1. Introduction

Computer security has become a major challenge for all industries. A large number of intrusion detection systems (IDS) have been built to deal with intruders' techniques and attacks. Hackers, however, are always looking for new ways to infiltrate computer networks. In particular, web services outside demilitarized zones are easy targets. There are a large number of intrusion detection systems available and they are based on a range of software and hardware technologies. Hackers are usually very smart in infiltrating software and hardware infrastructures, and have developed many different programs and practices to break into Wintel and UNIX systems, network services, web applications, databases, and other applications. As a result, companies have to spend millions of dollars to protect themselves from attack and on repairing the damage caused by hackers [57]. The 2001Code Red worm cost US$2.6 billion. The Microsoft SQL Sapphire/Slammer worm was very expensive and companies spent US$950–US$1.2 billion in the first week of attack in January 2003 [18] [24]. In 2002, a Computer Security Institute/FBI survey [63] reported substantial damages that costed $455 million. Firewalls were once considered a reliable protection method but intruders have also penetrated through firewalls. Most of the time, intruders hide their identities by changing the system's security parameters or by stealing important company data [46]. Under such circumstances, it is almost impossible to trace attacks using conventional

intrusion detection systems, monitoring systems or firewall information because all the application and systems processes look normal [7].

A large number of businesses use UNIX-based applications for their critical networks as UNIX is a very stable and reliable operating system. UNIX, a backbone of distributed systems nowadays, was once considered to be a secure system over the network. However, recently hackers have developed new ways to break into UNIX systems as well. As a result, the task of developing monitoring and security infrastructure for UNIX-based systems, applications and databases has become a more tedious task than ever. To determine the nature of the attacks they are subjected to, and the damage caused by intruders, companies need automated threat detection tools which are based on a multiple simultaneous threat detection model (MSTDM) [61] [7].

On the other hand, the distributed systems industry, particularly in relation to UNIX security, is facing severe problems because detection tools are yielding high volumes of false positives and false negatives. Existing monitoring and intrusion detection systems could only provide low-level alarm events generated from systems parameters and network traffic. Most of the intrusion detection system sensors are based on processes, memory, disk utilisation, unauthorised attempts to gain access and IP-header information. Events are checked and compared to a database of known attacks (signatures) or with a profile of normal traffic [22] [2].

Usually, Intrusion Detection Systems use single-threat detection models to identify and block hackers' intrusions and false alarms. One can classify a false

alarm as being either misuse detection or an anomaly detection. Network packets and system activities involving known misuses are examined for misuse detection. A pattern matching algorithm is used to verify packets and system data in misuse detections. Parameters or events that do not match the existing known patterns in the databases are considered to be anomalous. Alternatively, mathematical, statistical or machine learning techniques can be used for anomaly detection [52] [59].

It is the aim of this thesis to model, evaluate and build a multiple simultaneous threat detection system (MSTDS) in a UNIX distributed system environment that works alongside advanced intrusion detection systems. This research builds on computer security research on distributed simultaneous multiple sensor data fusion. The outcomes of this research are beneficial to cyberspace researchers as they add to the knowledge about which particular models and algorithms would be relevant for tests on multiple simultaneous threat detection in distributed systems, such as in a network of UNIX systems. Likewise, they will also shed light on which multi-sensor fusion models that will require further research.

1.1 Research Problems

The research problem that this thesis addresses is "multiple simultaneous threats detection" in distributed systems. It will use the UNIX environment as the platform for experimental evaluation. However, the benefits of this research are

more general, and are equally applicable to a distributed network running UNIX, Windows and other networked operating systems.

In the past decade, a lot of research on intrusion detection systems has been done to prevent intrusions, false alarms and financial loss to the computing industry [19] [51] [68]. Most of the intrusion detection systems are based on a multisensor fusion approach that detects one threat at a time, particularly in the UNIX environment. In addition, the mathematical/statistical models that underpin the multisensor fusion approach used so far provide sub-optimal precision in multiple threats detection, resulting in increasing false positives and false negatives. These models are often based on Bayesian Theory, Dempster-Shafer, Extended Dempster-Shafer Theory, and Markov chains, etc. However, none of the existing models has made use of a combination of data rule sets which are attained by applying the set cover theory, followed by applying a hybrid model of Bayesian and Dempster-Shafer Theorems for improved detection [3] [37]. The segregation of data into appropriate groupings adds another optimisation or permutation layer in data analysis that increases the precision of threat detection.

Existing multisensor fusion models were proposed to detect single threats at one particular time, but hackers can attack multiple systems simultaneously in a distributed computer environment. Therefore, there is a critical need for novel research in multiple simultaneous threats detection which provides improved precision and minimises the number of false positives and false negatives in live distributed system environments [14] [55].

1.2 Existing Solutions

Intrusion detection and monitoring systems are the most common tools used by security professionals. Most distributed networks, like in the UNIX environment, have deployed these monitoring systems in addition to security provided by firewalls. Pattern matching, signature-based and single-threat detection intrusion detection systems could only help in identifying and preventing known attacks. They possess a number of limitations against novel techniques and threats posed by insightful intruders [40][70].

As an example, the existing intrusion detection systems in UNIX were tested on low-speed networks of up to 100Mbps and are of limited use for high-volume network traffic as they do not support switched and encrypted networks. Due to recent advances in technologies, researchers can now have the opportunity to use Gigabit ISS/Networks in their experiments rather than the low-speed networks [7].

The following types of intrusion detection systems were identified during literature review:

(a) **Host-based intrusion detection systems:** These systems have many limitations as they rely only on a particular host and do not represent the entire environment. Furthermore, they are only used to detect and maintain the particular system on which the IDS resides.

(b) **Hybrid intrusion detection systems**: These systems are used to send alarm notifications to security professionals from both the network and the host during attacks.

(c) **Network node intrusion detection systems**: These systems have almost the same function as host-based intrusion detection systems. Network node intrusion detection systems only detect threats if they pass through the particular node where the IDS is installed.

The basic approaches used by the aforementioned intrusion detection systems in detecting security events include: pattern matching (anomaly detection), pattern templates, network data analysis, statistical inconsistency detection, state-based detection, covariance matrix-based detection (deviation from normality), and rule-based expert systems [52] [43] [21].

Different researchers have used various mathematical/statistical techniques in data fusion such as the Kalman filter, operational research, signal processing, pattern recognition, Bayesian theorem, Dempster-Shafer Evidential Theory and its extension, etc.[6] [44] [45].

Intrusion detection systems deal with sensors and use TCP/IP and its related protocols to detect network and internet packets. Different researchers use different sensor fusion models and their own strategies to detect threats. The most common statistical and mathematical techniques are Bayesian, Dempster-Shafer, and Markov chains. In most intrusion detection research, each investigation requires its own methods of data fusion, and they are targeted to

detect single attacks and threats while attempting to achieve better results by improving precision [20][71].

Each of the above techniques has its own limitations, which has been the subject of much research in the past decade. As a result, intrusion detection systems are now obtaining better results in monitoring, detecting and preventing attacks and reducing the number of false alarms.

1.3 The Importance of Multiple Simultaneous Threats Detection

Most commonly, systems are designed to detect threats in the computing industry and provide security in general, irrespective of whether they are UNIX systems, Wintel systems or any other environments. All intrusion detection systems fuse system, network and processor data obtained from their sensor models. These data are obtained from the Internet, hyper channels, farm, cyber or individual hosts. The sensor models may exist in a distributed or a client server environment [56] [61].

In large and complex computing businesses, it is very difficult to identify and evaluate threats. Existing intrusion detection systems are not able to trace and block immediate attacks. For example, hackers may attack a system through the TCP or launch a related attack from a different location. Attacks could also originate from separate switches or subnets by keep changing the IP addresses. Monitoring, evaluation, identification and tracking of such attacks require multiple simultaneous threats detection data fusion models. In conducting this thesis, not

much research in this area was found in the open literature. The few exceptions involve primarily the military networks [23] [69].

Theories of estimation can be considered the most powerful tool for fusing data and deducing cyber attack rates, targets, origins, and related information. Data fusion is done by using processor-intensive mathematical and statistical methods. Bayesian Theory, Dempster-Shafer Theory, Optimisation, Least square estimation and Sequential estimation are frequently used data fusion methods [27].

Threat identification and known pattern recognition has been a challenging task during the development of intrusion detection systems. It is quite often done at the time of raw data collection. Researchers have commonly used templates for pattern identification [66]. Intrusion detection systems largely use simple templates. Tracking and estimation of threats based on fusion models requires cluster analysis, adaptive neural networks, and rule based systems. However, in distributed systems like the UNIX environment, mathematical and statistical models have mainly been used [49].

Mathematical and statistical methods like Bayesian Theory, Dempster-Shafer and Extended Dempster-Shafer Theory, Theory of inference, Heuristic methods, and Parametric and Non-parametric approaches are the most widely used methods in data fusion models. These methods are also applicable in the decision making step of the data fusion process in distributed systems in order to identify multiple simultaneous threats. However, in developing advanced

intrusion detection systems, domain understanding of network application layers of the Internet should be incorporated [10].

Intrusion detection systems make inferences based on knowledge of the processes of multisensor data fusion – one of the most complex and challenging areas in the field. Different researchers have used different methods in order to achieve precision and accuracy in their results. One particular challenge is the difficulty in tracing the origin of attacks, and to determine the expected data loss or theft of the business information [47]. The problem becomes more complex when dealing with multiple threats, particularly in distributed systems on the Internet. The emergence of simultaneous security threats has underpinned the computing industry's continual research and development into cyberspace intrusion detection systems. It becomes an inevitable challenge for computer professionals to develop improved intrusion detection systems that can provide optimised security. Multiple simultaneous threats detection models, based on multisensor data fusion, are one of the key requirements in this development process [56] [5].

1.4 Approach and Methodology

Researchers have used both Bayesian and Dempster-Shafer theorems in multisensor data fusion research. The aim is to maximise the precision of inference. However, most research focuses only on single attack detection [61] [21], while no research has previously looked into the use of set cover theory to bring increased accuracy and reliability into their proposed models. Set cover

theory is not a separate fusion process or technique. It is studied and applied in this research to prioritise and schedule different rule sets based on certain signatures or criteria during the fusion process [17]. Therefore, the use of a hybrid model integrating set covering with data fusion will be an original contribution that has not been reported before.

Due to a lack of relevant prior research and literature on multiple simultaneous threats detection, it is not easy to compare intrusion detection systems, models or algorithms, particularly for distributed systems like the UNIX environment. Only a small set of papers have reported theories or algorithms that are relevant to the research on multiple threats detection. Furthermore, at the time of this research, multiple simultaneous threats detection models do not exist for the UNIX environment, in particular.

If one focuses on the UNIX environment, the existing multisensor data fusion models found in most intrusion detection systems use only three statistical/mathematical models including the parametric and non-parametric probability model, the Bayesian model, and the Dempster-Shafer and Extended Dempster-Shafer models. Almost all of these models have been used to detect single attacks such as denial of service, email bombs, or buffer overflow. These models have many limitations, as described in [59] [43] [9]. Among these models, those that address issues that are similar to this research include the Intrusion Detection System with SNORT, the Signal Detection System Based on Dempster-Shafer Theory, and the Comparison of Fuzzy Detection with Dempster-Shafer and Bayesian reasoning in multisensor data fusion.

In this thesis, a novel multiple simultaneous threats detection model based on a hybridization of Bayesian and Dempster-Shafer Theory of inferences, along with set cover theory, is proposed. This hybrid data fusion model provides significant increase in precision of detection. It also provides new knowledge into future research of multiple simultaneous threats detection in distributed systems, and in particular on the UNIX environment.

This thesis has involved a review of the relevant theories and literature that have been studied and applied in multisensory data fusion. The findings will be presented and analysed in the subsequent chapters of this thesis.

Furthermore, in this thesis software modules written in Perl and UNIX shell scripts are used to perform two main functions: 1) identifying the types of threats or attacks by capturing and analysing multisensor data collected from systems and network layers in a UNIX environment, and 2) monitoring and evaluating these results based on existing signatures and conveying them to system administrators or operators. In order to simulate these two functions, a private experimental network set up by using a combination of UNIX and Windows-based web applications as well as an Oracle database running business portal applications is used. Simulated data are collected in a distributed network environment from hosts as well as from the network. They are then analysed by using the proposed hybrid Bayesian and Dempster-Shafer model developed in this thesis. Additionally, the set covering module is used to populate threats data to the proper rule subsets.

The proposed data fusion model is able to identify more than one threat at a time. In addition, during experimentation, the number of simultaneous threats that can be detected by either of Bayesian-based model, Dempster-Shafer based model or the hybrid model (i.e. incorporating both theories and set covering) is compared. The set of empirical experiments was conducted on high speed networks since Gigabit Ethernet switches and routers have fast becoming affordable and available for private experimental environments.

Multisensor data originated from distributed network systems and their critical applications like Java, database and web services are collected and analysed using mathematical or statistical methods like Bayesian and Dempster-Shafer theories. These data constitute the knowledge base for analysing threats collected from the set of distributed detection sensors. The complete model, based on each individual system in the distributed environment, provides a precise and more reliable model to detect multiple simultaneous threats from diverse locations.

The proposed model and its optimisation can only be achieved using a combination of mathematical and statistical techniques [27] [42], augmented by the set covering theory. Set covering rules are rules of combination and scheduling. The latest approaches to applying these models are referenced in this research [3] [4] [62].

This research attempts to first transform the intrusion detection problem into a set covering problem. However, optimisation is not guaranteed. Efforts have been made to centralise the data fusion process using machine learning methods [8].

Finally, in order to ascertain which model achieves the highest precision in multiple simultaneous threats detection, Bayesian, Dempster-Shafer, Extended Dempster-Shafer, and Generalized Evidential Processing (GEP) are systematically compared in this thesis.

1.5 Significance and Innovation

The proposed multiple simultaneous threats detection model and associated setup are innovative yet different in ways from existing research. As the thesis presents, the focus is on multiple simultaneous threats instead of single threat detection in distributed systems like the UNIX environment. The underlying data fusion model makes use of a hybridization of Bayesian and Dempster-Shafer models, along with set cover based filtering for removing improbable threats prior to fusion.

The mathematical and statistical tools (Bayesian theory, Dempster-Shafer Theory, Generalized Evidential Processing, and set covering) work together to minimise response times for threats detection. The outcomes, when applied to the UNIX environment where many critical and high priority applications and databases of the IT industry are built upon, significantly improve UNIX security measures in practice. This is manifested in the lowering of false alarms from systems files, applications and databases that can cost companies involved millions of dollars. For example, system administrators set up rules for man-in-the-middle attacks on Oracle or Java in a UNIX environment. These rules could incur many false positives. However, if there is not an Oracle or Java application running in the actual environment, it could lead to excessive processing costs for

the business without deriving any benefit. Similarly, false negatives might confuse security professionals [12] [56].

Security professionals are always interested in detecting any alerts that might happen. However, genuine alerts should have the highest priority. This problem has been addressed by using evaluation monitoring and intrusion detection in security management systems. These intrusion detection systems, if deployed, should be demonstrated with high precision before they are applied in real businesses. Multisensor data fusion for multiple simultaneous threats detection, is therefore critical when setting up alert priorities [35].

In summary, security engineers in businesses are always looking out for extra security measures to protect their critical business data. Comprehensive security knowledge and good decision support strategies are required to develop security management systems for critical business environments. It is obvious that if proper security is not provided it is not possible for a business to keep their data secure for its clients and business partners. It is also impossible to maintain reliability, integrity and the trust of businesses in the industry. Advanced security methods, like the research behind this thesis, fill an important need in providing high security levels demanded in the computing industry [61]. In order to define usable standards for auditing the security of distributed systems like the UNIX environment, new theoretical results on multisensor data fusion modelling for multiple simultaneous threats detection are highly essential.

1.6 Contributions of this Thesis

This thesis proposes a novel data fusion model that is different from those used in existing intrusion detection methods. This data fusion model combines multiple simultaneous threats detection with multisensor data fusion techniques. It makes use of a hybrid model of Bayesian and Dempster-Shafer theories, along with set cover theory for filtering potential false positive and negatives. At the time of writing, no previous research has proposed a hybrid model, augmented with set cover theory, in distributed systems like the UNIX environment. Therefore, this thesis research can be considered a first step in making use of a combination of rule sets obtained through multisensor data fusion in distributed systems environment.

In addition, earlier works on threat detection mostly reported experiments conducted on 100Mbps networks. Therefore, these results are not very likely to be applicable in ultra high speed networks. Nowadays, it is affordable to deploy gigabit ethernet components in both private and public environments. In this research, the results are applicable to high speed networks.

In this thesis, the data fusion model proposed is a multiple simultaneous threats detection model based on a hybridization of Bayesian and Dempster-Shafer theories of inference, along with the set cover theory. This hybrid data fusion model increases the precision of detection while providing new knowledge for further research in intrusion detection in distributed systems like the UNIX environment. Successful implementation of the hybrid model is expected to reduce both financial and physical losses to various industries that require the

deployment of multiple servers in a distributed systems setting. In particular, web-based businesses and future IDS researchers can benefit from the research behind this model.

The proposed data fusion model is able to identify more than one threat at a time. In the experiments, how effective simultaneous threats can be dealt with by using Bayesian theory, Dempster-Shafer theory, and the hybrid model in conjunction with filtering based on set cover theory will be assessed. Another contribution of this research is that the results are applicable to high speed networks including the Internet /Intranet as well as both private and hyper-channel networks. The maximum speed attained during experimentation was one Gigabit per second.

1.7 Organisation of Remaining Chapters

The rest of this thesis details the research and development of the proposed multiple simultaneous threats detection system. Chapter 2 reviews the relevant literature. Chapter 3 gives an overview of the multiple simultaneous threats detection system, along with its architectural diagrams. Chapter 4 provides details of the data fusion models that were developed and applied in the empirical experimentation. Chapter 5 discusses the experimental setup, results, evaluation and comparison with related works. Chapter 6 gives the conclusion as well as identifying areas for future research.

2. Literature Review

In this chapter, literature on general concepts, issues and problems of intrusion detection are reviewed. In particular, current reports, progress and status regarding multiple intrusion detection systems used by security professionals, with emphasis on the UNIX environment, are examined. In addition, previous methods and approaches to security threats detection, especially on multisensor data fusion models and mathematical/statistical inference techniques, will also be discussed.

2.1 Problems in Intrusion Detection

An intelligent decision model for threat detection in any IDS using a multisensor data fusion approach often performs monitoring, evaluation and analysis of real time data. The intelligent decision model can be seen as the signature knowledge base of an inference engine [28] [41]. As a result of inferences derived from the decision model, the intrusion detection system sends an alarm to security professionals, system administrators or system engineers, so they can respond immediately.

Multisensor data fusion technology has been widely used in military surveillance, planning, commercial applications, robotics, sensing, and medical diagnosis [9]. A multisensor data fusion system is mainly a physical component of an intrusion detection system. It gathers data from distributed systems and process them

based on an intelligent rule set to deduce information about events such as type, intensity and location of compromised applications and hosts. Multisensor data fusion systems additionally analyse and track the characteristics of the attacks and their possible sources. They are capable of providing a greater degree of reliability in detecting threats when they could use a more accurate decision or inference model.

