $AP1- > STA1$ EAP Request, EAP-TLS |
| 14 16.67609: $STA1- > AP1$ TLS Certificate .. |
| 15 16.69887: $AP1- > STA1$ TLS Change Cipher .. |
| 16 16.70020: $STA1- > AP1$ EAP Response, EAP-TLS |
| 17 16.70363: $AP1- > STA1$ EAP Success |

detected by the previous modules. In this context we present the results obtained from analyzing the wireless traces captured during RSNAs between a number of stations and access point in our experimental wireless networking environment. Firstly, we collected about ten million wireless traces (over a period of one week) from a controlled environment to build the initial data cube. A special capturing tool was developed for this purpose [26]. The traces were converted with suitable mappings to suit the requirements of the data cube algorithms [27].

Once the captured traces are converted using a parsing program, we build the data cube with the attributes listed in Table I. The average time taken to build the data cube was 1.32 seconds. The time taken for indexing the data cube was arround 1.51 seconds in average. With a dimension of seven we get 128 different views of the captured traces. However, we built the data cube with only those selected views (Table II) necessary to process our queries. For example, consider the following data associations needed for analyzing network anomalies between stations and access points.

Source ID, Destination ID, Event ID
Source ID, Event ID, Protocol ID
Destination ID, Event ID, Protocol ID
Source ID, Destination ID
Source ID, Event ID
Destination ID, Event ID

As discussed in Section IV, In order to process these queries we require only ABC, ACF, BCF, AB, AC and BC views. However, depending on the anticipated security threats we may need to analyze more number of views. Next, we establish the GOS under normal conditions.

### *A. Normal Conditions*

Before exploring the application of GOS in substantiating abnormal events within associated group of events, we analyze the GOS values under normal conditions to establish threshold values. For this purpose we first analyze the GOS values for EAP-TLS events under normal conditions.

TABLE V.
GOS FOR EAP-TLS EVENTS UNDER NORMAL OPERATION

| No. | Src. ID | Dest. ID | Event ID | Count | GOS |
|---|---|---|---|---|---|
| 1 | 128 | 140 | 0 | 126 | 1.07 |
| 2 | 140 | 128 | 0 | 125 | 1.07 |
| 3 | 140 | 128 | 1 | 134 | 1.05 |
| 4 | 128 | 140 | 1 | 131 | 1.05 |
| 5 | 140 | 128 | 3 | 138 | 1.04 |
| 6 | 128 | 140 | 4 | 137 | 1.04 |
| 7 | 128 | 140 | 6 | 129 | 1.06 |
| 8 | 140 | 128 | 7 | 137 | 1.04 |
| 9 | 128 | 140 | 9 | 141 | 1.03 |
| 10 | 140 | 128 | 10 | 132 | 1.05 |
| 11 | 128 | 140 | 23 | 143 | 1.03 |
| 12 | 140 | 128 | 24 | 100 | 1.14 |
| 11 | 128 | 140 | 42 | 121 | 1.08 |
| 12 | 140 | 128 | 43 | 140 | 1.03 |
| 13 | 128 | 140 | 44 | 134 | 1.05 |
| 14 | 140 | 128 | 45 | 135 | 1.04 |
| 16 | 128 | 140 | 47 | 137 | 1.04 |
| 17 | 128 | 249 | 49 | 127 | 1.06 |
| 18 | 140 | 128 | 50 | 121 | 1.08 |
| 19 | 128 | 140 | 51 | 127 | 1.06 |

In order to establish the GOS values under normal conditions we used the initial data collected from the controlled environment. Table V show the GOS values obtained for EAP-TLS events under normal conditions. Here, the count values show the number of events captured during the test period. The GOS values in the table are calculated using Equation 4. In order to calculate the GOS we need to know the relative frequency (Equation 2) of the event and its uncertainty value (Equation 3). For example if we consider event $43$ in Table V it has a count of $140$. To calculate the relative frequency we need to make a second query to find the parent count, which in this case was $2615$. Then the GOS value is calculated as follows:

Using Equation 2

$$\begin{aligned} freq(c,k) &= 140/2615 \\ log(freq(c,k)) &= -1.271 \end{aligned}$$