Multisensor data fusion is a relatively a new area in threat detection and many scientists are working on it in different fields to increase the precision of their data simulations. Multisensor data fusion techniques have been used very little in UNIX systems due to the complexity of these environments. In the near future it may not be easy to standardise multisensor data fusion in UNIX environments [10].

Theoretical knowledge of multisensor data fusion is limited. The required knowledge is diverse because of the large number of different fields involved. The inference engines used to detect threats are also diverse. This area of knowledge is very dynamic and is always changing. Due to the complexity of cyberspace environments and their widespread accessibility, it has become more challenging than ever to determine the origins, rates and level of damage of multiple simultaneous attacks. Most critical databases and business applications run on UNIX networks, and so multiple simultaneous threat detection requires integration and development of areas like statistics, mathematics, artificial intelligence, pattern recognition, and cognitive and decision theory [5] [27].

There is a pressing need to apply multiple threats detection systems in

cyberspace. Data mining is one option which can assist in developing some useful signatures for intrusion detection systems. Data of legacy systems stored in data warehouses could also be analysed. Therefore, a lot of research works are expected to develop reliable intrusion detection systems using an efficient multisensor data fusion model. It is hoped that the review given in this section might encourage researchers and security professionals working in distributed systems like the UNIX environment to carry out research to increase their understanding of the current status and challenges of advanced multiple simultaneous threats detection in the cyberspace [49] [33].

2.2 Multisensor Data Fusion

The most commonly used inference techniques are Bayesian inference, Dempster-Shafer theorem, fuzzy rules, parametric and non-parametric approaches, and the Kalman filter [21][9][4]. Some researchers have also used other approaches like the Chapman-Kalmogorov prediction integral along with Bayesian and Dempster-Shafer models [27] [1].

Inferences are made about threats, groups and their locations using distributed sensing. Multisensor data fusion is used to combine data from multiple sensors on the same or separate networks. Cognitive rule sets are applied in these data fusion systems in exactly the same way as human brains process and respond to data from sensory organs [8].

Statistical parametric/non-parametric, mathematical and heuristic techniques

such as the basic theory of inference, Bayesian theory of inference, Dempster-Shafer and Extended Dempster-Shafer theories, artificial intelligence and operation research, are the basic requirements for any data fusion model in technical systems [64].

In the vast majority of cyberspace intrusion detection systems, data fusion models obtain their input from sensor data created by systems commands and priority data accumulated in already established databases. For example, system data may come from sniffer packets, system log files, SNMP traces, system messages and other related activities collected from various components of a complex distributed environment. It can be expected that after processing the inputs, any intrusion detection system will produce findings about the attacker's identity, the type and rate of threats, the location of the attack, and an evaluation of the severity of the damage to the environment [59] [5].

It is well known that intrusion detection systems cannot detect all attacks launched by skilled hackers. The Langley Cyber Attack is one example. Security professionals were unable to detect this email bomb attack until critical servers crashed. It has also been discovered that hackers usually devote a significant amount of time to understanding the rule set of any intrusion detection system and to finding loopholes in the application or database environments before they design an attack to breach the computer security. In other words, hackers take advantage of loopholes in the rule set within the intrusion detection system. At the same time, false positive and false negative alarms are one of the biggest problems in distributed systems like the UNIX environment. Quite often, system

administrators and security professionals sound the alarm for a possible threat, and this usually requires a lot of investigation and analysis of the database, application and system data. This exhausts both time and significant financial resources. If such alarms are considered normal and are not given any attention, it becomes easy for hackers to filter their attacks through such weaknesses in the design [52] [57].

Existing intrusion detection systems designed to combat advanced attacks and provide remedies are not adequate. The complexity of most UNIX based systems, for example, is very high and sequential monitoring and assessment of network traffic from all possible heterogeneous sources would be required to detect, verify, block and assess high-level cyber attacks. Internet protocols are evolving and could assist in detection in cyberspace environments, but TCP/IP security still remains a critical security issue for Information Technology groups [62].

Multiple intrusion detection models are also not yet capable of auto-tracking, identifying and remedying all suspected threats. For example, many of these models struggle to cope if hackers attack from various geographical locations or initiate attacks from one network and continue on another one, while continuously changing the IP addresses of attack packets. In order to develop intrusion detection systems which can cope with such situations, new technical solutions are needed. That is why there is currently a lot of research in the field of multiple intrusion detection systems, especially for the UNIX environments which execute critical business data for large companies. In summary, there is a

pressing need to develop multiple threat detection models in distribute systems, like UNIX [29].

Hall [34] discusses the application of multisensor data fusion processes using mathematical techniques. The application of multisensor data fusion in intrusion detection systems is a complex endeavour at the lowest levels of data association. Hall describes different approaches to data association in intrusion detection systems in his book.

Hall used parametric data to detect threats related data from the network. He argues that mathematical and statistical theories of inference are required to spot possible attacks, rates of attack, targets and other related internet or network parameters. In order to achieve more precision and reduce false alarms, threat detection involves data association, optimisation and also least square or sequential calculation. It is therefore a very computationally intensive process and requires complex models and analysis [53].

Another researcher further emphasizes the complexity of the processes involved in identification and pattern recognition of the fusion model because of technical problems. Pattern recognition is derived from special characteristics of raw data. Different types of templates, comparisons and evaluations are used in such processes. Special rule set-based templates are used in currently deployed intrusion detection systems. Bayesian and Dempster-Shafer theories of inference have been used in multiple threats detection. Smith pointed out that rule-based knowledge can be refined using set covering along with Dempster-Shafer Theory [26] [60].

Decision making in identifying threats using a fusion process requires very intensive analysis and inference. The most commonly used processes in intrusion detection systems are: heuristic and mathematical methods like classical inference, Bayesian theory of inference, Dempster-Shafer theory and its extension, and generalised evidence processing (GEP). As has been mentioned, the application of these data fusion processes within intrusion detection systems is a challenging task, requiring very good knowledge and skills of cyberspace trends and awareness of the cyberspace environment.

It has always been an enormous task to design a reliable multisensor data fusion model for any intrusion detection system in cyberspace. Multisensor data fusion models require fusing of data from distributed sensors both from within and outside of its network environments. The distributed sensors must be located in all software and hardware infrastructure, including all services, processes and network events that are vulnerable to threat detection.

One of the advanced steps in intrusion detection system development is the extension of the single threat detection models of Braun [9] and Dong and Deborah [28] into the Bayesian and Dempster-Shafer multiple threats detection models used by them in their multisensor data fusion model. To develop reliable threat detection models for generic intrusion detection in distributed systems environments, set covering is required for data refinement, data association rule set knowledge, data archiving, data cleansing, and data primary correlations and corrections.

2.3 The Importance of Simultaneous Threats Detection

Researchers have used both the Bayesian and Dempster-Shafer models for intrusion detection. The Dempster-Shafer model is an advanced version of the Bayesian model. Benefits of applying these models include more precision inference, etc. However, they are used solely for single attack scenarios [50] [41], while there has not been reported work on their integration with set cover theory to derive more accuracy and reliability on their inferences. Set covering is not a separate fusion step. It is used in this research for prioritising and scheduling different rule sets based on certain signatures or criteria during the fusion process. As such, the use of a hybrid model along with set covering is a novel contribution of this research.

In the review of literature mentioned earlier, different single threat detection models which use Bayesian and Dempster-Shafer theorems have been discussed. Extensions of these models to multiple threats detection are rare, but it is believed that their novel integration might bring about progress in research and analysis of multiple simultaneous threats detection.

In this research, another novelty will be the verification of models or algorithms for intrusion detection that work for the network protocol layer as to their usability in other protocols such as the file transfer protocols and secure shells as in the UNIX environment. Experiments aiming to associate, validate, and categorise data in groups using both single threat models and the proposed hybrid data fusion model on multiple simultaneous threats will be carried out. To accomplish this, data related to different parameters of the experimental multiple

simultaneous threats detection system developed in this research will be systematically compared using the selected statistical and mathematical techniques.

In the proposed multiple simultaneous threats detection system, there is a middle-tier component that simulates the application layer and web servers. However, this middle-tier component could use different protocols and service hosts. The application layer is often a place where multiple simultaneous threats would occur because the network application layer below often poses security problems. The Secure Shell (SSH) protocol of the network application layer can be infiltrated from remote SSH and can also be exploited using root logins [13]. FTP daemon has many security problems, however, the remote root login is the most critical one [15] [16].

There are many arguments against the use of web servers' cyberspace environments. Web servers are widely used as front-end applications, both for businesses and end users. Therefore, security of the web interface is more critical than the security of any other area of cyberspace, and threat detection in this area will have good returns in terms of threat detection and false positive rate problems. Web servers are the most commonly used cyber software and are more visible and less secure than other system components. They are mostly kept outside the demilitarized zone of private networks and therefore are good targets for hackers. Web-based infrastructure applications in distributed system environments are often being attacked. Microsoft's Internet Information Server, being a leader in terms of its percentage share in the market, has been attacked

many times by hackers [54]. Web Sphere, Apache and Web Logic of BEA had a good record, but they have also been attacked by shrewd attackers [24]. Apache and PHP have had their own security issues at their code level [13]. Due to the above problems, a web application will be posed as the middle-tier component in the experimental setup. By this, the usefulness of the proposed multisensor data fusion model for effectively detecting multiple simultaneous threats can be demonstrated.

Lastly, in the literature, single-threat detection models that have been applied to network application level existed [48] [11] [36]. However, the accuracy of their detection cannot be said to be sufficient. In addition, comparisons on multisensor and multiple threats detection in intrusion detection systems are few. Quality comparisons regarding multiple threats detection are even fewer, and there is none for simultaneous multiple threats detection in distributed system environment like UNIX. There is very few literature reported on the theory and the numerical results of multisensor data fusion techniques in intrusion detection systems, particular for the UNIX environment. Therefore, the present research in multiple simultaneous threats detection will fill a gap in current research of computer security. The outcomes of this research, when applied to distributed system environment like UNIX, can be expected to help in minimizing critical data losses as it is the most commonly used operating systems for critical business applications nowadays.

2.4 Related Work

There is no shortage of research work and literature on multisensor data fusion and intrusion detection in defence and other fields. However, if one investigates research about multiple simultaneous threats detection in distributed system environment, there is only a handful of research that has been done. Particularly, in the UNIX environment, multiple simultaneous threats detection research has been long overdue. This research will be of great benefit to UNIX computer security systems as hackers have become smart enough to break into mid-range operating systems like UNIX and its application security. Just a few years back, UNIX was considered a very secure operating system. All computing business based on credit cards, client profiles and other important financial transactions over the internet are now run on UNIX. Therefore, businesses need more security than ever before.

Dong and Deborah's [28] work on DARPA IDS evaluation data set proves that the combined fusion model of Dempster and the Extended Dempster-Shafer Theory in an intrusion detection system can improve its alert fusion algorithm. The use of this model has increased detection rates from 75% to 93.8% without getting many false positives.

Siaterlis and Maglaris [61] concluded in their combined research that a multiple data fusion model of an intrusion detection system is a valid method to find out the accuracy of threat detection and false positive rates in any intrusion detection system. They used a detection engine based on Bayesian and Dempster-Shafer theories. The key achievement of their combined Bayesian and Dempster-Shafer model is in providing a method to combine probability masses via independent

evidence variables. Their DS model was:

$$\sum_{i=0}^{n} Mi(T_i) = \sum_{i=0}^{n} \frac{P(\{T_i\})}{P(\{T_i\})+P(\{\neg T_i\})}$$

where M is probability mass function, T is the threat(s), $P(\{T_i\})$ is the probability of an i^{th} threat of the i^{th} intrusion detection system for a particular type of the threat, and *Mi* is the membership function. However, the authors admitted that their model has been unable to detect multiple simultaneous attacks.

Wu, Siegel and Rainer [33] discussed the relationship between Bayesian theory and Dempster-Shafer Theory in a data fusion model for an intrusion detection system as compared to a weighted probability method. They obtained promising results for research into using combined mathematical inference models.

Habib and Hefeeda [2] and Siaterlis and Maglaris [61] used classical Bayesian methods for their data fusion of intrusion detection systems, and found a significant improvement as compared to early research work. The capability for early threats detection of multiple attacks in a distributed network environment and the chances of multiple threats detection of a distributed denial of service (DoS) threat increased.

Diego Zamboni [26] used a signature-based detection model by looking for well defined patterns of attack that exploit weaknesses of the system. He used intrusion detection system infrastructure, implementation, testing and analysis of the multisensor data fusion to detect new attacks on the network. He also used anomalous behaviour patterns for anomaly detection. However, Diego does not

speak about any particular fusion model in his research.

V. Chatzigiannakis, A. Lenis , C. Siaterlis, M. Grammatikou, D. Kalogeras, S. Papavassiliou and V. Maglaris [67] proposed a data fusion algorithm for intrusion detection systems based on the application of principal component analysis of multisensor data for anomaly detection. They found that their fusion model is more effective than single metric analysis.

Vladimir and Oleg [68] worked on an intrusion detection system fusion model in a distributed environment and suggest that a model which combines decisions is better than a meta model for intrusion detection.

Kapil [48] researched a data fusion model for intrusion detection and analysis and argued that the development of novel solutions for intrusion detection architecture is required. He believed that the data fusion model should be based on rule set knowledge, expert systems, state models and string matches.

Hervaldo, Carvalho, Heinzelman, Murphy, Claudionor and Coelho [35] concluded that threats detection fusion architecture can be used in other similar environments and circumstances.

Hugh F and Durrant-Whyte [39] worked on decentralised sensing networks and described the mathematical methods underpinning a fusion model. They used the Kalman filter algorithm and other theoretical methods derived from the Bayesian theorem.

Terry Brugger [60] discussed an offline data fusion model in threat detection by augmenting existing real time sensors. His research proposed a new area of fusion

using data mining techniques. However, he did not produce any particular model in his research.

Although there is a large amount of work in data fusion for intrusion detection systems, there is very little work on multiple simultaneous threats detection in distributed systems like UNIX.

This literature review will conclude by summarising some important points on multiple threats detection in different fields. Only a few are in distributed systems, and none involves multiple simultaneous threats detection in UNIX environment in particular.

- Few papers were found on the comparison of intrusion detection systems models and algorithms, especially inference models based on Dempster-Shafer and Bayesian models. Literature about intrusion detection systems based on set covers rule set is very limited and none exists for UNIX environment in particular.

- Theoretical study of inference methods in data fusion models for multiple threats detection models is rare, and research in dealing with multiple simultaneous threats detection models incorporating set cover theory is not known. It seems that the theory of inference methods in data fusion models is still in its infancy. This is confirmed when multiple simultaneous threat detection models are reviewed.

- From the literature reviewed, it is found that multisensor data fusion in intrusion detection systems for the UNIX environment have been done

using only three models to date. They were based on parametric and non-parametric probability model, Bayesian theory of estimation, and Dempster-Shafer and Extended Dempster-Shafer theories. Almost all of the models are used to detect single IDS attacks like denial of service (DoS) and email bombs that have many limitations.

- Anomaly-based intrusion detection systems do not use a network application layer in their data fusion models.

- Development, testing and comparing data fusion models in multiple threats detection are rare, while research related to multisensor data fusion is even rarer.

- Attacks from hackers have costed billions of dollars of damage [30] [65]. Currently available intrusion detection systems do not fully detect multiple simultaneous threats, and hence there is a pressing need for developing advanced-level intrusion detection systems. For this research, the following contributions are expected:

    - A multiple simultaneous threats detection model targeting future intrusion detection systems will be proposed

    - A high-level model/architecture that can address multiple simultaneous attacks in distributed systems like UNIX will be developed

    - An advanced multisensor data fusion model will be introduced

- An experimental environment for comparing data fusion algorithms in multiple simultaneous threats detection scenario will be constructed and used in empirical experimentation to demonstrate the usefulness of the approach

- A comparison of data fusion models based on different mathematical inference techniques will be carried out

Through verifying the model proposed in this research using both simulated and public domain datasets, an effective multiple simultaneous threats detection system will be produced. In addition to distributed system environment like UNIX, the outcomes of this research will be likewise applicable to the Windows environment on the condition that they operate on similar OSI layers and protocols.

2.5 Results and Benefits

The research of this thesis will facilitate the development of multiple simultaneous threats detection system to counter multiple intrusion attacks in distributed systems like the UNIX environments. It also provides an objective comparison on which inference models would perform the best. These models include Bayesian, Dempster-Shafer, and their hybridisation combined with set cover based filtering. Empirical experiments on both simulated and public datasets confirmed the benefits of applying the proposed multiple simultaneous threats detection model in a distributed systems environment like UNIX.

3. Multiple Simultaneous Threats Detection System

The main aim of this research is to develop a hybrid data fusion model, comprised of set cover, Dempster-Shafer, Extended Dempster-Shafer and Generalised Evidential Processing (GEP), which can identify exact threat(s) with a high degree of precision. An analysis of the origins and directions of the threats themselves is beyond the scope of this research.

In the simulation environment, which is conceptually the same as a client and server environment, a multiple simultaneous threats detection system is set up on different computer nodes across the distributed network. Computer nodes are comprised of multiple operating systems and are located on different networks, predominantly running the UNIX environment. Additionally, there are some that run the Windows environment. Each computer node has a different intrusion detection system that filters all the network data and collects threat-related information, which are then transferred to the computer node that hosts the multiple simultaneous threats detection system for further analysis.

Under our assumption, computer nodes across different sub-networks receive different threats. In the experiments, four types of threats, including denial of service, man-in-the-middle, buffer overflow and Trojan, are initiated from the computer nodes (refer to Figure 3.1).

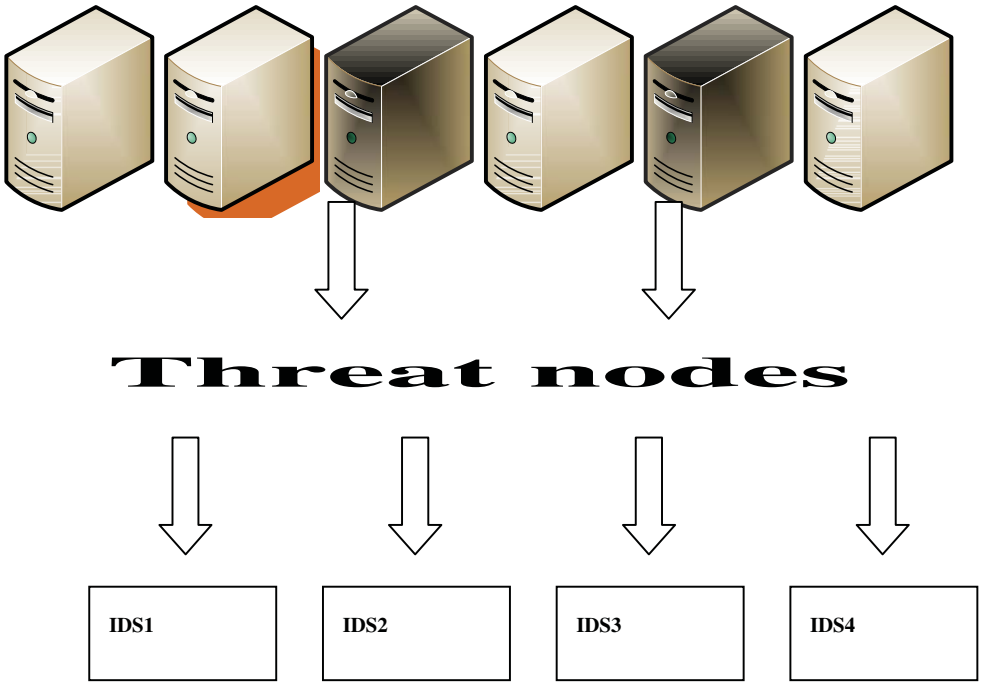

Fig 3.1: Threat nodes used in experiments of the multiple simultaneous threats detection system

The architecture of the multiple simultaneous threats detection system can be seen as a hybridization of the US Joint Directors of Laboratories (JDL) fusion architecture, the Waterfall fusion model, and the Omnibus model architecture [33] [43].

3.1 Types of Threats

In order to generate four types of the following attacks, there are many software tools available in the market, however, I selected the following based of their efficiency and cost effectiveness.

- DoS – Denial of Service

- MITMA – man-in-the-middle attack or bucket-brigade attack or Janus attack
- Buffer overflow or buffer overrun
- Trojan Horses

Each intrusion detection system collects network data, and filters them using an algorithm based on the cover set theory. The filtered data are then moved to the next level of data fusion within the multiple simultaneous threats detection system. Data might contain one, two, three, or four of the above threats, and their combination. There can also be false alarms in the data. The multiple simultaneous threats detection system processes the data through several statistical and mathematical techniques implemented, and makes decisions about the threats.

Each of the client nodes of the multiple simultaneous threats detection system uses the set covering approach to filter the data into small subsets, followed by scheduling these subsets of data for onward statistical and mathematical data fusion. Another benefit of the middle-tier set covering layer is that the subsets combined will contain all the potential threats anticipated.

3.2 Threats Generation Utilities

The following set of utilities has been used to generate the threats data.

- *ICMP* and *ping floods* were used to generate “Denial-of-service attack” threats
- *Dsniff*, *Cain*, *wsniff*, and *airjack* were used to generate “man-in-the-middle attack” threats
- *elf-replication* was used to generate “Trojan horse” threats
- *Morris* and *conflicker worms* were used to generate “Buffer overflow” threats

3.3 Why only Four Threats

Naturally, there could be any number of threats happening simultaneously. However, in the experimentation, the number of simultaneous threats is limited to no more than four, in order to reduce the overhead in computing the threats’ probability masses and weights. As remarked in earlier sections, the processing cost for detecting more than four simultaneous threats will become too high for most practical purposes in small businesses. On the other hand, model based on the Generalised Evidential Processing (GEP) approach does not impose any restrictions on the number of simultaneous threats.

3.4 Intrusions Detection Systems as Independent Observers

In order to monitor all data packets that come through the testing environment, a switch on the network was installed to monitor and replicate all the packets.

Network data that pass through the OSI layer 2 (i.e. data link protocols, like Ethernet for local area network, ppp and others) and layer 3 (network layer, like source-to-destination packet delivery) are also gathered. The following software based on their market reputation and cost effectiveness are used for data collection:

- MARS
- Sniffers
- Snoop
- Wireshark

4. Data Fusion Process Models

4.1 Architecture of the Multisensor Data Fusion Process Model

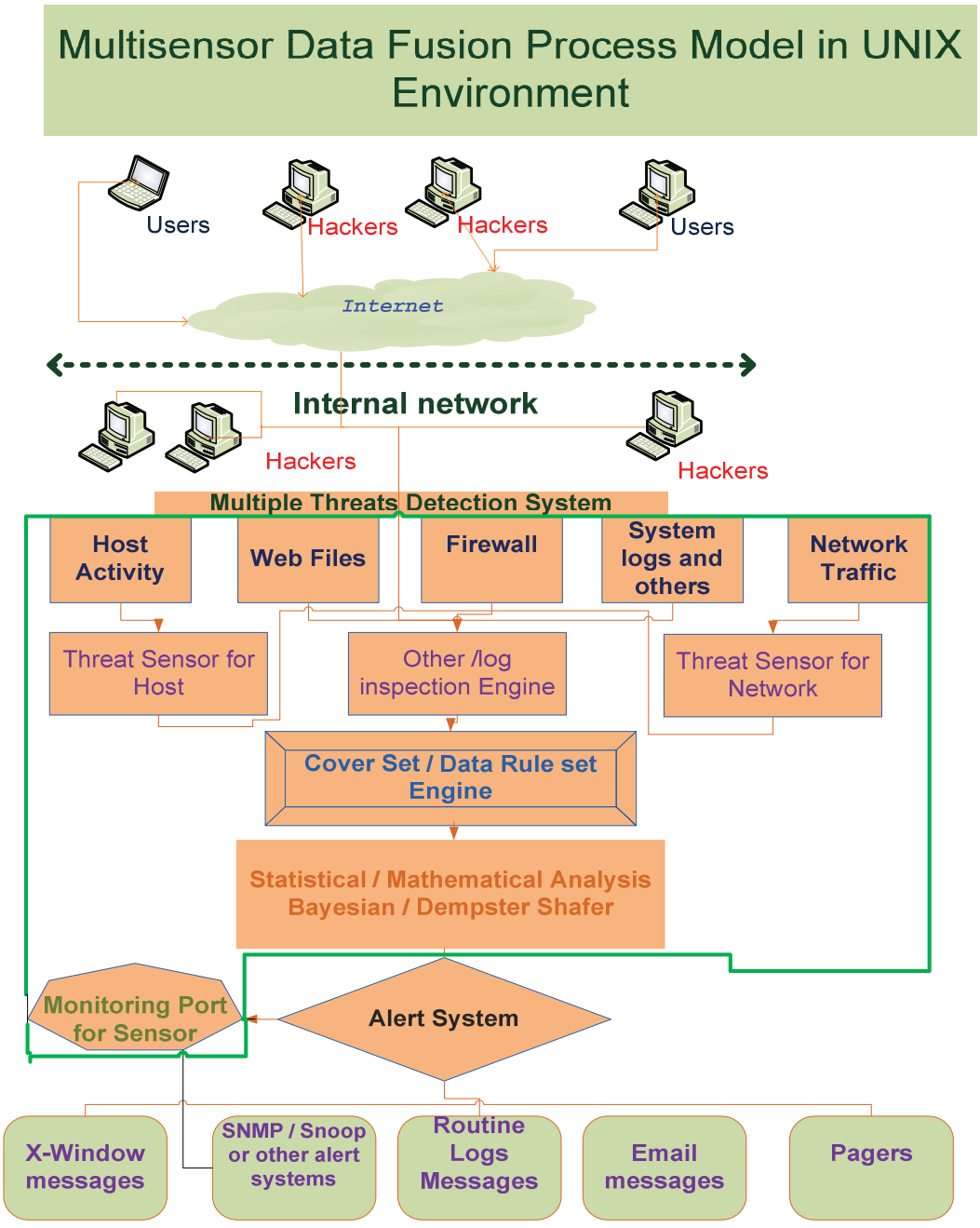

Fig 4.1: Architecture of the Multiple Simultaneous Threats Detection System

(The references and further details of these data fusion models are given in section 4.2 of this chapter)

4.2 Brief Description of the Multisensor Data Fusion Process Model

The main components of the models are Set cover, Dempster-Shafer, Extended Dempster-Shafer and Generalised Evidential Processing theories underpin the multisensor data fusion model proposed in this thesis. Before presenting the proposed data fusion model, each of these four theories and their associated data fusion model will be described.

4.3 Set Cover Theory

Set covering helps the filtering of threats data into smaller subsets. At the same time, it assists in an optimal choice of computer nodes to process these data subsets. Details of the underlying process are discussed in the following sub-sections.

4.3.1 Overview of Set Cover

Set covering is a branch of mathematics that deals with sets, subsets and their interaction sets. Simple facts regarding sets and their subsets are used in the cover sets of a multiple simultaneous threats detection system [37] [38]. In this thesis, the set union operation is applied in the generation of subsets.

Importantly, set covering is used as the middle tier in the multiple simultaneous threats detection model to reduce the number of threats and schedule them for analysis in the next level of processing. For example, researchers at IBM found 500 viruses/threats that had 9,000 data sub-strings of 20 bytes or longer. By using set covering, a set of 180 sub-strings that is sufficient to cover the

existence of 5,000 known viruses was obtained [38]. An additional benefit of set covering is its efficiency and short processing time.

Data that contain threats like DoS, Buffer Overflow, Trojan horse, Man-in-the Middle, etc, constitute the universal set of the testing environment. Each data packet is assigned a number and placed in its relevant set based on the type of attacks it might belong to. Some of the data packets share a similar number, which are often neighbouring elements that are close to each other in their respective subsets.

In the multiple simultaneous threats detection system of this study, the total number of elements is 2,274 (explained in sub-section 5.1.2) denoted by:

$$\bigcup = \sum_{i=1}^{n} u_i \qquad (1)$$

Where $\bigcup$ denotes the universal set, and $\sum_{i=1}^{n} u_i$ is the sum of all the element (threats) in the universal set.

In the experiment, the types of threats are represented by the corresponding subsets

$S_1, S_2, S_3, ..., S_j \subseteq \bigcup$ while the cost of each set is $C_1, C_2, C_3, \ldots\ldots, C_k$

In this thesis, threat(s) are present in different data sub-strings collected from any of the four different intrusion detection systems in the experimental setup. The objective is to find the group of subsets that together encompasses the minimum number of sub-strings representing threats, so that each set has all the relevant

sub-strings. A cover set obtained by using a greedy algorithm also provides the minimum cost represented by the quantity, Q, defined as follow:

$$Q = \sum_{i=1}^{m} C_i \qquad (2)$$

where $\sum_{i=1}^{m} C_i$ is the combined cost of identifying the set of threats through set covering taken by the computers in the middle tier layer.

4.3.2 Greedy Algorithm

In this research, Greedy algorithm provides optimistic approach in finding the minimum cost of a node in its selection for the threat detection process. Greedy algorithm help to determine the most cost effective computer node, assign a value to all the nodes, decide if the node should be used or not in the experiment based on its cost value and assist in the progress of the threat detection for further data fusion using Dempster Shafer, Extended Dempster Shafer and Generalized Evidential Processing (GEP) [3][37].

The cost effectiveness of selecting a computer node based on a greedy algorithm is denoted by

$$\beta = \frac{C(Q)}{Q - Z} \qquad (3)$$

where $C(Q)$ is the initial cost for selecting the nodes by each intrusion detection system, and Z is the set with minimum elements and Q is the minimum cost of selecting the new node [36] [58].

4.3.3 The Generation of Threats for Set Cover

In order to collect the four types of threats, four individual intrusion detection systems in the experimental environment were used to simulate the collection of 2,274 malicious sub-strings of 15 bytes or longer. The threat data is a combination of all four types of generated threats. Set covering using the parameter $K = 4$ enables the creation of 4 pairwise disjoint subsets, which together results in a total of 295 threats. A Perl script was used to separate the set of pairwise disjoint strings into four subsets, that is a total 295 substrings after set covering was applied. In Table 4.2, one can observe that there is a considerable reduction in the number of threats in each of four original subset of threats collected by the four intrusion detection systems:

Threats Data of Intrusion Detection Systems

IDS	Before Set Cover	After Set Cover
Wireshark	128	122
Sniffers	439	82
Snoop	646	32
MARS	1061	59
Total Threats Alerts	2274	295

Table 4.1: Set Cover reduces the sizes of the subsets of threat data

The computer that runs the set covering algorithm then forwards the set of filtered data (i.e. the 295 sub-strings) on to the next level of the multiple simultaneous threats detection system, which executes the multisensor data

fusion algorithm. In this level of the architecture, the multiple simultaneous threats detection system analyses the filtered threat data by using a multisensor data fusion model that is a hybridization of Bayesian theory, Dempster-Shafer and Extended Dempster-Shafer theories, as well as Generalized Evidential Processing (GEP) theory.

4.3.4 Empirical Tests

The initial processing cost (in units of dollars) for selecting nodes by each intrusion detection system is:

$C_{(A)}$ = 8, cost of node selection by ID1 (4)

$C_{(B)}$ = 5, cost of node selection by ID2 (5)

$C_{(C)}$ =12, cost of node selection by ID3 (6)

$C_{(D)}$ = 8, cost of node selection by ID4 (7)

The cost of node selection (excluding hardware cost) varies for different Intrusion Detection Systems depending on the estimated cost involved in the system percentage usage of the CPU, memory and storage required for the IDSs. In this experiment as the experimental environment was small, therefore, the estimated cost is very low as compared to the enterprise IDS set up for any enterprise level organization.

These values are determined empirically, based on the computation required by each intrusion detection system involved in the experiment. The minimum

number of elements *Z* (sets of threats) and the minimum cost *Q* (in dollars) for each intrusion detection system were determined during the experiment, which are given as follow:

$$Z_{(A)} = 0,\ Z_{(B)} = 3,\ Z_{(C)} = 3,\ Z_{(D)} = 4 \qquad (8)$$

$$Q_{(A)} = 8,\ Q_{(B)} = 4,\ Q_{(C)} = 7,\ Q_{(D)} = 8 \qquad (9)$$

Cost for each intrusion detection system (i.e. A, B, C and D):

Minimum number of sets covered:

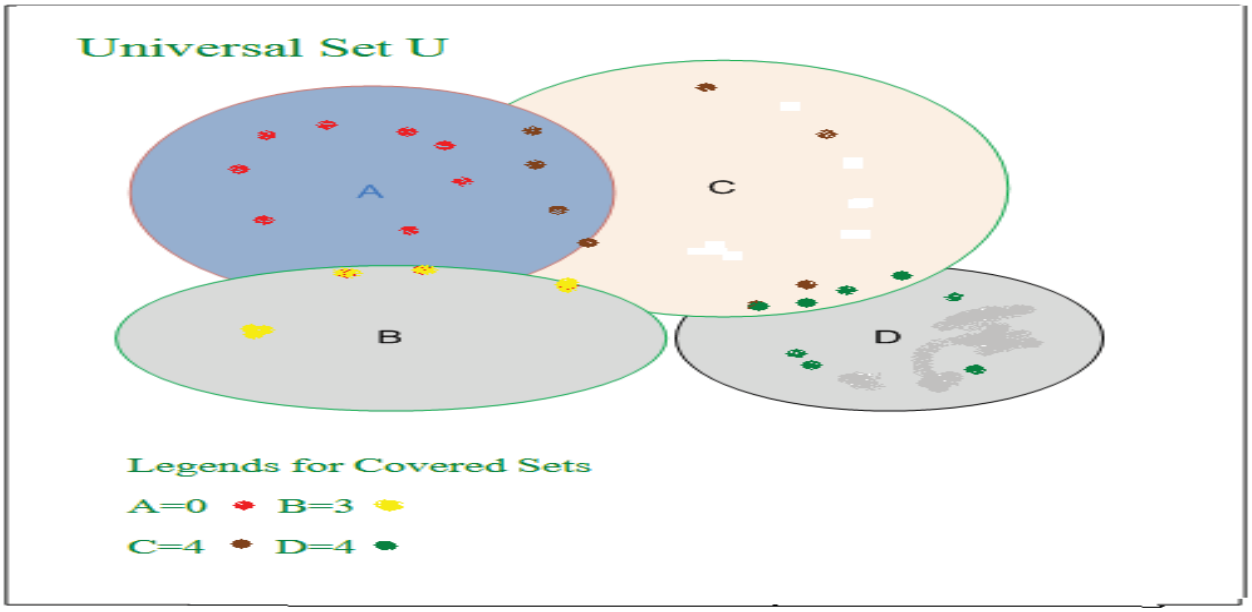

Fig 4.2: Z (Set Cover) and Q (sets with minimum cost) of the Universal Set

The optimal cost depending on the estimated cost involved in the system utilization i.e. CPU, memory and storage required for the IDSs, calculated as per cost effectiveness β of each intrusion detection system, using equation (3) are:

Selecting A: $\beta_A = \frac{C(A)}{Q - Z} = \frac{8}{8 - 0} = 1$ (10)

Selecting B: $\beta_B = \frac{C(B)}{Q - Z} = \frac{5}{4 - 3} = 5$ (11)

Selecting C: $\beta_C = \frac{C(C)}{Q - Z} = \frac{12}{7 - 3} = 3$ (12)

Selecting D: $\beta_D = \frac{C(D)}{Q - Z} = \frac{8}{8 - 4} = 2$ (13)

It is noteworthy that the *Z* values refer to the numbers of the sets, not elements of each set that will result in each of the intrusion detection systems (refer to Figure 4.2). The optimal cost, as per cost effectiveness β of the nodes, calculated using equation (3) would be A+D+C=8+8+12=28. The cost effectiveness of the nodes are referred to by A, D, C and B respectively [3].

4.3.5 Benefits of Set Cover as a Middle-tier Data Fusion Tool

1) Four types of threats were observed by the four intrusion detection systems, which generated 2,274 threat alerts. That meant there was an average of 569 overlapping and conflicting evidences for each threat detected by the four intrusion detection systems. The application of set cover theory reduced the number of alerts from

2,274 to 295. The application of set cover reduced the number of threats that the Multiple Simultaneous Threats Detection system had to deal with. In other words, the system had to deal with unambiguous types of evidence (less conflicting and less overlapping) instead of the significant amount of processing that would be needed in the absence of set cover processing.

2) Cost Effectiveness

As a result, only 4,425 (15 x 295) non-repetitive propositions had to be processed and tested by the Multiple Simultaneous Threats Detection system as compared to 773,160 (340 x 2,274) if repetitive propositions had been included. If one CPU can process 50 requests/propositions per second, the total time for non-repetitive propositions would therefore be 128.7 seconds = 2.15 minutes. Repetitive propositions would take 15,463 seconds = 4.5 hours. Moreover, repetitive propositions also require significantly more processing and memory resources, and therefore the Multiple Simultaneous Threats Detection system would require at least 16 CPU cores with 1.2 GHz under our assumptions. Due to the very high cost of IT resources, many businesses would be unable to afford such a high-cost on detecting intrusion attacks on their networks and systems.

3) The benefits of applying set covering in determining the cost associated with the selection of compute nodes for the Multiple

Simultaneous Threats Detection system are demonstrated according to our experiments.

4.4 The Dempster-Shafer Fusion Model

4.4.1 Overview of the Dempster-Shafer Theory

Dempster-Shafer theory of evidence helps to compute (quantify) a degree of belief (also called probability mass) by combining evidences from available resources. It is an extension of the Bayesian theory [9].

Since the introduction of this theory, researchers have been working on plausible rules for assigning and combining probability masses. In order to resolve conflicting evidences more effectively, different researchers have proposed different approaches to combine evidence from different sources. Overall, the Dempster-Shafer Theory is known for its ability to assign a probability to an event when it is difficult to calculate it accurately. The theory provides a higher degree of confidence to an event (also called probability mass) with the availability of increase evidences [1]. Unlike the Bayesian theory, the Dempster-Shafer theory is able to assign a value to quantify the uncertainty of an event.