Using Equation 3

$$\begin{aligned} H(c,k) &= -\sum_{c' \in neighbor(c,k)} freq(c',k)\log(freq(c',k) \\ &= 1.232 \end{aligned}$$

Using Equation 4

$$\begin{aligned} G(c,k) &= -\Big(\frac{\log(freq(c,k))}{H(c,k)}\Big) \\ Hence,\ G(c,k) &= -\Big(\frac{-1.271}{1.232}\Big) \\ &= 1.03 \end{aligned}$$

Having calculated the GOS values, it is essential to establish a viable threshold for effective substantiation of anomalies. In order to do this we used the GOS values where the number event count is the lowest. It could be noticed from the table that the GOS values increase as the event count decrease. When the event count decreases with reference to other associated events it means the event is deviating from its normal behavior.

TABLE VI.
GOS Threshold Value

| Authentication | Max GOS | Threshold GOS |
|---|---|---|
| EAP-TLS | 1.14 | 1.20 |

From table V the maximum GOS values for EAP-TLS authentication scheme is 1.14. However, since we must also accommodate legitimate reduction in count which may arise due to the inherent nature of the wireless environment, we use the maximum GOS value with a tolerance of 5% to fix the threshold value. Table VI shows the threshold value thus derived for EAP-TLS authentication scheme.

### B. *Replay Attack*

The query results shown in Table VII were obtained using view ABCD. Here attributes A, B, C and D represent ”Src. ID”, ”Dest. ID”, ”Event ID” and “Time of Day” respectively. In this query attribute D has the value of 11 (11am to 12noon). The query result shows the number of events associated with a particular station during a particular time period. Here Source/Dest ID 247 refers to station STA3 and Source/Dest ID 128 refers to access point AP1. Event IDs ranging from 3 to 51 refers to various events, such as 3 representing “Association Request”, 4 representing “Association Response”, 6 representing “EAP Request Identity 1” etc. Table VIII shows two abnormal events captured during a different time period (16). From this table it is evident that station STA3 is issuing some superfluous messages without any prerequisites. In such a scenario if we consider only view ABCE, provided there are no other messages passed on other channels, we will have an accumulated count of 43 for event with ID 7, and 31 for event with ID 43 as shown in Table IX.

With reference to the GOS values in Table IX, if we set a threshold value of 1.14 (as discussed above) to detect rare events, our system will report both events as normal on view ABCE and one event as abnormal on view ABCD. Since events with ID 7 and 43 do not have any other associations during this time period we can substantiate this as an anomaly and categorize it as a Replay Attack because of view ABCD (Section IV). In another similar experiment we considered a different incident involving view ABCE, triggered due to a behavioral anomaly. Table X shows the query results obtained using view ABCE on channel 11 for station STA4. In this scenario we can see two events (with ID’s 9 & 10) having a very low count. The GOS value for those two events worked out to be 2.53, revealing them as rare events. Thus we substantiate those two incidents as anomalies and categorize it as a Replay Attack because of views ABCE and ABCD.

TABLE VII.
RSNA Events on STA3

| Seq. No. | Src. ID | Dest. ID | Event ID | Count |
|---|---|---|---|---|
| 1 | 247 | 128 | 3 | 34 |
| 2 | 128 | 247 | 4 | 36 |
| 3 | 128 | 247 | 6 | 32 |
| 4 | 247 | 128 | 7 | 41 |
| 5 | 128 | 247 | 20 | 36 |
| 6 | 247 | 128 | 21 | 41 |
| 7 | 128 | 247 | 40 | 36 |
| 8 | 247 | 128 | 41 | 36 |
| 9 | 128 | 247 | 42 | 33 |
| 10 | 247 | 128 | 43 | 32 |
| 11 | 128 | 247 | 44 | 33 |
| 12 | 247 | 128 | 45 | 38 |
| 13 | 128 | 247 | 47 | 35 |
| 14 | 247 | 128 | 48 | 28 |
| 15 | 128 | 247 | 49 | 32 |
| 16 | 247 | 128 | 50 | 39 |
| 17 | 128 | 247 | 51 | 41 |

TABLE VIII.
Abnormal RSNA Events on STA3

| Seq. No. | Src. ID | Dest. ID | Event ID | Count |
|---|---|---|---|---|
| 4 | 247 | 128 | 7 | 2 |
| 10 | 247 | 128 | 43 | 1 |