Contrary to classical theories of inference such as the Bayesian theory, the Dempster-Shafer theory can make a basic probability assignment (bpa) to multiple co-occurring events. It has three major functions that provide a higher level of confidence when determining the output of an event. These functions are: basic probability assignment, belief function (Bel) and plausibility function

(Pl). The bpa is the a-priori evidence represented by *m*. The belief of a set is defined as the sum of all the bpa's of the subsets (B) of a set (A) $(B \cap A)$ [21].

4.4.2 Evidence to Proposition Assignments

Using the terminologies of Dempster-Shafer theory in explaining the experiments, the frame of discernment θ will be the set of elementary propositions or combinations of the hypotheses. Threats, denoted by T, can be overlapping or different to each other. In the set of *n* mutually exclusive and exhaustive set of hypotheses about the threat $T_1 \ldots .T_n$.

$$\theta = \{T_1, T_2 \ldots T_n\} \qquad (14)$$

If θ contains *n* hypotheses, the Boolean combination for this set will contain θ^n hypotheses.

Dempster-Shafer Theory is not used to calculate the probability of a hypothesis, but it is used in determining the probability of the degree of evidential support for a hypothesis. Unlike Bayesian theory and the classical theory of inference, the Dempster-Shafer Theory of inference determines the probability mass ($m(\theta)$) by assigning evidences to each proposition. Each intrusion detection system can assign evidences via probability masses to each of the four types of threats, for example, m1 will be the probability mass of threat1 (T1) and similarly, m2 will be the probability mass of threat2 (T2) , m3 will be the probability mass of threat3 (T3) and m4 will be the probability mass of threat4 (T4). The sum of the probability masses of all the propositions, including general propositions, will be

equal to 1. The probability masses, covered by the frame of discernment θ, satisfy the following conditions, θ (theta) is an ancient Greek letter, can be used both in lower and upper case. Here in this formula, θ represents total set of hypothesis.

$$m(\theta) \leq 1 \quad (15)$$

$$\sum_{i=1}^{n} m(\theta) = 1 \quad (16)$$

where $m(\theta)$ is the probability mass of any possible hypothesis. In the empirical experiments, there could be either a single or a combination of the four threats.

Proposition 1: T_1 , threat is a DoS

Proposition 2: T_2 , threat is a MITMA

Proposition 3: T_3 , threat is a Buffer Overflow

Proposition 4; T_4 , threat is a Trojan horse

The above four propositions are the elementary hypotheses. Using their Boolean combinations, the total number of propositions will be $2^4 = 16$

$$\theta = \{T_1, T_2 \ldots T_{16}\} \quad (17)$$

4.4.3 Threats as Propositions in Intrusion Detection

A proposition can be a single hypothesis or a combination of hypotheses. In the empirical experiments, four sensors and four types of threats were being used. Sensors could receive a single threat or any possible combinations of the four

threats. The total possible base propositions, calculated using the mathematical theory of combinatorics with and without repetitions are 340 and 15, respectively. Repeated threats are of no significance during testing, and if tested for would also increase the processing cost and time unnecessarily. Therefore, in the experimentation, only the 'without repetitions' propositions are assumed. This reduced the total number of threats to 295.

Only 4,425 (15 x 295) non-repetitive propositions will be processed and tested by the Multiple Simultaneous Threats Detection System as compared to 773,160 (340 x 2,274) if repetitive propositions were included. If one proposition took 1 nanosecond of operating time on a CPU core, one core can process 50 requests/propositions per second. The total time for testing for non-repetitive propositions would be 128.7 seconds or 2.15 minutes. Testing for repetitive propositions would take 15,463 seconds or 4.5 hours. Moreover, repetitive propositions also require significantly more processing and therefore would require at least 16 cores CPU with 1.2 GHz. That would not be acceptable to any business.

The general formula for selecting a permutation of the threats would be:

$$P(n, r) = \frac{n!}{r!(n - r)!} \quad (18)$$

Where n is the number of sensors (in the intrusion detection system), r is the number of threats to be selected ($0 \leq r \leq n$), where $n = 4$ in the experiments.

Case 1: when a single threat is detected by each sensor, the total number of hypotheses/propositions with and without repetitions would both be 4.

By putting values of n and r in the above formula without repetitions:-

$$P(n,r) = \frac{4!}{1!(4-1)!} = 4$$

Hypotheses/propositions with repetitions:-

$$n^r = 4^1 = 4$$

Case 2: when two threats are detected by each sensor, the total number of hypotheses/propositions with and without repetitions would be 6 and 16 respectively.

By putting values of n and r in the above formula without repetitions:-

$$P(n,r) = \frac{4!}{2!(4-2)!} = 6$$

Hypotheses/propositions with repetitions:-

$$n^r = 4^2 = 16$$

Case 3: when three threats are detected by each sensor, the total number of hypotheses/propositions with and without repetitions would be 64 and 4, respectively.

By putting values of n and r in the above formula without repetitions:-

$$P(n,r) = \frac{4!}{3!(4-3)!} = 4$$

Hypotheses/propositions with repetitions:-

$$n^r = 4^3 = 64$$

Case 4: when four threats are detected by each sensor, the total number of hypotheses/propositions with and without repetitions would be 256 and 1, respectively.

By putting values of n and r in the above formula without repetitions:-

$$P(n, r) = \frac{4!}{4!(4-4)!} = 1$$

Hypotheses/propositions with repetitions:-

$$n^r = 4^4 = 256$$

4.4.4 Limitations of the Dempster-Shafer Theory

Due to the high complexity of the probability masses and weights calculations, it was not feasible to cover all the 15 non-repetitive hypotheses. For example in case of two threats, the number of hypothesis as per DS will be $\theta^n - 1 = 4^2 - 1 = 15$, where θ is number of sensors and n is the number of threats. Therefore, only four elementary hypotheses as mentioned in Section 4.1 were covered. This will facilitate an adequate evaluation of the multiple threats detection system which caters for four types of threats

To understand better the complexity of the hypothesis calculation using Dempster Shafer theory of inference, also read the description given in section 5.2.3.

4.4.5 Fusion Without Considering Weights of Each Sensor

The multiple simultaneous threats detection system used in the empirical experiment comprises a combination of four intrusion detection systems, with each having its own means of threat detection. Each intrusion detection system has a different degree of reliability. To simulate that each intrusion detection system is equally likely to receive any of the four types of threats, their individual weights (W) are set to be equal, i.e. W1=W2=W3=W4.

For example, intrusion detection system 1 believes that hypotheses T1 is true with a confidence represented by the mass probability M1 (T1). Similarly, the 2nd, 3rd and 4th systems believe with confidences M2 (T2), M3 (T3) and M4 (T4), respectively. The Dempster-Shafer model combines beliefs and confidences from these independent intrusion detection systems using the Dempster-Shafer theory.

Unlike the Bayesian theory, a-priori knowledge of the four intrusion detection systems is not required. The Dempster-Shafer model can combine the beliefs of these four intrusion detection systems to estimate a single result for each type of the threats.

The Dempster-Shafer model for combining the probability masses of the threats from more than two independent intrusion detection systems is computed as follow:

$$\sum_{i,j=0}^{n} M_i(T_{i,j}) = \sum_{i,j=0}^{n} \frac{p(\{T_{i,j}\})}{p(\{T_{i,j}\}) + p(\{\neg T_{i,j}\})} \quad (19)$$

Where $M_i(T_{i,j})$ is the combined probability mass function of threats i and j, $T_{i,j}$ is the i^{th} threat of the j^{th} intrusion detection system, and $P(\{T_{i,j})$ is the probability of the i^{th} threat of the j^{th} intrusion detection system for a particular type of the threat.

4.4.6 Dempster-Shafer Combined Probability Mass Functions

The calculation of the combined probability mass function will be:

$$P(\{T_{1,1}\}) = \frac{Detected\ \ Alerts}{Observed\ \ Alerts} = \frac{17}{51} = 0.33 \quad (20)$$

$P(\{T_{1,1}\})$ is the probability assigned to the 1st threat by the 1st intrusion detection system.

$$P(\{T_{2,2}\}) = \frac{Detected\ \ Alerts}{Observed\ \ Alerts} = \frac{9}{33} = 0.27 \quad (21)$$

$P(\{T_{2,2}\})$ is the probability assigned to the 2nd threat by the 2nd intrusion detection system.

$$P(\{T_{3,3}\}) = \frac{Detected\ \ Alerts}{Observed\ \ Alerts} = \frac{5}{13} = 0.38 \tag{22}$$

$P(\{T_{3,3}\})$ is the probability assigned to the 3rd threat by the 3rd Intrusion detection system.

$$P(\{T_{4,4}\}) = \frac{Detected\ \ Alerts}{Observed\ \ Alerts} = \frac{9}{26} = 0.34 \tag{23}$$

$P(\{T_{4,4}\})$ is the probability assigned to the 4th threat by the 4th Intrusion detection system.

4.4.6.1. An Example of Two Threats

In the following example, equation (19) is applied to a situation in which there are two threats.

$$M_{1,2}(T_{1,2}) = \frac{P(\{T_1\})\ P(\{T_2\})}{P(\{T_1\})\ P(\{T_2\}) + P(\{\neg T_1\})\ P(\{\neg T_2\})} \tag{24}$$

where $M_{1,2}(T_{1,2})$ is the combined probability mass function of threats T_1 and T_2

$P(\{T_1\})$ is the probability mass of threat T_1

$P(\{T_2\})$ is the probability mass of threat T_2

$$P(\{\neg T_1\}) = 1 - P(\{T_1\}) \quad (25)$$

$$P(\{\neg T_2\}) = 1 - P(\{T_2\}) \quad (26)$$

Putting these values in the formulae:

$$P(\{T_1\}) = 0.33 \quad (27)$$

$$P(\{T_2\}) = 0.27 \quad (28)$$

$$P(\{\neg T_1\}) = 1 - P(\{T_1\}) = 0.66 \quad (29)$$

$$P(\{\neg T_2\}) = 1 - P(\{T_2\}) = 0.72 \quad (30)$$

Putting values in the above equation

$$M_{1,2}(T_{1,2}) = \frac{0.33 * 0.27}{0.33 * 0.27 + 0.66 * 0.72}$$

$M_{1,2}(T_{1,2})$ = 0.1578 is the weighted combined probability mass assigned to the 1st and 2nd threats by the 1st and 2nd intrusion detection systems using equation (24).

4.4.6.2. An Example of Three Threats

In the following example, equation (19) is applied to a situation in which there are three threats.

$$M_{1,2,3}(T_{1,2,3}) = \frac{P(\{T_1\})\ P(\{T_2\})\ P(\{T_3\})}{P(\{T_1\})\ P(\{T_2\})\ P(\{T_3\})\ + P(\{\neg T_1\})\ P(\{\neg T_2\})\ P(\{\neg T_3\})} \quad (31)$$

Where $M_{1,2,3}(T_{1,2,3})$ is the combined probability mass function of threats T_1, T_2 and T_3

$P(\{T_1\})$ is the probability mass of threat T_1

$P(\{T_2\})$ is the probability mass of threat T_2

$P(\{T_3\})$ is the probability mass of threat T_3

$$P(\{\neg T_1\}) = 1 - P(\{T_1\}) \quad (32)$$

$$P(\{\neg T_2\}) = 1 - P(\{T_2\}) \quad (33)$$

$$P(\{\neg T_3\}) = 1 - P(\{T_3\}) \quad (34)$$

Putting values in the equations:

$$P(\{T_1\}) = 0.33 \quad (35)$$

$$P(\{T_2\}) = 0.272 \quad (36)$$

$$P(\{T_3\}) = 0.38 \quad (37)$$

$$P(\{\neg T_1\}) = 1 - P(\{T_1\}) = 0.66 \quad (38)$$

$$P(\{\neg T_2\}) = 1 - P(\{T_2\}) = 0.72 \quad (39)$$

$$P(\{\neg T_3\}) = 1 - P(\{T_3\}) = 0.61 \quad (40)$$

Putting values in the equation

$$M_{1,2,3}(T_{1,2,3}) = \frac{0.33 * 0.27 * 0.38}{0.33 * 0.27 * 0.38 + 0.66 * 0.72 * 0.61}$$

$M_{1,2,3}(T_{1,2,3})$ = 0.104895105, is the weighted combined probability mass assigned to the 1st, 2nd and 3rd threats by the 1st, 2nd and 3rd intrusion detection systems using equation (31).

4.4.6.3. An Example of Four Threats

In the following example, equation (19) is applied to a situation in which there are four threats.

$$M_{1,2,3,4}(T_{1,2,3,4}) = \frac{P(\{T_1\})\ P(\{T_2\})\ P(\{T_3\})\ P(\{T_4\})}{P(\{T_1\})\ P(\{T_2\})\ P(\{T_3\})\ P(\{T_4\}) + P(\{\neg T_1\})\ P(\{\neg T_2\})\ P(\{\neg T_3\})\ P(\{\neg T_4\})} \quad (41)$$

Where $M_{1,2,3}(T_{1,2,3})$ is the combined probability mass function of the threats T_1, T_2 and T_3

$P(\{T_1\})$ is the probability mass of threat T_1

$P(\{T_2\})$ is the probability mass of threat T_2

$P(\{T_3\})$ is the probability mass of threat T_3

$P(\{T_4\})$ is the probability mass of threat T_4

$P(\{\neg T_1\}), P(\{\neg T_2\})$ and $P(\{\neg T_3\})$ will have same formula as equations 32, 33 and 34.

$$P(\{\neg T_4\}) = 1 - P(\{T_4\}) \quad (42)$$

Putting values in the equations:

$$P(\{T_1\}) = 0.33 \quad (43)$$

$$P(\{T_2\}) = 0.27 \quad (44)$$

$$P(\{T_3\}) = 0.38 \quad (45)$$

$$P(\{T_4\}) = 0.34 \quad (46)$$

$$P(\{\neg T_1\}) = 1 - P(\{T_1\}) = 0.66 \quad (47)$$

$$P(\{\neg T_2\}) = 1 - P(\{T_2\}) = 0.72 \quad (48)$$

$$P(\{\neg T_3\}) = 1 - P(\{T_3\}) = 0.61 \quad (49)$$

$$P(\{\neg T_4\}) = 1 - P(\{T_4\}) = 0.65 \quad (50)$$

Putting values in the equation

$$M_{1,2,3,4}(T_{1,2,3,4}) = \frac{0.33 * 0.27 * 0.38 * 0.34}{0.33 * 0.27 * 0.38 * 0.34 + 0.66 * 0.72 * 0.61 * 0.65}$$

Solving the above equation, the combined probability mass of the four intrusion detection systems is:

$M_{1,2,3,4}(T_{1,2,3,4}) = 0.05$ which is the weighted combined probability mass of the 1st, 2nd, 3rd and 4th threats by the 1st, 2nd, 3rd and 4th intrusion detection systems using equation (41).

In this experiment, only four threats and four intrusion detection systems participated in gathering data. That is how the combined probability mass formulae for two, three and four threats were calculated.

4.4.7 Limitation of Proposed Enhancements

One of the formulae above for combining the probability masses of two threats based on the Dempster-Shafer theory is only valid when there are only two intrusion detection systems. As the experiment progressed, both the number of threats and the number of intrusion detection systems were gradually increased. In other words, two threats and two intrusion detection systems were used initially, then three threats and three intrusion detection systems, when eventually four threats and four intrusion detection systems were used.

4.5 Extended Dempster-Shafer Theory to Fuse Data

Extending the Dempster-Shafer theory, the Extended Dempster-Shafer theory postulates the weights of the sensors (i.e. the reliability of the sensors) while calculating the combined probability mass.

4.5.1 Overview of the Extended Dempster Shafer Theory

Unlike conventional probability theories, the Extended Dempster-Shafer Theory assigns a mathematical value of uncertainty to an event by assigning probability masses to sets or intervals. Extended Dempster-Shafer Theory does not require any assumption for assigning probability masses to the sets or intervals. Therefore, this extension of the original theory is a reliable method for obtaining precise inferences from experiments when data are collected by expert systems like the intrusion detection systems. The main feature of the Extended Dempster-Shafer theory is the assignment of weights to each expert system (i.e. intrusion detection system in this research) while implementing the combination rules.

The main difference between the Dempster-Shafer and the Extended Dempster-Shafer theories is that the original theory, without the extension, has the disadvantage that all of the observers or sensors might not detect threats with the same degree of accuracy. This is remedied in the Extended Dempster-Shafer theory as it provides more precision by giving weights of the evidences. Further details of the Extended Dempster-Shafer theory are given in the following section.

4.5.2 Evidence to Proposition Assignments

The Bayesian and Dempster-Shafer theories have certain shortcomings. Therefore, in order to obtain higher accuracy and precision in decision making about threats detection, advancement in data fusion techniques are required, as discussed below.

Unlike the Bayesian decision theory, the Dempster-Shafer model can assign evidence to a single proposition or group of propositions in an experiment, as well as combine the probability masses of the propositions emerging from more than two sources. However, the definition of evidence (probability mass) is not sufficiently accurate. The Dempster-Shafer Theory of inference also has some issues in the renormalisation of the probability mass during the combination step.

In addition, the Bayesian decision theory has shortcomings due both to its inability to test proposition(s) and to its combination rules during the processing of multisensor data. The Bayesian decision theory cannot differentiate between uncertainty and ignorance. Moreover, it requires the assignment of evidence to a hypothesis. The Dempster-Shafer Theory of inference is an extension of the Bayesian decision theory which overcomes this issue. It can assign evidences to a single proposition or group of propositions in an experiment and can combine the probability masses of the propositions emerging from more than two sources.

In this research, two different approaches of weighting the observations are used to improve decision making.

4.5.3 Fusion With Considering Weight of Each Sensor

The assumption made in the aforementioned data fusion model is that all intrusion detection systems have the same weights or degrees of accuracy in detecting a particular type of threat. This assumption is not valid in this research because the four intrusion detection systems are different and as such have different levels of accuracy. As a result, each of the four intrusion detection

systems, even when detecting the same type of threats, may provide different levels of precision. If one intrusion detection system is more accurate than the others in determining a particular type of threats, it would be misleading to assign the same weights to all of the intrusion detection systems.

Therefore, one needs to measure the weight of each intrusion detection system that signifies its level of precision and reliability for the detection of a particular threat. There are many methods to perform such a measurement. The one that is used in this research is the Maximum Entropy method which calculates the weight of each intrusion detection system in threat detection. (Graham Wallies derivation) [72]

As each of the four intrusion detection systems is running on a different computer, it is a reasonable to expect that the reliability of each would be different. The Dempster-Shafer Theory and the Extended Dempster-Shafer models provided the basis for numerical methods used in the detection of multiple threats in the data collected while varying reliabilities of the intrusion detection systems are assumed.

The formula for calculating the probability masses and weights of an intrusion detection system for a particular threat is:

$$\sum_{i,j=0}^{n} M_{i}(T_{i,j})^{W_{i}^{n}} = \sum_{i,j=0}^{n} \frac{P(\{T_{i,j}\})^{W_{i}^{n}}}{P(\{T_{i,j}i\})^{W_{i}^{n}} + P(\{\neg T_{i,j}\})^{W_{i}^{n}}} \qquad (51)$$

where T is the threat and W is the weight of the intrusion detection system, and P is the probability of the i^{th} threat of the j^{th} intrusion detection system.