TABLE IX.
GOS from different Views

| View | Event ID | count | GOS |
|---|---|---|---|
| ABCE | 7 | 43 | 1.03 |
| ABCE | 43 | 33 | 1.01 |
| ABCD | 7 | 2 | 1.11 |
| ABCD | 43 | 1 | 4.06 |

TABLE X.
RSNA Events on STA4

| Seq. No. | Src. ID | Dest. ID | Event ID | Count |
|---|---|---|---|---|
| 1 | 249 | 128 | 3 | 17 |
| 2 | 128 | 247 | 4 | 62 |
| 3 | 128 | 249 | 6 | 33 |
| 4 | 249 | 128 | 7 | 64 |
| 5 | 128 | 249 | 9 | 2 |
| 6 | 249 | 128 | 10 | 2 |
| 7 | 128 | 249 | 20 | 38 |
| 8 | 249 | 128 | 21 | 39 |
| 9 | 128 | 249 | 40 | 35 |
| 10 | 249 | 128 | 41 | 37 |
| 11 | 128 | 249 | 42 | 45 |
| 12 | 249 | 128 | 43 | 62 |
| 13 | 128 | 249 | 44 | 46 |
| 14 | 249 | 128 | 45 | 39 |
| 15 | 128 | 249 | 47 | 39 |
| 16 | 249 | 128 | 48 | 38 |
| 17 | 128 | 249 | 49 | 35 |
| 18 | 249 | 128 | 50 | 44 |
| 19 | 128 | 249 | 51 | 40 |

To further illustrate the effectiveness of our substantiation mechanism we present the GOS values of the various incidents associated with the different stations in a single representation. Figure 4 shows the GOS values of RSNA events during an EAP-TLS authentication for stations STA2, STA3 and STA4. The GOS values of all events for station STA2 are in the range of 1.0 to 1.14, and therefore we consider it as normal. Whereas, station ST3 has two events with high GOS values. These are the two events with ID 7 and 43 (in Figure 4 events 5 and 11), which are usually present during a regular EAP-TLS authentication process. However, in this case these two events have been captured in isolation without the other relevant messages. Hence when we calculate their GOS values on view ABCD, they appear to be high and

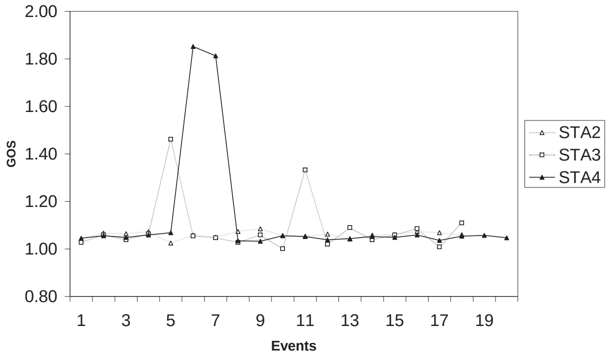

Figure 4. Normal and Abnormal RSNA Events

therefore we categorized it as due to a Replay Attack. Similarly, station STA4 also has two events with high GOS values. Events with ID's 9 and 10 (in Figure 4 events 6 and 7) are considered abnormal because these two events do not match the normal behavior of station STA4 during an EAP-TLS authentication process. The GOS values of these two events on view ABCE are also high. Hence, with a threshold value of 1.2 we categorize those events as abnormal.

Above, we have illustrated some experiments showing the use of GOS values for detecting rare events and grouping events based on their remoteness. Detecting rare events in the wireless environment could be useful in identifying any unusual messages passed between the wireless hosts that may eventually lead to an impending security breach. In the case of an impending security threat it would be appropriate to categorize the threat so that relevant warning alarms could be raised.

### C. Malicious Association

As discussed in Section IV, attackers can perform malicious associations with a legitimate wireless host during the discovery phase. This can be done by masquerading as a legitimate host and providing wrong credentials to the the access point. This is done by providing invalid RSN IE information in the "Association Request" message. Having received the wrong credentials, the access point will ignore the legitimate host assuming it as RSN incapable. Thereafter, the intruder host can masquerade as the legitimate access point and continue the association with the legitimate host.

The query results shown in Table XI were obtained using view ABCF. Here attributes A, B, C and F represent ”Src. ID”, ”Dest. ID”, ”Event ID” and "Protocol ID" respectively. The query result shows the number of events associated with a particular station during EAP-LEAP authentication process. Here Source/Dest ID 247 refers to station STA3 and Source/Dest ID 128 refers to access point AP1. Event IDs ranging from 0 to 21 refers to the various EAP-LEAP events as listed in Table XII, such as 3 representing "Association Request", 4 representing "Association Response", 6 representing "EAP Request Identity 1" and so on.

TABLE XI.
MALICIOUS ASSOCIATION

| No. | Src. ID | Dest. ID | Event ID | Count |
|---|---|---|---|---|
| 1 | 128 | 247 | 0 | 14 |
| 2 | 247 | 128 | 0 | 13 |
| 3 | 128 | 247 | 1 | 14 |
| 4 | 247 | 128 | 2 | 15 |
| 5 | 247 | 128 | 3 | 31 |
| 6 | 128 | 247 | 4 | 13 |
| 7 | 128 | 247 | 6 | 14 |
| 8 | 247 | 128 | 7 | 14 |
| 9 | 128 | 247 | 11 | 15 |
| 10 | 247 | 128 | 12 | 14 |
| 11 | 128 | 247 | 13 | 13 |
| 12 | 247 | 128 | 14 | 16 |
| 13 | 128 | 247 | 18 | 14 |
| 14 | 128 | 247 | 20 | 13 |
| 15 | 247 | 128 | 21 | 15 |

TABLE XII.
EAP-LEAP EVENTS DURING NORMAL BEHAVIOR

| ID | Event |
|---|---|
| 0 | OPEN AUTHENTICATION |
| 0 | OPEN AUTHENTICATION |
| 3 | OPEN ASSOCIATION REQUEST |
| 4 | OPEN ASSOCIATION RESPONSE |
| 6 | EAP REQUEST 1 (IDENTITY) |
| 7 | EAP RESPONSE 1 (IDENTITY) |
| 20 | EAP REQUEST 2 (PEAP) |
| 21 | EAP RESPONSE 2 (NAK) |
| 11 | EAP REQUEST 3 (LEAP) |
| 12 | EAP RESPONSE 3 (LEAP) |
| 13 | EAP SUCCESS |
| 14 | EAP REQUEST 4 (LEAP) |
| 18 | EAP RESPONSE 4 (LEAP) |
| 52 | EAPOL KEY |
| 52 | EAPOL KEY |
| 52 | EAPOL KEY |
| 52 | EAPOL KEY |

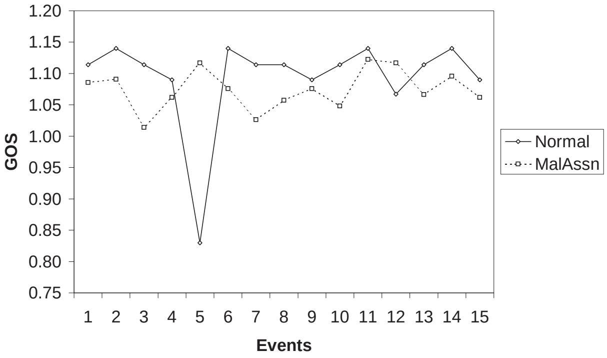

Figure 5. GOS for Malicious Association

The events and their corresponding counts listed in Table XI were obtained during a malicious association attack. According to the table, event number 3 has almost double the number of count compared with the other events. This is due to the "Association Request" message sent by the intruder and the legitimate host. Event 3 - "Association Request" has a count of thirty one compared to the thirteen "EAP Success" messages (Event 13).