And $$P(\{T_{i.j}\})^{W_i^n} = (1 - P\{\neg T_{i.j}\})^{W_i^n} \quad (52)$$

$P(\{T_{i.j}\})^{W_i^n}$ is the probability assigned to the i^{th} threat by the j^{th} intrusion detection system with weight.

The formula for calculating the weight of an intrusion detection system for a particular threat is: $$W_j^n = -\sum_{i,j=1}^{n} P_i \log P \quad (53)$$

where W is the combined weight of the intrusion detection systems (sensors) and P is the probability of an i^{th} threat of the j^{th} intrusion detection system [72].

4.5.3.1. Determining the Weights of Observations

Proposition 1: T_1 threat is DoS

(T1 will have four weights assigned by each of the four intrusion detection systems)

Proposition 2: T_2 threat is MITMA

(T2 will have four weights assigned by each of the four intrusion detection systems)

Proposition 3: T_3 threat is Buffer Overflow

(T3 will have four weights assigned by each of the four intrusion detection systems)

Proposition 4: T_4 threat is Trojan

(T4 will have four weights assigned by each of the four intrusion detection systems)

4.5.3.2. Limitations in Calculating Weights

Weights calculated by using the Max Entropy (MaxEnt) method is verified using the Minimum Mean Square Error (MMSE) and the standard gradient descending algorithm.

4.5.3.3. Note on Generalized Evidential Processing

Thomopoulos [25] made an extension to the Bayesian and Dempster-Shafer theories and proposed the Generalised Evidence Processing (GEP) method that presents separate propositions from decisions while calculating the combined probability masses. Therefore, it provides an improved method for calculating the combined probability masses of the evidences of an occurrence or an event. Each proposition or set of propositions can be tested and analysed separately at different levels of the data [21].

Through the process of renormalisation, GEP minimises the differences in evidences caused by Dempster-Shafer when it assigns evidences to different

propositions. This is done by minimising the gaps between sensors' evidences. For further details on GEP, please refer to [21].

4.5.4 Extended Dempster-Shafer Enhanced With Weights

Calculations of probability masses based on the Extended Dempster-Shafer model are as follows:

$$P(\{T_{1,1}\})^{W_1^n} = \frac{Detected \quad Alerts}{Observed \quad Alerts} = \frac{18}{33} = 0.54 \tag{54}$$

where $P(\{T_{1,1}\})^{W_1^n}$ is the weighted probability assigned to the 1st threat by the 1st intrusion detection system.

$$P(\{T_{2,2}\})^{W_2^n} = \frac{Detected \quad Alerts}{Observed \quad Alerts} = \frac{18}{28} = 0.64 \tag{55}$$

where $P(\{T_{2,2}\})^{W_2^n}$ is the weighted probability assigned to the 2nd threat by the 2nd intrusion detection system.

$$P(\{T_{3,3}\})^{W_3^n} = \frac{Detected \quad Alerts}{Observed \quad Alerts} = \frac{8}{22} = 0.36 \tag{56}$$

where $P(\{T_{3,3}\})^{W_3^n}$ is the weighted probability assigned to the 3rd threat by the 3rd intrusion detection system.

$$P(\{T_{4,4}\})^{W_4^n} = \frac{Detected \quad Alerts}{Observed \quad Alerts} = \frac{9}{17} = 0.52 \tag{57}$$

where $P(\{T_{4,4}\})^{W_4^n}$ is the weighted probability assigned to the 4th threat by the 4th intrusion detection system.

The sum of weights of the intrusion detection systems:

$$W_1^n = -\sum_{i=1}^{n} P1 \log P1 = 0.14 \tag{58}$$

where W_1^n is the weight of the 1st intrusion detection system

$$W_2^n = -\sum_{i=1}^{n} P2 \log P2 = 0.12 \tag{59}$$

where W_2^n is the weight of the 2nd intrusion detection system

$$W_3^n = -\sum_{i=1}^{n} P3 \log P3 = 0.15 \tag{60}$$

where W_3^n is the weight of the 3rd intrusion detection system

$$W_4^n = -\sum_{i=1}^{n} P4 \log P4 = 0.14 \tag{61}$$

where W_4^n is the weight of the 4th intrusion detection system

4.5.4.1. An Example of Two Threats

In this example, equation (51) is applied to a situation in which there are two threats:

$$M_{1,2}(T_{1,2})^{W_i^n} = \frac{P(\{T_1\})^{W_1^n} P(\{T_2\})^{W_2^n}}{P(\{T_1\})^{W_1^n} P(\{T_2\})^{W_2^n} + P(\{\neg T_1\})^{W_1^n} P(\{\neg T_2\})^{W_2^n}} \quad (62)$$

where $M_{1,2}(T_{1,2})^{W_i^n}$ is the weighted combined probability mass function of threats T_1 and T_2

$P(\{T_1\})^{W_1^n}$ is the weighted probability mass of threat T_1

$P(\{T_2\})^{W_2^n}$ is the weighted probability mass of threat T_2

$$P(\{\neg T_1\})^{W_1^n} = 1 - P(\{T_1\})^{W_1^n} \quad (63)$$

$$P(\{\neg T_2\})^{W_2^n} = 1 - P(\{T_2\})^{W_2^n} \quad (64)$$

Referring to the above equations, we can calculate the followings:

$$P(\{T_1\})^{W_1^n} = 0.14 \quad (65)$$

$$P(\{T_2\})^{W_2^n} = 0.12 \quad (66)$$

$$P(\{\neg T_1\})^{W_1^n} = 1 - P(\{T_1\})^{W_1^n} = 0.86 \quad (67)$$

$$P(\{\neg T_2\})^{W_2^n} = 1 - P(\{T_2\})^{W_2^n} = 0.88 \quad (68)$$

Putting values in the equations:-

$$M_{1,2}(T_{1,2})^{W_i^n} = \frac{0.14 * 0.12}{0.14 * 0.12 + 0.88 * 0.86}$$

Therefore, $M_{1,2}(T_{1,2})^{W_i^n}$ = 0.52, is the weighted combined probability mass of the probabilities assigned to the 1st and 2nd threats by the 1st and 2nd intrusion detection systems using equation (62).

4.5.4.2. An Example of Three Threats

In this example, equation (51) is applied to a situation in which there are three threats:

$$M_{1,2,3}(T_{1,2,3})^{W_i^n} = \frac{P(\{T_1\})^{W_1^n} P(\{T_2\})^{W_2^n} P(\{T_3\})^{W_3^n}}{P(\{T_1\})^{W_1^n} P(\{T_2\})^{W_2^n} P(\{T_3\})^{W_3^n} + P(\{\neg T_1\})^{W_1^n} P(\{\neg T_2\})^{W_2^n} P(\{\neg T_3\})^{W_3^n}} \quad (69)$$

where $M_{1,2,3}(T_{1,2,3})^{W_i^n}$ is the weighted combined probability mass function of threats T_1, T_2 and T_3

$P(\{T_1\})^{W_1^n}$ is the weighted probability mass of threat T_1

$P(\{T_2\})^{W_2^n}$ is the weighted probability mass of threat T_2

$P(\{T_3\})^{W_3^n}$ is the weighted probability mass of threat T_3

$$P(\{\neg T_1\})^{W_1^n} = 1 - P(\{T_1\})^{W_1^n} \quad (70)$$

$$P(\{\neg T_2\})^{W_2^n} = 1 - P(\{T_2\})^{W_2^n} \quad (71)$$

$$P(\{\neg T_3\})^{W_3^n} = 1 - P(\{T_3\})^{W_3^n} \quad (72)$$

Referring to the above equations, we can calculate the followings:

$$P(\{T_1\})^{W_1^n} = 0.91 \quad (73)$$

$$P(\{T_2\})^{W_2^n} = 0.94 \quad (74)$$

$$P(\{T_3\})^{W_3^n} = 0.85 \quad (75)$$

$$P(\{\neg T_1\})^{W_1^n} = 1 - P(\{T_1\})^{W_1^n} = 0.89 \quad (76)$$

$$P(\{\neg T_2\})^{W_2^n} = 1 - P(\{T_2\})^{W_2^n} = 0.88 \quad (77)$$

$$P(\{\neg T_3\})^{W_3^n} = 1 - P(\{T_3\})^{W_3^n} = 0.93 \quad (78)$$

Putting values in the equations:-

$$M_{1,2,3}(T_{1,2,3})^{W_i^n} = \frac{0.91 * 0.94 * 0.85}{0.89 * 0.88 * 0.85 + 0.89 * 0.88 * 0.93}$$

$$M_{1,2,3}(T_{1,2,3})^{W_i^n} = 0.50 \quad (79)$$

where $M_{1,2,3}(T_{1,2,3})^{W_i^n}$ is the weighted combined probability mass assigned to the 1st, 2nd and 3rd threats by the 1st, 2nd and 3rd intrusion detection systems using equation (69).

4.5.4.3. An Example of Four Threats

In this example, equation (51) is applied to a situation in which there are four threats:

$$M_{1,2,3,4}(T_{1,2,3,4})^{W_i^n} = \frac{P(\{T_1\})^{W_1^n} P(\{T_2\})^{W_2^n} P(\{T_3\})^{W_3^n} P(\{T_4\})^{W_4^n}}{P(\{T_1\})^{W_1^n} P(\{T_2\})^{W_2^n} P(\{T_3\})^{W_3^n} P(\{T_4\})^{W_4^n} + P(\{\neg T_1\})^{W_1^n} P(\{\neg T_2\})^{W_2^n} P(\{\neg T_3\})^{W_3^n} P(\{\neg T_4\})^{W_4^n}} \quad (80)$$

where $M_{1,2,3,4}(T_{1,2,3,4})^{W_i^n}$ is the weighted combined probability mass function of threats T_1, T_2, T_3 and T_4

$P(\{T_1\})^{W_1^n}$ is the weighted probability mass of threat T_1

$P(\{T_2\})^{W_2^n}$ is the weighted probability mass of threat T_2

$P(\{T_3\})^{W_3^n}$ is the weighted probability mass of threat T_3

$P(\{T_4\})^{W_4^n}$ is the weighted probability mass of threat T_4

$P(\{\neg T_1\})^{W_1^n}$, $P(\{\neg T_2\})^{W_2^n}$ and $P(\{\neg T_3\})^{W_3^n}$ have the same formula as equation 70, 71 and 72.

$$P(\{\neg T_4\})^{W_4^n} = 1 - P(\{T_4\})^{W_4^n} \quad (81)$$

Referring to the above equations, we can calculate the followings:

$P(\{T_1\})^{W_1^n}$, $P(\{T_2\})^{W_2^n}$ and $P(\{T_3\})^{W_3^n}$ formula are the same as equations 73, 74 and 75.

$$P(\{T_4\})^{W_4^n} = 0.57 \quad (82)$$

$$P(\{\neg T_1\})^{W_1^n} = 1 - P(\{T_1\})^{W_1^n} = 0.89 \quad (83)$$

$$P(\{\neg T_2\})^{W_2^n} = 1 - P(\{T_2\})^{W_2^n} = 0.88 \quad (84)$$

$$P(\{\neg T_3\})^{W_3^n} = 1 - P(\{T_3\})^{W_3^n} = 0.93 \quad (85)$$

$$P(\{\neg T_4\})^{W_4^n} = 1 - P(\{T_4\})^{W_4^n} = 0.89 \quad (86)$$

Putting values in the equations:-

$$M_{1,2,3,4}(T_{1,2,3,4})^{W_i^n} = \frac{0.91 * 0.94 * 0.85 * 0.57}{0.89 * 0.88 * 0.85 * 0.57 + 0.89 * 0.88 * 0.93 * 0.89}$$

Therefore $M_{1,2,3,4}(T_{1,2,3,4})^{W_i^n}$ = 0.39, is the weighted combined probability assigned to the 1st, 2nd, 3rd and 4th threats by the 1st, 2nd, 3rd and 4th intrusion detection systems using equation (80).

4.6 Data Fusion using Generalized Evidential Processing Theory

4.6.1 Overview of the Generalized Evidential Processing Theory

Generalized Evidential Processing Theory (GEP) is a generalization of Bayesian theory. Similar to the Bayesian theory, GEP can assign evidence to a hypothesis only, while Dempster-Shafer has an additional advantage that it can assign evidence to both conflicting propositions and hypotheses. As this research deals only with testing hypotheses, GEP and Dempster-Shafer assume much the same role in assigning evidences to hypotheses. However, GEP assigns and combines probability masses based on a-priori conditional probability while Dempster-Shafer updates a-priori probability of the hypothesis based on observational evidence. Therefore, in this research, a-priori probability of the hypotheses (threats) will be assigned by the evidence gathering approach (i.e. Dempster-Shafer, Extended Dempster-Shafer, or GEP) that the underlying intrusion detection system has implemented.

The main benefit of GEP is that it separates hypotheses from decisions. GEP also describes the relationship of evidential assignments with fusion decisions. This provides the ability to test hypotheses at different quantization of the data.

The formula for combining the probability masses of the threats from more than two independent intrusion detection systems by GEP is:

$$\sum_{i,j=0}^{n} M_i(T_{i,j}) = \sum_{i,j=0}^{n} \frac{P(\{T_{i,j}\})P(\{T_{i,j}/S_{i,j}\})}{P(\{T_{i,j}\})P(\{T_{i,j}/S_{i,j}\})+P(\{\neg T_{i,j}\})P(\{\neg T_{i,j}/S_{i,j}\})} \quad (87)$$

where $\sum_{i,j=0}^{n} M_i(T_{i,j})$ is the combined probability mass of the independent intrusion detection systems.

$P(\{T_{i,j} / S_{i,j}\})$ is the probability of assumed threats being tested positive given the i^{th} threat of the j^{th} intrusion detection system. That is, the detection accuracy of each intrusion detection system is expressed as a percentage. The detection accuracies of MARS, Sniffers, Snoop and Wireshark are 95%, 80%, 75% and 80%, respectively.

$P(T_{i,j})$ is the probability of the i^{th} threat observed by the j^{th} intrusion detection system. The numerical figures for the accuracy of the intrusion detection systems are not available in the literature. Therefore, these values are purely assumptions based on the detection accuracies obtained from experiments in this research. When more reliable numerical figures become available in the future, research can make use of these figures to obtain better detection accuracies under the GEP fusion model.

4.6.2 Empirical Assessment

In this example, equation (87) is applied to a situation in which there are two threats:

$$\sum_{i,j=1}^{2} M_{1,2}(T_{1,2}) = \sum_{i,j=1}^{2} \frac{P(\{T_{1,2}\})P(\{T_{1,2} / S_{1,2}\})}{P(\{T_{1,2}\})P(\{T_{1,2} / S_{1,2}\}) + P(\{\neg T_{1,2}\})P(\{\neg T_{1,2} / S_{1,2}\})} \quad (88)$$

where $M_{1,2}(T_{1,2})$ is the combined probability mass of the 1st and 2nd threats assigned by the 1st and 2nd intrusion detection systems.

$P(\{T_{1,2}/S_{1,2}\})$ is the probability of a test returned positive given for the 1st and 2nd threats by the 1st and 2nd intrusion detection systems.

$P(\{T_{1,2}\})$ is the probability of the 1st and 2nd threats observed by the 1st and 2nd intrusion detection systems.

$$P(\{\neg T_{1,2}\}) = 1 - P(\{T_{1,2}\}) \quad (89)$$

$$P(\{\neg T_{1,2}/S_{1,2}\}) = 1 - P(\{T_{1,2}/S_{1,2}\}) \quad (90)$$

4.4.6.1 An Example of Two Threats

In the following example, the calculation for the probability of a test returned positive given two threats by intrusion detection systems using actual figures:

1st Threat:-

Detected Alerts=19 (91)

Observerd Threats=31 (92)

Detection Accuracy by IDS 1=95%

Probability of a test positive given for the 1st threat by 1st intrusion detection system

$$P(T_1 / S_1) = \frac{\text{Detected Alerts}}{\text{Observerd Threats}} * \text{Detection Accuracy by IDS} \quad (93)$$

Putting values in the equations

$$P(T_1 / S_1) = \frac{19}{31} * 0.95 = 0.58 \quad (94)$$

2nd Threat:-

Detected Alerts=19 (95)

Observerd Threats=28 (96)

Detection Accuracy by IDS 1=80%

Probability of a test positive given for the 2nd threat by 2nd intrusion detection system

$$P(T_2 / S_2) = \frac{\text{Detected Alerts}}{\text{Observerd Threats}} * \text{Detection Accuracy by IDS} \quad (97)$$

Putting values in the equations

$$P(T_2 / S_2) = \frac{19}{28} * 0.80 = 0.54 \quad (98)$$

$$P(\{T_{1,2} / S_{1,2}\}) = P(T_1 / S_1) * P(T_2 / S_2) \quad (99)$$

Probability of a test positive given for the 1st & 2nd threats
by 1st & 2nd intrusion detection systems

$$P(\{T_{1,2} / S_{1,2}\})=0.582258065*0.542857143 = 0.316082949 \quad (100)$$

Putting values in the equations

$$P(\{T_{1,2}\}) = 0.58 \quad (101)$$

$$P(\{T_{1,2} / S_{1,2}\}) = 0.54 \quad (102)$$

$$P(\{\neg T_{1,2}\}) = 1 - P(\{T_{1,2}\}) = 0.41 \quad (103)$$

$$P(\{\neg T_{1,2} / S_{1,2}\}) = 1 - P(\{T_{1,2} / S_{1,2}\}) = 0.45 \quad (104)$$

Putting values in the equations:-

$$M_{1,2}(T_{1,2}) = \frac{0.58 * 0.54}{0.58 * 0.54 + 0.41 * 0.45}$$

$$M_{1,2}(T_{1,2}) = 0.623375443$$

4.6.2.1. An Example of Three Threats

In this example, equation (87) is applied to a situation in which there are three threats:

$$\sum_{i,j=1}^{3} M_{1,2,3}(T_{1,2,3}) = \sum_{i,j=1}^{3} \frac{P(\{T_{1,2,3}\})P(\{T_{1,2,3} / S_{1,2,3}\})}{P(\{T_{1,2,3}\})P(\{T_{1,2,3} / S_{1,2,3}\}) + P(\{\neg T_{1,2,3}\})P(\{\neg T_{1,2,3} / S_{1,2,3}\})} \quad (105)$$

where $M_{1,2,3}(T_{1,2,3})$ is the combined probability mass of the 1st, 2nd and 3rd threats assigned by the 1st, 2nd and 3rd intrusion detection systems.

$P(\{T_{1,2,3}/S_{1,2,3}\})$ is the probability of a positive test result given for the 1st, 2nd and 3rd threats by the 1st, 2nd and 3rd intrusion detection systems.

$P(\{T_{1,2,3}\})$ is the probability of the 1st, 2nd and 3rd threats observed by the 1st, 2nd and 3rd intrusion detection systems.

$$P(\{\neg T_{1,2,3}\})=1-P(\{T_{1,2,3}\}) \quad (106)$$

$$P(\{\neg T_{1,2,3}/S_{1,2,3}\})=1-P(\{T_{1,2,3}/S_{1,2,3}\}) \quad (107)$$

1st and 2nd threat, $P(T_1/S_1)$ and $P(T_2/S_2)$ formula and results will be the same as equations 93 and 97.