Figure 5 shows the effect of GOS values in this type of attack. As in the figure, except for event 5 rest of the events are within the allowable range for GOS. However, in this case since the abnormal event has higher count than the normal events the GOS value for the abnormal event assumes a value less than 1.00. In the next example

TABLE XIII.
GOS VALUES DURING A SESSION HIJACK

| No. | Src. ID | Dest. ID | Event ID | GOS |
|---|---|---|---|---|
| 1 | 249 | 128 | 3 | 1.05 |
| 2 | 128 | 247 | 4 | 1.04 |
| 3 | 128 | 249 | 6 | 1.05 |
| 4 | 249 | 128 | 7 | 1.06 |
| 5 | 128 | 249 | 9 | 1.07 |
| 6 | 249 | 128 | 10 | 1.06 |
| 7 | 128 | 249 | 20 | 1.03 |
| 8 | 249 | 128 | 21 | 1.03 |
| 9 | 128 | 249 | 40 | 1.05 |
| 10 | 249 | 128 | 41 | 1.05 |
| 11 | 128 | 249 | 42 | 1.03 |
| 12 | 249 | 128 | 43 | 1.04 |
| 13 | 128 | 249 | 44 | 1.05 |
| 14 | 249 | 128 | 45 | 1.08 |
| 15 | 128 | 249 | 46 | 1.24 |
| 16 | 128 | 249 | 47 | 1.06 |
| 16 | 249 | 128 | 48 | 1.03 |
| 17 | 128 | 249 | 49 | 1.06 |
| 18 | 249 | 128 | 50 | 1.04 |
| 19 | 128 | 249 | 51 | 1.05 |

we discuss an attack with less number of abnormal events than normal events.

### D. Session Hijack

A Session Hijack is a more advanced attack, where the association between a legitimate station and the access point is hijacked by an illegitimate user. In this case the illegitimate user can force a channel change with the access point/station and masquerade as a legitimate access point/station. Hence, in this kind of a threat we need to track the source, destination, event, channel and protocol of the messages exchanged. By tracking the protocol we establish whether the illegitimate session establishes a different kind of association. Therefore, for this type of a threat, views associated with attributes A, B, C, E and F are are considered. We now consider this vulnerability and discuss how it could be mitigated using our outlier detection technique.

Table XIII shows the GOS values for EAP-TLS events obtained from view ABCF. Here except for event 46, “EAP Failure”, all other events have GOS values less than 1.2. In this scenario although the GOS value of the “EAP Failure” event is above the threshold value, the message itself cannot be considered abnormal because it is a possible behavior during a EAP-TLS authentication process. Therefore, to further verify this abnormality we consider this scenario from a different view point. Hence, we consider view ABCE which includes the channel ID of the communication.

As could be seen in Table XIV the GOS values for all events above and including event 46 are different than that of all events less than event 46. This concludes that this abnormal condition is due to a session hijack attack. The intruder does a channel change with the legitimate host with event 46 and then continues to masquerade as a legitimate access point duping the legitimate host. The legitimate host unknowingly associates with the intruder.

TABLE XIV.
GOS VALUES FROM A DIFFERENT VIEW

| No. | Src. ID | Dest. ID | Event ID | GOS |
|---|---|---|---|---|
| 1 | 249 | 128 | 3 | 1.08 |
| 2 | 128 | 247 | 4 | 1.08 |
| 3 | 128 | 249 | 6 | 1.09 |
| 4 | 249 | 128 | 7 | 1.10 |
| 5 | 128 | 249 | 9 | 1.08 |
| 6 | 249 | 128 | 10 | 1.09 |
| 7 | 128 | 249 | 20 | 1.08 |
| 8 | 249 | 128 | 21 | 1.09 |
| 9 | 128 | 249 | 40 | 1.08 |
| 10 | 249 | 128 | 41 | 1.09 |
| 11 | 128 | 249 | 42 | 1.08 |
| 12 | 249 | 128 | 43 | 1.10 |
| 13 | 128 | 249 | 44 | 1.09 |
| 14 | 249 | 128 | 45 | 1.10 |
| 15 | 128 | 249 | 46 | 1.24 |
| 16 | 128 | 249 | 47 | 1.24 |
| 16 | 249 | 128 | 48 | 1.24 |
| 17 | 128 | 249 | 49 | 1.24 |
| 18 | 249 | 128 | 50 | 1.24 |
| 19 | 128 | 249 | 51 | 1.24 |