3rd Threat:-

$$Detected \quad Alerts = 9 \tag{108}$$

$$Observerd \quad Threats = 15 \tag{109}$$

Detection Accuracy by IDS 1=76%

$$P(T_3 / S_3) = \frac{Detected \quad Alerts}{Observerd \quad Threats} * Detection\ Accuracy\ by\ IDS \tag{110}$$

Putting values in the equations

$$P(T_3 / S_3) = \frac{9}{15} * 0.76 = 0.45 \tag{111}$$

Probability of a test positive given for the 1st, 2nd and 3rd threats by 1st, 2nd and 3rd intrusion detection systems

$$P(\{T_{1,2,3} / S_{1,2,3}\}) = P(T_1 / S_1) * P(T_2 / S_2 * P(T_3 / S_3) \tag{112}$$

$$P(\{T_{1,2,3} / S_{1,2,3}\}) = 0.58 * 0.54 * 0.45 = 0.14 \tag{113}$$

$$P(\{T_{1,2,3}\}) P(\{T_{1,2,3} / S_{1,2,3}\}) = 0.14 \tag{114}$$

$$P(\{\neg T_{1,2,3}\}) P(\{\neg T_{1,2,3} / S_{1,2,3}\}) = 0.10 \tag{115}$$

Putting values in the equations:-

$$M_{1,2,3}(T_{1,2,3}) = \frac{0.14}{0.14 + 0.10}$$

$$M_{1,2,3}(T_{1,2,3}) = 0.58 \tag{116}$$

$M_{1,2,3}(T_{1,2,3})$ is the combined probability mass of the threats 1, 2 and 3 assigned by intrusion detection systems 1, 2 and 3.

4.6.2.2. An Example of Four Threats

In this example, equation (87) is applied to a situation in which there are four threats:

$$\sum_{i,j=1}^{4} M_{1,2,3,4}(T_{1,2,3,4}) = \sum_{i,j=1}^{4} \frac{P(\{T_{1,2,3,4}\})P(\{T_{1,2,3,4}/S_{1,2,3,4}\})}{P(\{T_{1,2,3,4}\})P(\{T_{1,2,3,4}/S_{1,2,3,4}\})+P(\{\neg T_{1,2,3,4}\})P(\{\neg T_{1,2,3,4}/S_{1,2,3,4}\})} \quad (117)$$

where $M_{1,2,3,4}(T_{1,2,3,4})$ is the combined probability mass of the 1st, 2nd, 3rd and 4th threats assigned by the 1st, 2nd, 3rd and 4th intrusion detection systems.

$P(\{T_{1,2,3,4}/S_{1,2,3,4}\})$ is the probability of a positive test result being given for the 1st, 2nd and 3rd threats by the 1st, 2nd and 3rd intrusion detection systems.

$P(\{T_{1,2,3,4}\})$ is the probability of the 1st, 2nd, 3rd and 4th threat observed by the 1st, 2nd, 3rd and 4th intrusion Detection systems.

$$P(\{\neg T_{1,2,3,4}\}) = 1 - P(\{T_{1,2,3,4}\}) \quad (118)$$

$$P(\{\neg T_{1,2,3,4}/S_{1,2,3,4}\}) = 1 - P(\{T_{1,2,3,4}/S_{1,2,3,4}\}) \quad (119)$$

Putting the above values into the equations:-

1st, 2nd and 3rd threat, $P(T_1/S_1)$, $P(T_2/S_2)$ and $P(T_3/S_3)$ formula and results will be the same as equations 93, 97 and 112.

4th Threat:-

Detected Alerts = 9 (120)

Observerd Threats = 18 (121)

Detection Accuracy by IDS 1=82%

$$P(T_4 / S_4) = \frac{Detected\ \ Alerts}{Observerd\ \ Threats} * Detection\ Accuracy\ by\ IDS \quad (122)$$

Putting values in the equations

$$P(T_4 / S_4) = \frac{9}{18} * 0.82 = 0.41 \quad (123)$$

$$P(\{T_{1,2,3,4} / S_{1,2,3,4}\}) = P(T_1 / S_1) * P(T_2 / S_2 * P(T_3 / S_3 * P(T_4 / S_4) \quad (124)$$

Probability of a test positive given for the 1st, 2nd, 3rd & 4th threats
by 1st, 2nd, 3rd and 4th intrusion detection systems

$$P(\{T_{1,2,3,4} / S_{1,2,3,4}\}) = 0.58 * 0.54 * 0.45 * 0.41 = 0.05 \quad (125)$$

Putting values into the equations:-

$$P(\{T_{1,2,3,4}\}) P(\{T_{1,2,3,4} / S_{1,2,3,4}\}) = 0.05 \quad (126)$$

$$P(\{\neg T_{1,2,3,4}\}) P(\{\neg T_{1,2,3,4} / S_{1,2,3,4}\}) = 0.06 \quad (127)$$

Putting values in the equations:-

$$M_{1,2,3,4}(T_{1,2,3,4}) = \frac{0.05}{0.05 + 0.06}$$

Combined Probability mass of four threats
by four intrusion detection system

$$M_{1,2,3,4}(T_{1,2,3,4}) = 0.49 \quad (128)$$

$M_{1,2,3,4}(T_{1,2,3,4})$ is the combined probability mass of the 1st, 2nd, 3rd and 4th threat assigned by 1st, 2nd, 3rd and 4th intrusion detection system.

Table 4.2: Summary of the probability mass calculation by MSTDS

Dempster-Shafer	Extended Dempster-Shafer	Generalised Evidential Processing
0.06	0.39	0.49
0.11	0.5	0.58
0.16	0.52	0.62

The table 4.2 provides the comparison of the combined probability masses calculated by Dempster Shafer, Extended Dempster Shafer and Generalises Evidential Processing (GEP). It is obvious from the results that Dempster Shafer detected probability mass is 0.16 for two threats detected by two intrusion detection systems, extended Dempster Shafer detected 0.52 and GEP calculated 0.62. So an increase of 0.46 in probability mass has been achieved for two threats. Similarly increases of 0.47 and 0.43 during three and four threats have been achieved with Extended Dempster Shafer and GEP inferences. The increase in probability mass means increase in the precision of threat detection by the intrusion detection system.

5. Experimental Evaluation

In the experiments, threats affected the network nodes. All threats were generated on local private networks off the web and intrusion detection systems filtered the malicious network packets into four groups (each by one of the IDS). Altogether, there were 2,272 threats detected. Each group gathered data that could contain all four types of threats. The threat data contained a mixture of all four types of generated threats. Therefore, the first challenge was to reduce the size of the data by placing threats into different disjoint subsets. The next challenge was in calculating the cost of using each of the IDS.

To resolve the above issues, the set covering approach was used. In this approach, a stage of set packing enabled the selection of the four subsets of the union set *N* of 2,274 threats such that each subset was pair-wise disjoint from other subsets. Thus, each subset had similar or closely related strings representing threats whose union was *N*.

In the next step, in order to reduce the size of the four pair-wise disjoint subsets, the set of *N*=2,274 threat data was processed using a Perl script. The script generated four subsets of a total of 429 threats that were similar or related to each other in one way or another.

Then, the subsets were refined in terms of the types of threats before passing through a filtering process called “threat refinement and identity declaration” as shown in Figure 5.1 below. Bayesian and Dempster-Shafer models were used to measure the uncertainty of the evidences assigned by the intrusion detection

systems to each proposition or hypothesis. The details of the results are given below in Tables 5.2 and 6.1. The data flow process of the experiment was shown in Figure 5.1.

5.1 Experimental Setup

5.1.1 MSTDS Process Model of the Experiment

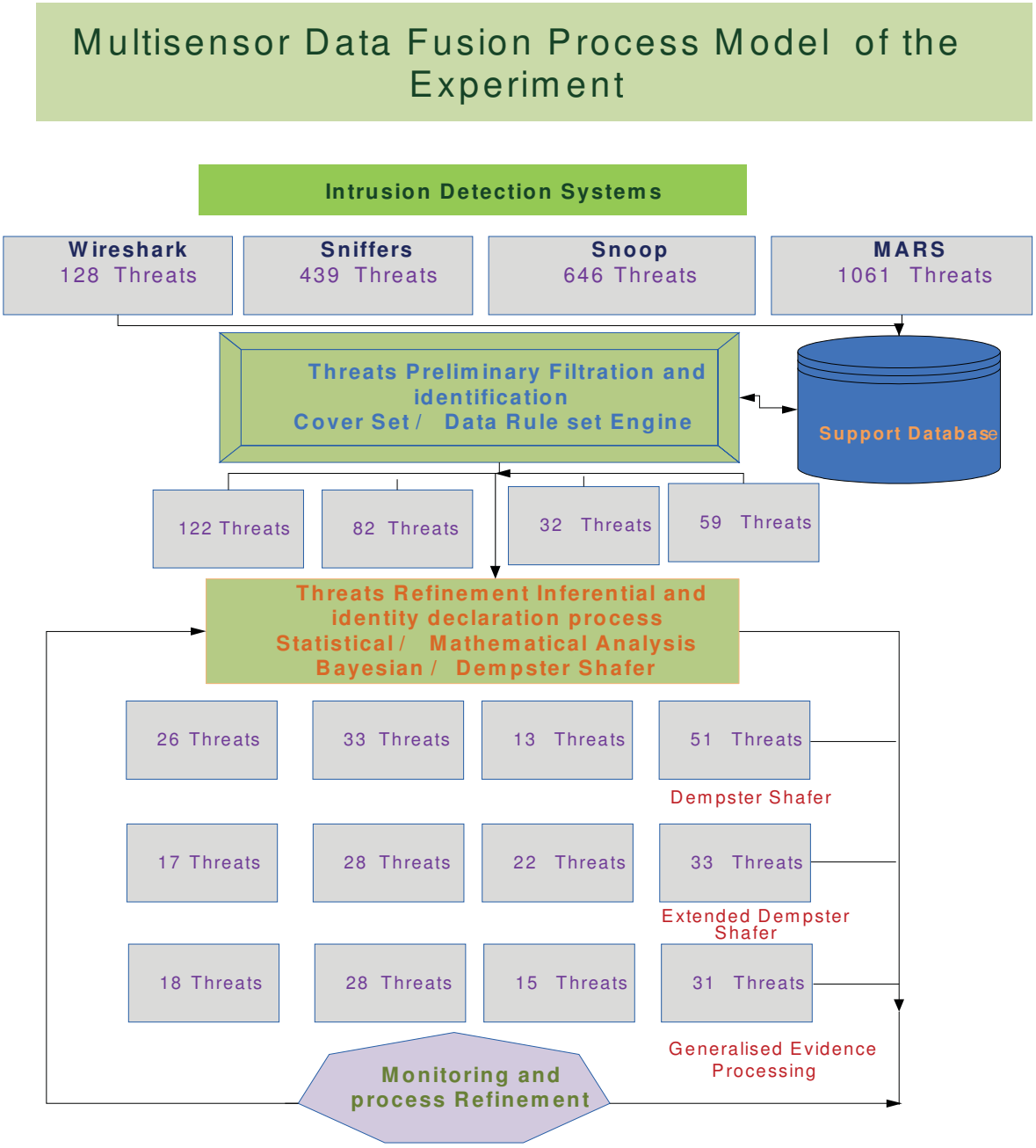

Fig 5.1: Multisensor Data Fusion Process Model of the Experiment

From this, the subsets were fused by using Bayesian, Dempster-Shafer, or Extended Dempster-Shafer theory that runs on each of the IDS.

5.1.2 Context Diagram

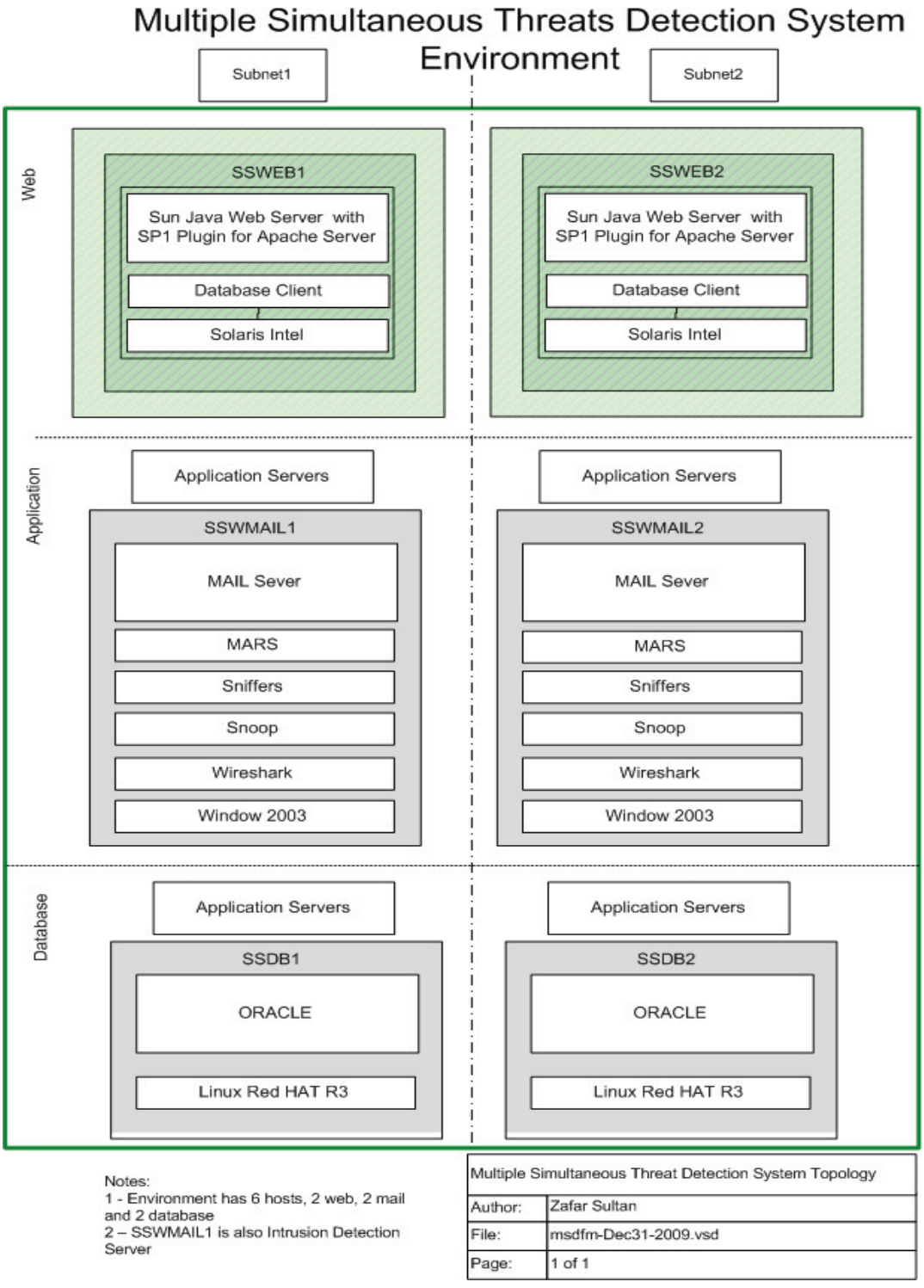

Fig 5.2: Infrastructure of the MSTDS Environment

5.1.2.1. Description of the Context Diagram

The MSTDS environment is shown in the infrastructure in Figure 5.2. Details of the infrastructure are as follows:

- **Hardware:** six Intel servers (across two networks, that means each network had three servers) dual core CPUs, about 2.8GHz or more speed, min 500MB or more RAM, single board and controller along with other accessories. Host names will be SSWEB1, SSDB1, and SSWMAIL1 in the first network and SSWEB2, SSDB2, SSWMAIL2 in the second network.
- **Software:** SSWEB and SSDB with the latest UNIX / Linux operating system, apache, webLogic, Oracle / SQL, one of the security servers, NSUs / IMAT, Cisco Mars and or Cisco Security Agent (CSA). SSWMAIL with a Window server and necessary operating softwares. MARS gathers inputs from all over the network (syslog, SNMP traps, email, log files, etc.) and tries to collate them to determine if there is a network attack going on. It can also proactively enforce ACLs or QOS policy on network devices to mitigate the attack automatically.
- **Networks:** Two private gigabit networks, one with a hyper channel, two four-port switches and two hubs, one internet connection

5.1.3 Multiple Simultaneous Threats Detection Process Model

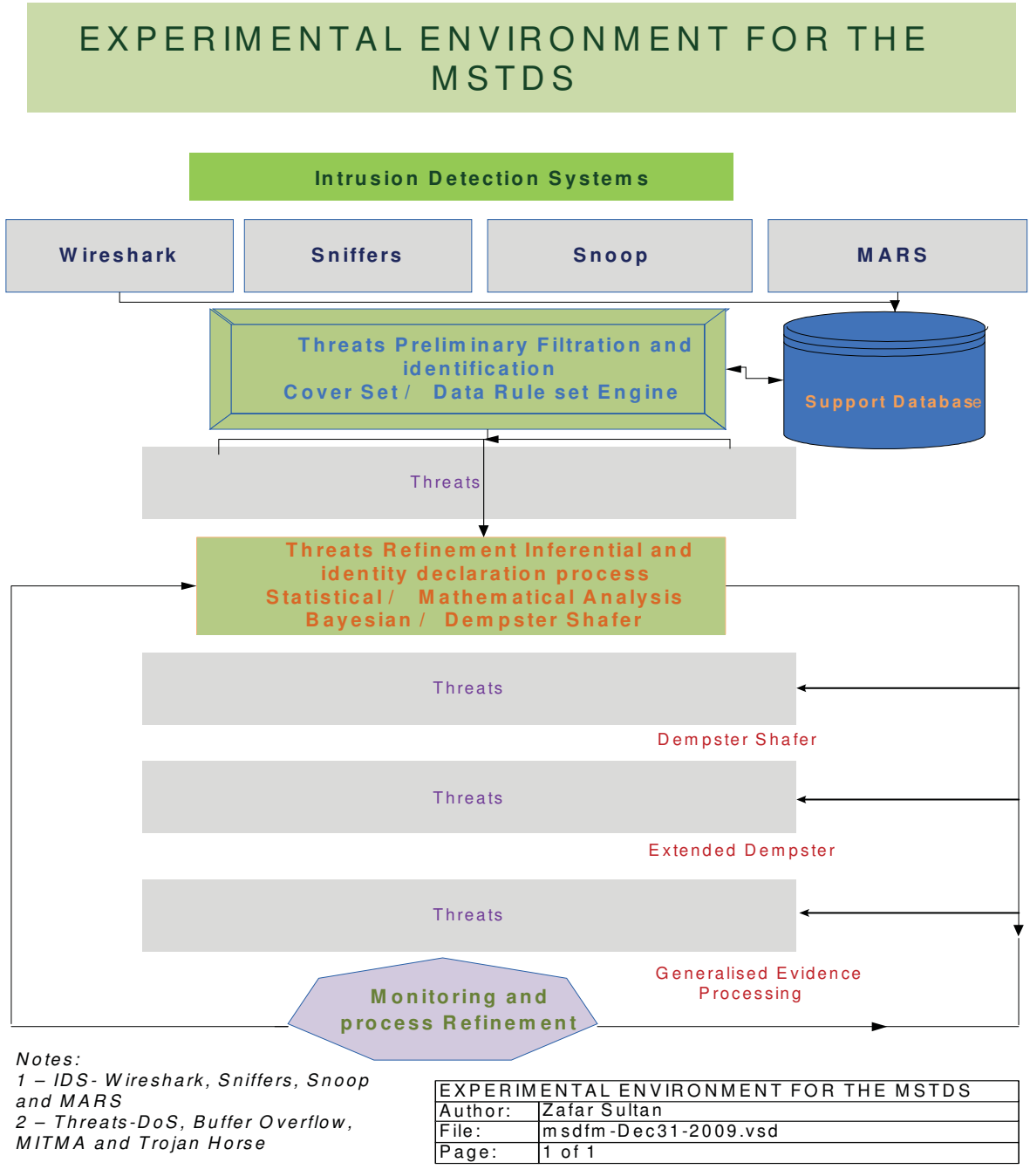

Fig 5.3: Experimental Environment for the MSTDS

In the infrastructure, four independent intrusion detection systems: MARS, Sniffers, Snoop and Wireshark were used. They worked as separate observers on different subnets. There were four types of threats: DoS, MITMA, Buffer Overflow and Trojan horse.