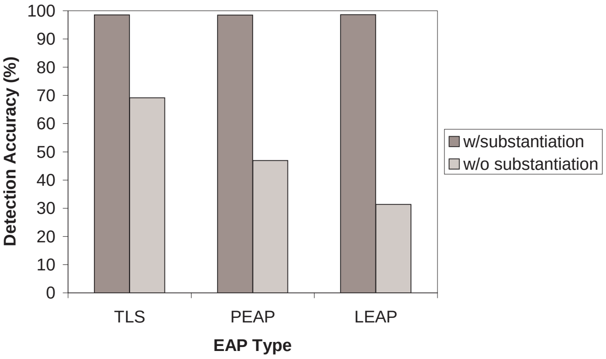

Figure 6. EWS Accuracy for Replay Attack

## VI. EFFECTIVENESS OF GOS

Figure 6 shows the accuracy of EWS with and without substantiation of EAP-TLS, PEAP and LEAP authenticated hosts during a Replay attack. Here the abnormalities were forced with some having an effect on thewireless hosts and the others being replications. The EWS without substantiation raised large number of false alarms and the EWS with substantiation demonstrated 99% accuracy for a threshold value of 1.2. The large number of alarms raised when substantiation was not effective was due to the fact that the EWS was merely reporting all timing and/or behavior anomalies. According to figure 6 all three authentication mechanisms demonstrate high accuracy with substantiation. Whereas, without substantiation the accuracy of EAP-LEAP authenticated hosts are less compared to that of EAP-TLS and PEAP. Next, we establish the relation between the threshold value and the accuracy.

Figure 7 shows the accuracy with and without substantiation of EAP-TLS, PEAP and LEAP authenticated hosts during a DoS attack. Here again all three authentication mechanisms demonstrate high accuracy with substantiation. However, the accuracy of EAP-LEAP authenticated hosts is less without substantiation as in the case of replay attack. Hence, it is evident that EAP-LEAP authentication is even more vulnerable and comprises of many ambiguities.

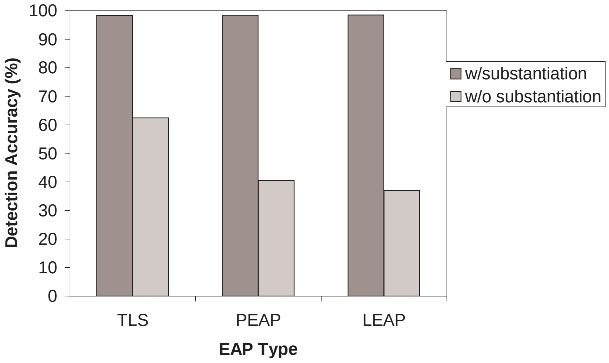

Figure 7. EWS Accuracy for DoS Attack

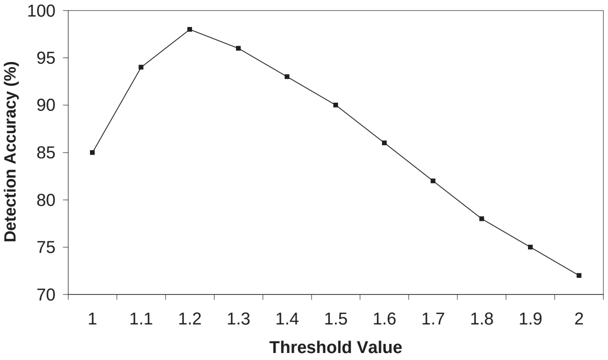

Figure 8. EWS Accuracy on GOS Threshold

In view of analyzing the effectiveness of the substantiation mechanism for different threshold values, we studied the performance of the EWS introducing various types abnormalities. Figure 8 shows the accuracy of the substantiation mechanism for different threshold values. Accordingly, in this scenario the best value for the threshold value appears to be 1.2. However, we cannot fix the threshold level to any particular value in practice and it will be left to the discretion of the EWS to adaptively choose the best possible value depending on the nature of operation at any given time.

## VII. Conclusions

In this paper, we have discussed the use of GOS values to detect rare events and associate them to a security threat in the wireless environment. The inspiration with the GOS value is that it could be used to detect individual rare/frequent events and to group associated events. Hence, it was discussed how various events could be grouped based on the GOS values. The GOS gives a suggestion as to which events can be grouped based on their association. We have also shown the effectiveness of our mechanism with and without substantiation and its performance with different threshold values for GOS.

The main contribution of this study is the substantiation mechanism used to validate anomalies. Although anomalies can be of several forms, detecting rare events and grouping them based on their association is challenging. Our concept could be applied to several fields including credit card transactions, health monitoring systems, Internet security, maritime border security, air traffic control and the like. The wireless environment considered in this paper is one such example where the number of anomalies and their nature vary drastically. Hence, we have demonstrated that our validation mechanism is capable in managing such situations successfully. Since this study was mainly focused on establishing the effectiveness of GOS, we have not considered comparing the anomaly detection capabilities of GOS with other methods.

In the future work, we intend to compare our system with similar systems to demonstrate the effectiveness of our system. Further, we intend to use this concept in other applications to detect both rare and frequent events that are significant in categorizing an abnormal behavior.

## VIII. Acknowledgments

The software for generating and querying the data cube were obtained from Prof. Todd Eavis at the Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada.

## References

[1] E. M. Knorr and R. T. Ng, “Algorithms for mining distance-based outliers in large datasets,” in *VLDB ’98: Proceedings of the 24th International Conference on Very Large Data Bases*, 24–27 1998, pp. 392–403. [Online]. Available: citeseer.ist.psu.edu/knorr98algorithm.html

[2] E. M. Knorr, R. T. Ng, and V. Tucakov, “Distance-based outliers: Algorithms and applications,” *VLDB Journal: Very Large Data Bases*, vol. 8, no. 3–4, pp. 237–253, 2000. [Online]. Available: citeseer.ist.psu.edu/knorr00distancebased.html

[3] T. Johnson, I. Kwok, and R. T. Ng, “Fast computation of 2-dimensional depth contours,” in *Proceedings of the Knowledge Discovery and Data Mining Conference*, 1998, pp. 224–228. [Online]. Available: citeseer.ist.psu.edu/johnson98fast.html

[4] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999. [Online]. Available: citeseer.ist.psu.edu/jain99data.html

[5] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF: identifying density-based local outliers,” *ACM SIGMOD*, vol. 29, no. 2, pp. 93–104, june 2000. [Online]. Available: citeseer.ist.psu.edu/breunig00lof.html

[6] S. Sarawagi, R. Agrawal, and N. Megiddo, “Discovery-driven exploration of olap data cubes,” in *EDBT ’98: Proceedings of the 6th International Conference on Extending Database Technology*. London, UK: Springer-Verlag, 1998, pp. 168–182.

[7] E. Sithirasenan and V. Muthukkumarasamy, “Substantiating Security Threats Using Group Outlier Detection Techniques,” in *IEEE Globecom’08: Proceedings of the 51st IEEE International Global Communication Conference*, December 2008, pp. 1–6.

[8] W. Lee and S. Stolfo, “Data mining approaches for intrusion detection,” in *Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX, 1998. [Online]. Available: citeseer.ist.psu.edu/article/lee98data.html

[9] W. Lee, S. J. Stolfo, and K. W. Mok, “Mining audit data to build intrusion detection models,” in *Proceedings of the Knowledge Discovery and Data Mining Conference*, 1998, pp. 66–72. [Online]. Available: citeseer.ifi.unizh.ch/lee98mining.html

[10] S. Manganaris, M. Christensen, D. Zerkle, and K. Hermiz, “A data mining analysis of rtid alarms,” *Comput. Networks*, vol. 34, no. 4, pp. 571–577, 2000.

[11] E. Bloedorn, A. D. Christiansen, W. Hill, C. Skorupka, L. M. Talbot, and J. Tivel, “Data mining for network intrusion detection: How to get started,” http:// www.mitre.org/work/tech_ papers/tech_ papers_01/bloedorn_ datamining/bloedorn_ datamining.pdf, August 2001.

[12] W. Jin, A. K. H. Tung, and J. Han, “Mining top-n local outliers in large databases,” in *Proceedings of the Knowledge Discovery and Data Mining Conference*, 2001, pp. 293–298. [Online]. Available: citeseer.ist.psu.edu/440808.html

[13] C. C. Aggarwal and P. S. Yu, “Outlier detection for high dimensional data,” *ACM SIGMOD*, vol. 30, no. 2, pp. 37–46, June 2001. [Online]. Available: citeseer.ist.psu.edu/264937.html

[14] S. Lin and D. Brown, “An outlier-based data association method for linking criminal incidents,” *Decision Support Systems*, vol. 41, no. 3, pp. 604–615, March 2006. [Online]. Available: http://www.sciencedirect.com/science/article/B6V8S-4DFT4HY-1/2/72b8c46bff2d3906934424feb82dab3b

[15] S. Lin and D. E. Brown, “Outlier-based data association: Combing olap and data mining,” Dept. of Systems Engineering, University of Virginia, Tech. Rep. SIE-020011, 2002.