The first level of data fusion was done by the middle-tier tool “set cover” that converts data into smaller subsets of non-disjoint subset of threats. Then, the threat data was passed to the next level that was the multisensor data fusion process. The threat data was analysed by a hybridization of three different mathematical and statistical tools: Dempster-Shafer, Extended Dempster-Shafer, and Generalized Evidential Processing.

5.2 Experimental Results

5.2.1 Based on Dempster-Shafer Fusion Model

After the set covering stage, a total of 295 threats (Table 5.1) remained. The threat data was further processed by the Dempster-Shafer based fusion model, which is the next level of the multiple simultaneous threats detection system as shown in Fig 5.1. In order to increase precision, each threat was passed through multiple hypotheses testing as proposed in Section 5.2.3. The intrusion detection system classified the Dempster-Shafer inferences into four types: observed threats, observed alerts, detected alerts and real alerts. This classification helped in determining the real threats and reduced false positive rates. The final results of this part of the fusion are given in Table 5.4.

False Positive rates are determined using the formula:

$$\text{False Positive Rates} = 1 - \frac{\text{Real Alerts}}{\text{Observerd Alerts}} * 100 \quad (129)$$

And threat detection rates are calculated using the equation:

$$\text{Threat Detection Rate} = \frac{\text{Detected Alerts}}{\text{Observerd Threats}} * 100 \quad (130)$$

Table 5.1: Threat Results based on Dempster-Shafer Theory of Inference

IDS	OT	OA	DA	RA	FPR	Detect Rate
MARS	51	23	31	17	26	60.78
Sniffers	33	19	21	9	53	63.63
Snoop	13	7	7	5	29	53.84
Wireshark	26	12	12	9	25	46.15
Total	123	61	71	40		

where OT stands for Observed Threats, OA for observed Alerts, and DA for Detected Alerts, RA for Real Alerts and FPR for False Positive Rate

5.2.2 Based on Extended Dempster-Shafer Fusion Model

As was the case for the Dempster-Shafer inference, data for 295 threats were analysed using the Extended Dempster-Shafer inference and the intrusion

detection system then grouped the data as shown in Table 5.2. It is obvious that real alerts have gone up from 40 to 53. That is a clear indication that false positive rates have been reduced compared to the rates when using Dempster-Shafer inference.

Likewise, there is an obvious improvement in the threat detection rate as well.

Table 5.2: Threat Results based on Extended Dempster Shafer Theory of Inference

IDS	OT	OA	DA	RA	FPR	Detect Rate
MARS	33	19	31	18	5.3	93.93
Sniffers	28	21	26	18	14	92.85
Snoop	22	9	14	8	11	63.63
Wireshark	17	10	12	9	10	70.58
Total	100	59	83	53		

where OT stands for Observed Threats, OA for observed Alerts, and DA for Detected Alerts, RA for Real Alerts and FPR for False Positive Rate

5.2.3 Based on Generalised Evidential Processing Fusion Model

Generalised Evidential Processing (GEP) improved the detection rates of almost all of the intrusion detection systems, according to our experiments. MARS detection and false positive rates indicated that the GEP model as a highly

reliable fusion model. The other three intrusion detection systems (Sniffers, Snoop and Wireshark) also showed significant improvements on threats detection, showing a considerable drop in false positive rates (Table 5.1).

Table 5.3: Threat Results based on Generalised Evidential Processing

IDS	OT	OA	DA	RA	FPR	Detect Rate
MARS	31	19	31	19	0	100
Sniffers	28	20	26	19	5	92.85
Snoop	15	10	14	9	10	93.33
Wireshark	18	10	17	9	10	94.44
Total	92	59	88	56		

where OT stands for Observed Threats, OA for Observed Alerts, and DA for Detected Alerts, RA for Real Alerts and FPR for False Positive Rate

5.3 Verification of MSTDS using Public Domain Data Sets

In order to verify further the performance of the proposed Multiple Simultaneous Threats Detection System, it was tested on a public domain dataset, the DARPA Intrusion Detection Evaluation, made available by the Lincoln Laboratory of Massachusetts Institute of Technology. This dataset contains strings of TROJAN horse and DDoS attacks collected by using sniffers through TCPdumps. Although

the DARPA Intrusion Detection Evaluation Data Set contains data of only two threats as compared to four simultaneous threats, the MSTDS model was able to detect them accurately based on the experimental results, described below.

5.3.1 Data Fusion Process

As dataset contains two types of threats, DDoS and TROJAN horse, the set cover intermediate level filtered the threat data into two sets, as shown in the table below.

Table 5.4: Set Cover reduces the size of the data of Public Domain Data Sets

IDS	Before Set Cover	After Set Cover
Sniffers / MARS	11538	6171
Snoop / Wireshark	5955	3185
Total	**17493**	**9356**

In order to analyse the data set, VMWARE server 2.0 with Virtual machine and Linux Ubuntu was used to run the Perl script which counted the number of threats and separated the threat data into two disjoint sets, as shown in the table above. As it was not clear from the DARPA dataset which types of Intrusion Detection Systems were used to collect these threats, an assumption was made which allocated sniffers / MARS to DDoS threats and Snoop / Wireshark to TROJAN horse, respectively.

The main function of set cover, as explained in early chapter, is to filter the threat data into small subsets and for selecting the IDS with the minimum cost. In this dataset, 17493 malicious strings were identified from the file (LLS_DDoS). The set of threat data was then filtered into two disjoint groups containing these strings.

Simultaneously, the set cover algorithm was able to determine the cost effectiveness of each IDS in the MSTDS since the number of IDS used in this experiment was the same as the experimental setup I used in earlier experiments. For details of the experimental setup, please refer to Chapter 4.

By applying set cover, a preliminary understanding of the expected number and types of the attacks became possible. In realistic environments, the set cover analysis would be critical since an organization might not be aware of the number and types of the attacks involved. Furthermore, another benefit of the set cover analysis is the cost and effectiveness in processing. In this experiment, only two non-repetitive types of propositions as compared to the initial group of 17,493 propositions that each IDS would have to be processed. By the filtering achieved using set cover, the highly tedious task, both in terms of time and resources, for the subsequent multisensor data fusion process can be reduced.

Based on set cover, there are only two types of propositions on all 4 nodes of the experimental setup, as follows:

- Proposition 1: T_1 , threat is a DoS
- Proposition 2: T_2 , threat is a Trojan

1) Fusion without weights of each sensor

The calculation of the combined probability mass function will be:

$$P(\{T_{1,1}\}) = \frac{Detected\ \ Alerts}{Observed\ \ Alerts} = \frac{539}{1617} = 0.33 \qquad (131)$$

$P(\{T_{1,1}\})$ is the probability assigned to the 1st threat by the 1st intrusion detection system.

$$P(\{T_{2,2}\}) = \frac{Detected\ \ Alerts}{Observed\ \ Alerts} = \frac{285}{1047} = 0.27 \qquad (132)$$

$P(\{T_{2,2}\})$ is the probability assigned to the 2nd threat by the 2nd intrusion detection system.

5.3.2 An Example of Threat detection with two Threats

In the following equation, I apply equation (19) to a situation in which there are two sensors

$$M_{1,2}(T_{1,2}) = \frac{P(\{T_1\})\ P(\{T_2\})}{P(\{T_1\})\ P(\{T_2\}) + P(\{\neg T_1\})\ P(\{\neg T_2\})} \qquad (133)$$

where $M_{1,2}(T_{1,2})$ is the combined probability mass function of threats T_1 and T_2

$P(\{T_1\})$ is the probability mass of threat T_1

$P(\{T_2\})$ is the probability mass of threat T_2

$$P(\{\neg T_1\}) = 1 - P(\{T_1\}) \quad (134)$$

$$P(\{\neg T_2\}) = 1 - P(\{T_2\}) \quad (135)$$

Putting the above values in the formulae:

$$P(\{T_1\}) = 0.33 \quad (136)$$

$$P(\{T_2\}) = 0.27 \quad (137)$$

$$P(\{\neg T_1\}) = 1 - P(\{T_1\}) = 0.67 \quad (138)$$

$$P(\{\neg T_2\}) = 1 - P(\{T_2\}) = 0.73 \quad (139)$$

Putting values in the above equation

$$M_{1,2}(T_{1,2}) = \frac{0.33 * 0.27}{0.33 * 0.27 + 0.67 * 0.73}$$

$M_{1,2}(T_{1,2})$ = 0.16 is the weighted combined probability mass assigned to the 1st and 2nd threats by the 1st and 2nd intrusion detection systems using equation (133).

2) Extended Dempster-Shafer Fusion With Weights

Calculations using the Extended Dempster-Shafer fusion are as follows:

$$P(\{T_{1,1}\})^{W_1^n} = \frac{Detected\ \ Alerts}{Observed\ \ Alerts} = \frac{602}{1041} = 0.58 \qquad (140)$$

where $P(\{T_{1,1}\})^{W_1^n}$ is the weighted probability assigned to the 1st threat by the 1st intrusion detection system.

$$P(\{T_{2,2}\})^{W_2^n} = \frac{Detected\ \ Alerts}{Observed\ \ Alerts} = \frac{540}{888} = 0.61 \qquad (141)$$

where $P(\{T_{2,2}\})^{W_2^n}$ is the weighted probability assigned to the 2nd threat by the 2nd intrusion detection system.

The weights of the intrusion detection systems are calculated as follows:

$$W_1^n = -\sum_{i=1}^{n} P1\log P1 = 0.14 \qquad (142)$$

where W_1^n is the weight of the 1st intrusion detection system for 1st threat.

$$W_2^n = -\sum_{i=1}^{n} P1\log P1 = 0.13 \qquad (143)$$

where W_2^n is the weight of the 1st intrusion detection system for 2nd threat.

An Example of Threat detection with two Threats

In the following example, equation (51) was applied to a situation in which there are two threats.

$$M_{1,2}(T_{1,2})^{W_i^n} = \frac{P(\{T_1\})^{W_1^n} P(\{T_2\})^{W_2^n}}{P(\{T_1\})^{W_1^n} P(\{T_2\})^{W_2^n} + P(\{\neg T_1\})^{W_1^n} P(\{\neg T_2\})^{W_2^n}} \quad (144)$$

where $M_{1,2}(T_{1,2})^{W_i^n}$ is the weighted combined probability mass function of threats T_1 and T_2

$P(\{T_1\})^{W_1^n}$ is the weighted probability mass of threat T_1

$P(\{T_2\})^{W_2^n}$ is the weighted probability mass of threat T_2

$$P(\{\neg T_1\})^{W_1^n} = 1 - P(\{T_1\})^{W_1^n} \quad (145)$$

$$P(\{\neg T_2\})^{W_2^n} = 1 - P(\{T_2\})^{W_2^n} \quad (146)$$

Putting the above values in the formulae:

$$P(\{T_1\})^{W_1^n} = 0.58 \quad (147)$$

$$P(\{T_2\})^{W_2^n} = 0.61 \quad (148)$$

$$P(\{\neg T_1\})^{W_1^n} = 1 - P(\{T_1\})^{W_1^n} = 0.42 \quad (149)$$

$$P(n,r) = \frac{n!}{r!(n-r)!}$$

$$= \frac{4!}{1!(4-1)!} = 4$$

$$n^r = 4^1 = 4$$

4

$$P(\{\neg T_2\})^{W_2^n} = 1 - P(\{T_2\})^{W_2^n} = 0.39 \quad (150)$$

Putting values in the equations: -

$$M_{1,2}(T_{1,2})^{W_i^n} = \frac{0.58 * 0.61}{0.58 * 0.61 + 0.42 * 0.39} \quad (151)$$

Therefore, $M_{1,2}(T_{1,2})^{W_i^n}$ = 0.53, is the weighted combined probability mass assigned to the 1st and 2nd threats by the 1st and 2nd intrusion detection systems using equation (144).

Comparisons of the combined probability masses of the proposed Multiple Simultaneous Threats Detection System (MSTDS) on both experimental data and public domain data sets are given below:-

Table 5.5: Comparisons of the Combined Probability Masses of MSTDS Experimental Data and Public Domain Data sets

Data Source	Combined Probability Mass		
	DS	Extended DS	GEP
MSTDS	0.16	0.52	0.62
Public Domain Data set	0.16	0.53	0.63

The results above clearly demonstrated the validity of the MSTDS, both on the simulated and public domain datasets. In addition, the following threat detection results derived from the public domain data set using the MSTDS are obtained. The DS, Extended DS and GEP methods were used to analyse the data.

Table 5.6: Threat Detection Rate using Dempster Shafer on Public Domain Data Sets

IDS	OT	OA	DA	RA	FPR	Detect Rate
MARS	1617	729	983	539	26.08	60.78
Sniffers	1047	603	666	285	52.63	63.63
Snoop	412	222	222	159	28.57	53.84
Wireshark	825	381	381	285	25	46.15
Total	3901	1935	2252	1269		

Table 5.7: Threat Detection Rate using Extended Dempster Shafer on Public Domain Data Sets

IDS	OT	OA	DA	RA	FPR	Detect Rate
MARS	1047	603	983	602	0.11	93.93
Sniffers	888	666	825	540	18.94	92.85
Snoop	698	285	444	254	11.11	63.63
Wireshark	539	317	381	285	10	70.58
Total	3172	1871	2632	1681		

Table 5.8: Threat Detection Rate using GEP on Public Domain Data Sets

IDS	OT	OA	DA	RA	FPR	Detect Rate
MARS	983	603	983	616	0	100
Sniffers	888	634	825	590	7.04	92.85
Snoop	476	299	444	302	0	93.33
Wireshark	571	335	539	268	19.90	94.44
Total	2918	1871	2791	1776		

5.4 Comparisons with Related Works

5.4.1 Description of the Related Works

Many researchers have worked on multisensor data fusion to detect threats ranging from single threats to multiple simultaneous threats. Most of the work has been done in the defence area. There is very little reported work in distributed systems like the UNIX environment. Multiple simultaneous threats detection is a new area of research, particularly for the UNIX environments. The main reason for this is that until recently, UNIX is considered a very safe and secured operating system. However, intruders developed new intrusion techniques, and the ability for intrusion detection systems running on distributed systems like UNIX has become questioned. By comparing the experimental results obtained in this research with those of other scientists, it provides a fair idea about the robustness, accuracy, efficiency and performance of the proposed multiple simultaneous threat detection system.

5.4.2 Results

5.4.2.1. Data Fusion Model Approaches

In this research, the data fusion model, called a “multiple simultaneous threats detection system”, is comprised of a hybrid model incorporating an intermediate level based on set cover, and a data fusion level using combination of IDS that use Dempster-Shafer, Extended Dempster-Shafer, and Generalised Evidential processing (GEP), respectively.

Existing data fusion models related to this research mainly applied classical inference, Bayesian inference, Dempster-Shafer, Extended Dempster-Shafer, Kalman Filter or Generalised Evidential processing (GEP) in their fusion model separately. Therefore, the proposed hybrid data fusion model can be considered a novel approach as it uses a combination of individual fusion models in a concerted manner.

The performances of inference theories such as Bayesian, Dempster-Shafer, Generalised Evidential Processing (GEP), Kalman Filter, and classical inferences have not been compared adequately in a systematic manner. However, different researchers have made efforts to compare their individual experiments or fusion technique models with others. For example, David L. Hall and Sonya [25] worked on exact inference and assigned belief versus time in their experiments for an object having four sensors to detect enemy aircraft or objects. They concluded that Bayesian inference had better results than Dempster-Shafer. David L. Hall and Sonya also concluded that Bayesian inference performed better in situations

involving a smaller number of floating points operations. In cases of more than four floating points operations, Dempster-Shafer would have performed better. They also reported that as the adversaries also had multiple ways to hide their identities, more research on the inference approach would be required for this field [21].

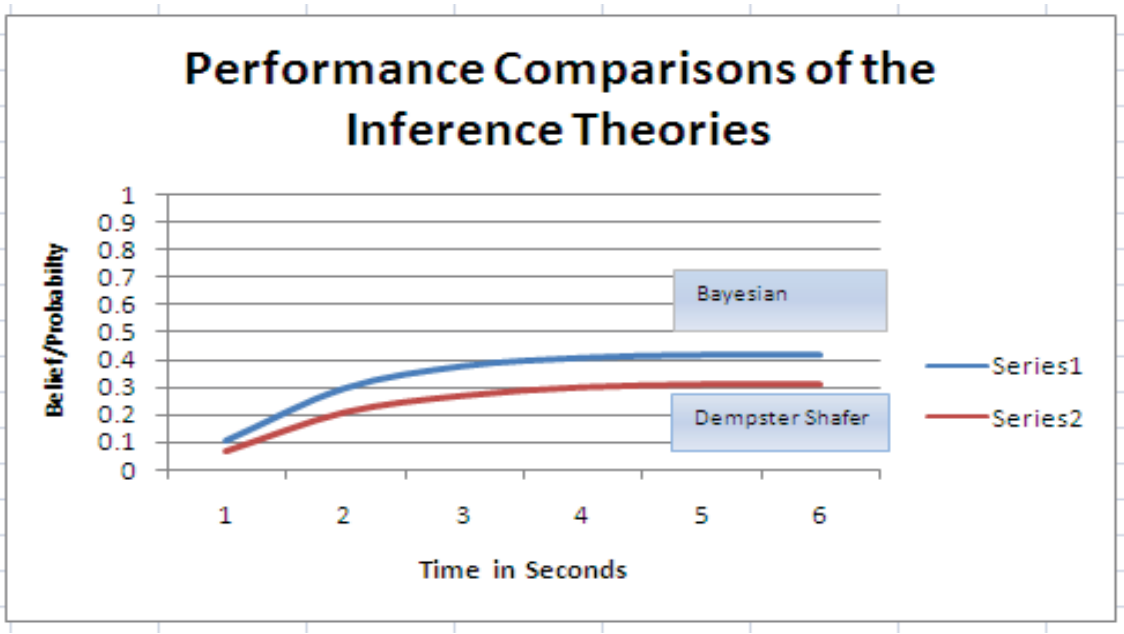

Fig 5.4: Performance Comparisons between Bayesian and Dempster-Shafer

5.4.2.2. Comparisons of the Performance of the Inference Theory

Here, the results of different data fusion models including the proposed hybrid model that was used to increase precision in multiple threats detection are compared. In the DARPA IDS project, the Hidden Colored Petri-Net correlation-based alert system was used. In order to increase the accuracy and threat detection precision, Dong and Deborah [28] used a hybrid model which

incorporated Dempster-Shafer and Extended Dempster-Shafer to combine beliefs in threat detection results. They found that their hybrid model resulted in a 19% improvement in precision of threat results compared to the performance of a system which used only the Dempster-Shafer model. They also discovered that a simultaneous threats detection system (using a hybrid of Bayesian and Dempster-Shafer showed a 12% (Ref; Fig 5.4) improvement in precision of threat detection compared to a hybrid of Dempster-Shafer and Extended Dempster-Shafer.