[16] E. Sithirasenan, Y. Chen, F.Dehne, T. Eavis, A. Chaplin, and D. Green, “cgmOLAP: Efficient Parallel Generation and Querying of Terabyte Size ROLAP Data Cubes,” in *ICDE ’06: Proceedings of the 22nd International Conference in Data Engineering*, April 2006, pp. 164–167.

[17] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh, “Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals,” *J. Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 29–53, 1997. [Online]. Available: citeseer.ist.psu.edu/gray97data.html

[18] F. Dehne, T. Eavis, S. Humbrusch, and A. Chaplin, “Parallelizing the Data Cube,” *The Journal of Distributed and Parallel Databases*, vol. 11, no. 2, pp. 181–201, 2002.

[19] F. Dehne, T. Eavis, and A. Chaplin, “Top Down Computation of Partial ROLAP Data Cubes,” in *HICSS-37: Proceedings of the 37th Annual Hawaii International Conference On System Sciences*, January 2004, pp. 181–193.

[20] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Amendment 6: Medium Access Control (MAC) Security Enhancements*, IEEE Standard 802.11i Part 11, July 2004.

[21] L. Blunch and J. Vollbrecht, “Extensible Authentication Protocol (EAP),” http:// www.ietf.org/rfc/rfc2284.txt, March 1998.

[22] C. S. Inc., “Cisco LEAP (LEAP),” http://ietfreport.isoc.org/all-ids/draft-josefsson-pppext-eap-tls-eap-07.txt, October 2003.

[23] A. Palekar, D. Simon, G. Zorn, J. Salowey, H. Zhou, and S. Josefsson, “Protected EAP Protocol (PEAP),” http://ietfreport.isoc.org/all-ids/draft-josefsson-pppext-eap-tls-eap-07.txt, October 2003.

[24] B. Aboba and D. Simon, “PPP EAP TLS Authentication Protocol,” http://tools.ietf.org /wg/pppext/draft-ietf-pppext-eaptls/draft-ietf-pppext-eaptls-06.txt, August 1999.

[25] “Olap council,” http://www.olapcouncil.org, cited March 2006.

[26] S. Mathews, “RSN Association Traces - Capturing Tool,” Master’s thesis, School of Communication and Information Technology, Griffith University, Gold Coast, Australia, Ocotober 2006.

[27] Y. Chen, F. Dehne, T. Eavis, and A. Rau-Chaplin, “Parallel ROLAP Datacube Construction on Shared Nothing Multi-Processors,” *The Journal of Distributed and Parallel Databases*, vol. 15, no. 3, pp. 219–236, May 2004.

**Elankayer Sithirasenan** received his PhD degree in network security from Griffith University of Australia in 2009, his MS degree in software engineering from Griffith University in 2004, and his BS degree in electrical engineering and computer science from the University of Peradeniya, Sri Lanka in 1991.

He is currently a lecturer at the Gold Coast campus of Griffith University in Australia. Before joining Griffith University he was a lecturer at the University of Peradeniya in Sri Lanka and has served as a consultant to many financial institutions in Sri Lanka. His research interests include security, data mining, bio security, and cloud computing.

Dr. Sithirasenan is a member of the IEEE Communication Society and the ACS.

**Vallipuram Muthukkumarasamy** received his PhD degree in electrical engineering from Cambridge University in 1990, and his BS degree in electrical engineering and computer science from the University of Peradeniya, Sri Lanka in 1986.

He is currently a Senior Lecturer in Information Technology at Griffith University in Gold Coast Australia. Before joining Griffith University he was a senior lecturer at University of Peradeniya in Sri Lanka. His current research interests include security in wireless networks, intrusion detection and prevention systems, sensors in health systems and trust in e-government models.

Dr. Muthukkumarasamy is a member of the IEEE Communication Society and the IEE.