Christos and Basil [61] worked on DoS detection using Dempster-Shafer Theory, but no quantitative comparisons in their research reports were found, although they concluded that the model used in their research produced detection results with improved quantitative measurements of beliefs and plausibility.

Huadong Wu, Mel Siegel and Rainer [33] experimented on context sensing. They compared audio and video sensors based on the Dempster-Shafer and Extended Dempster-Shafer theory of inferences, and found that there was significant improvement in estimation. However, the results for Dempster-Shafer and Extended Dempster-Shafer alone were almost similar. They suggested that further research in this area would be necessary.

5.4.2.3. Decision Level Techniques

The multiple simultaneous threats detection system uses three different decision level techniques:

1) Dempster Unweighted

2) Dempster Weighted

3) Generalised Evidential Processing

4) In the experiments, each observer (intrusion detection system) assigned evidence (probability mass). The probability masses were combined using the above three mathematical inference approaches. Their results were compared, and it was discovered that the integration of the observers improved the precision of the threats detection significantly. Of the three techniques, GEP showed better precision in threat detection than either Dempster-Shafer or Extended Dempster-Shafer theories. The comparisons are presented in the following charts:

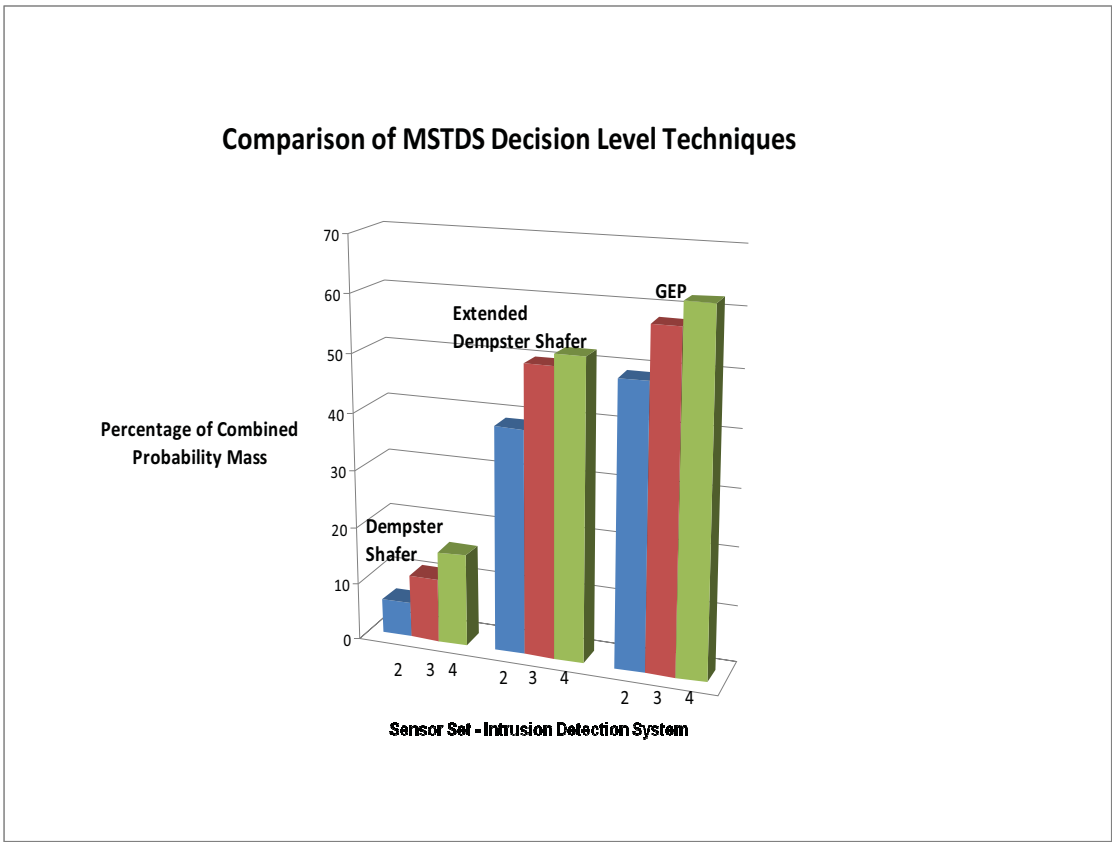

Fig 5.5: Comparison of MSTDS Decision Level Techniques

The average combined probability mass for threats detection by the three decision levels techniques, Dempster-Shafer, Extended Dempster-Shafer and GEP were 6%, 41% and 53%, respectively. This supported the conclusion that GEP was the most effective technique in threats detection by an average increase of 47 % in the combined probability mass.

5) Comparison of results between Dempster-Shafer (unweighted) and Extended Dempster-Shafer (weighted) revealed that Extended Dempster-Shafer provided better combined probability masses by an average of 35% compared to Dempster-Shafer. The following chart shows a comparison of the above two techniques.

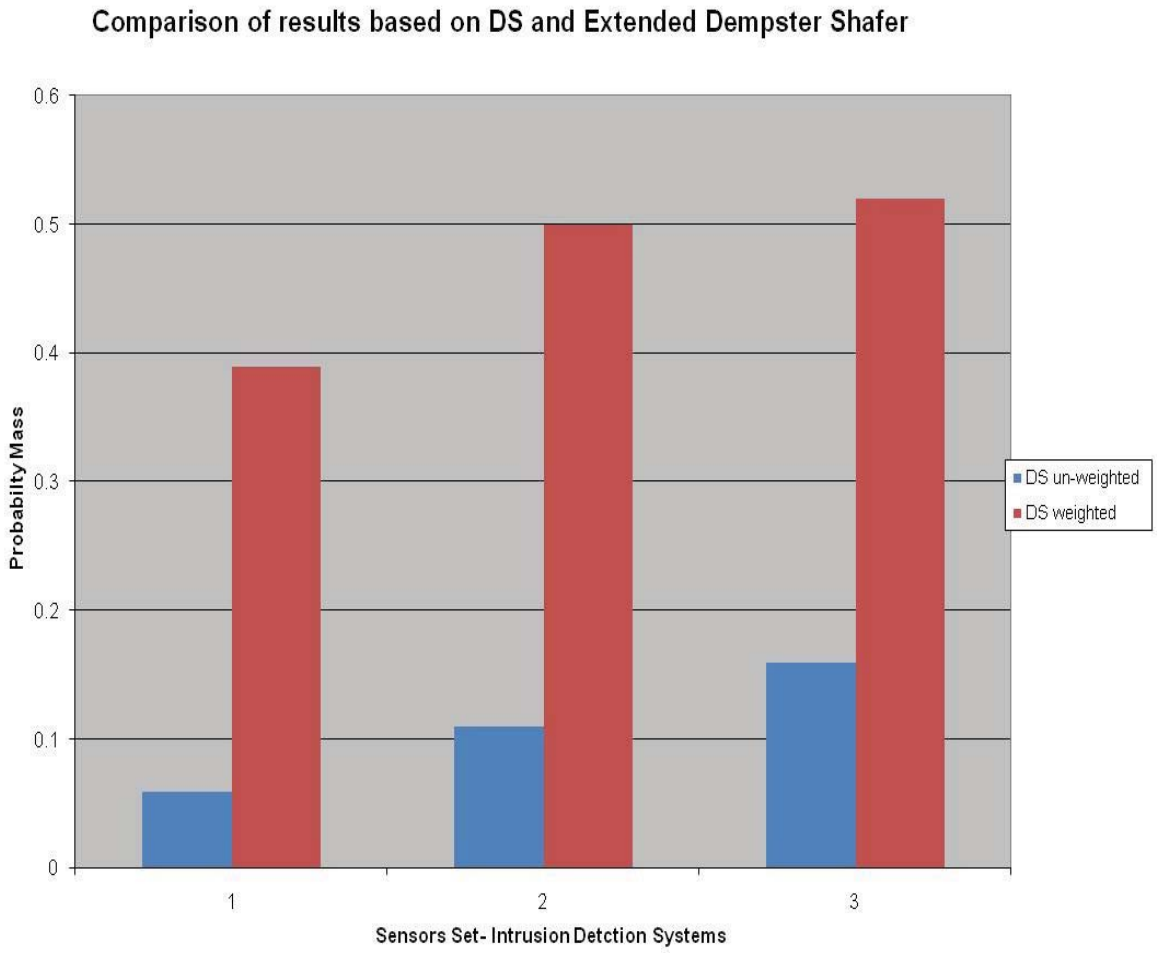

Fig 5.6: Results based on Dempster Shafer (unweighted) and Extended Dempster Shafer (weighted)

5.5 Discussion

Both Bayesian and Dempster-Shafer theories provide effective processing models in multisensor data fusion. However, their implementations involve overly complex iteration of the data fusion process in terms of the calculation of probability masses and weights. Therefore, for all practical purposes, it would be difficult to use them in combining probability masses efficiently when overlapping

and conflicting propositions occur, in the cases where there are more than four sensors. The greater the number of sensors, the greater the precision in threats detection is expected to be achieved.

In this thesis, experiments were conducted in three steps using evidences from two, three and then four sensors (intrusion detection systems) to detect the four types of threats. The sensors were the four intrusion detection systems. The experimental results confirmed that the combined results of the four sensors improved simultaneous threats detection significantly. Bayesian, Dempster-Shafer and GEP theories of inference, taken together, provided an effective tool to combine evidences of these sensors and measured the uncertainty of a hypothesis (i.e. potential threat).

The next set of graphs compare the efficiency of the Dempster-Shafer, Extended Dempster-Shafer and Generalised Evidential Processing data fusion techniques, Figures 5.5 to 5.7 show a significant increase in the combined probability masses for Extended Dempster-Shafer Theory and Generalised Evidential Processing. That is a good indication of enhanced precision, accuracy and better performance of GEP data fusion in threat detection over the Dempster-Shafer and Extended Dempster-Shafer data fusion techniques.

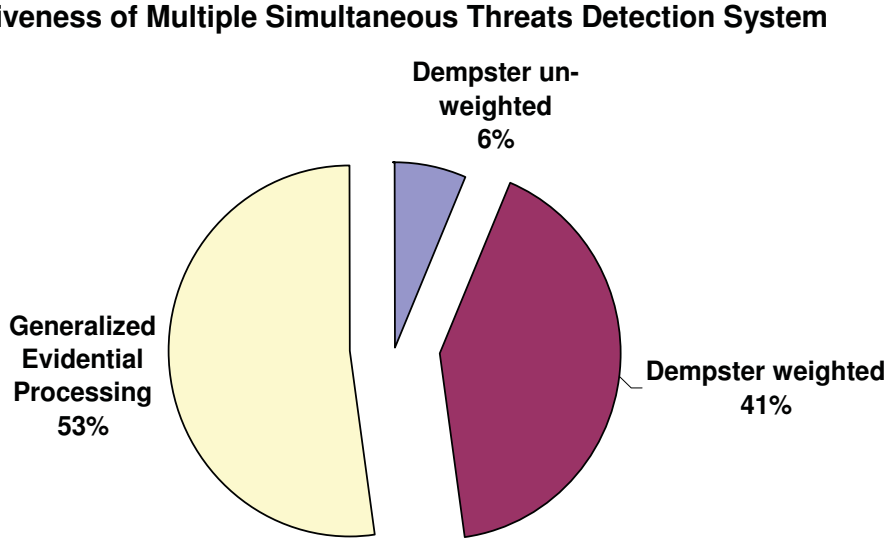

Fig 5.7: Effectiveness of the Multiple Simultaneous Threats Detection System

The average combined probability masses for threat detection by the three methods, Dempster-Shafer, Extended Dempster-Shafer and GEP were 6%, 41% and 53% respectively. This clearly demonstrates that GEP is the most effective method and it increased the combined probability mass by 47%. The general principle is that the higher the degree of confidence in measuring the probability mass, the greater the precision in threat detection.

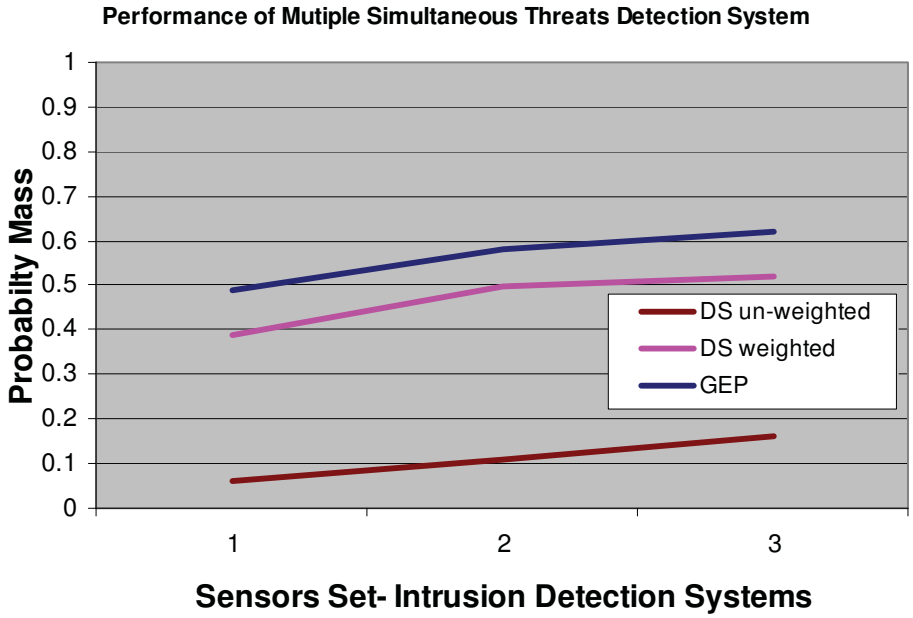

Fig 5.8: Performance of the Multiple Simultaneous Threats Detection Systems

Figure 5.8 shows another view of the relative performances of three intrusion detection systems. GEP, once again, is shown to be an efficient data fusion tool in threats detection.

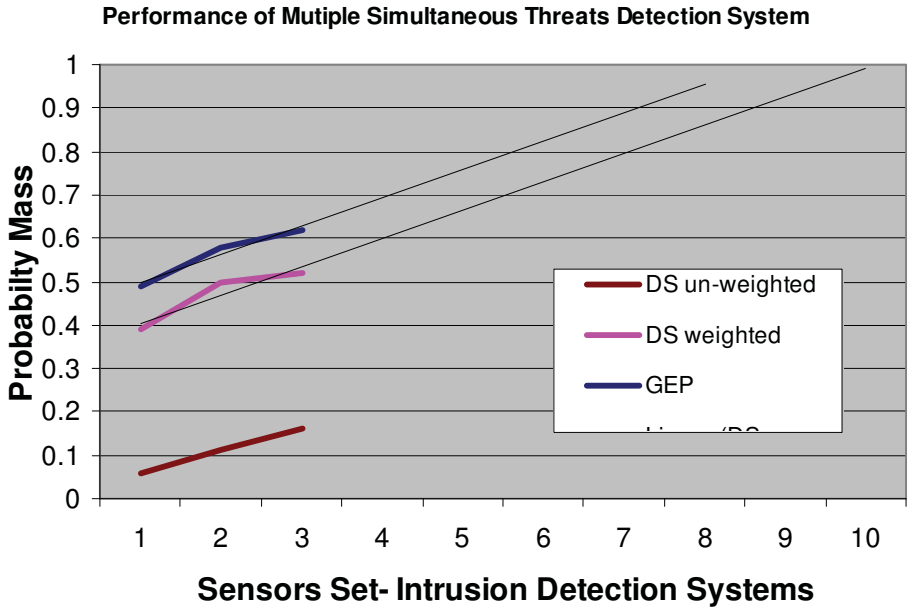

Fig 5.9: Performance Trend of the Multiple Simultaneous Threats Detection System

Figure 5.9 shows the trend line of the performance in threats detection that could involve more than four intrusion detection systems. The higher the number of participating sensors in threat detection, the greater will be the precision in the results. However, the calculations and flow of data become very complicated when dealing with more than four sensors and with multiple hypotheses.

Finally, Figure 5.10 shows the overall performance comparison of intrusion detection systems.

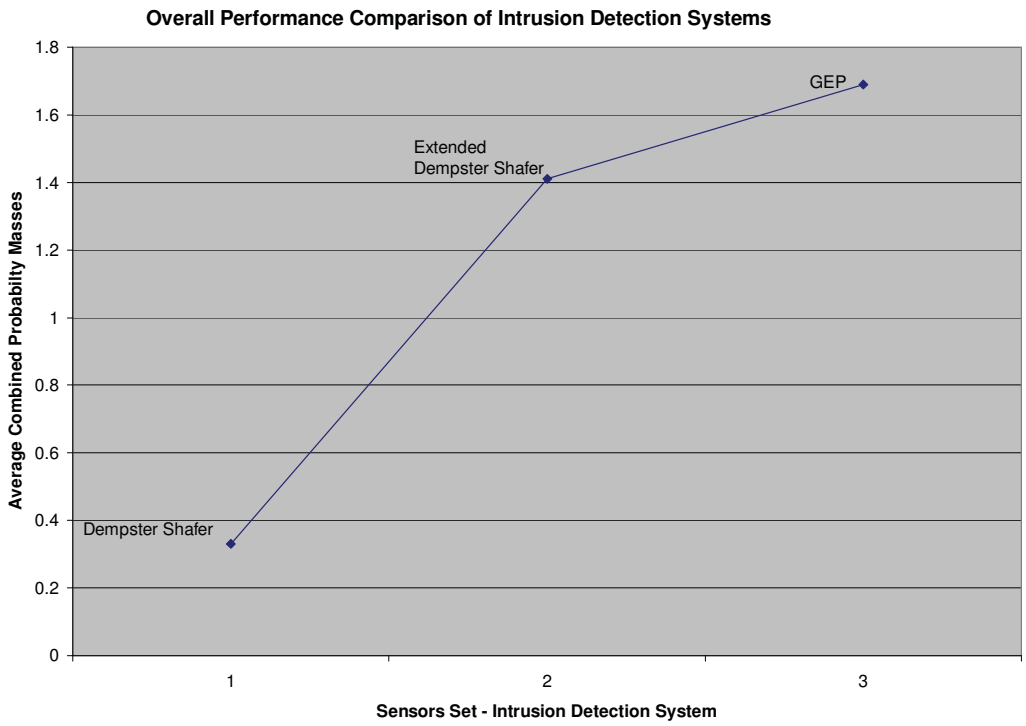

Fig 5.10: Overall Performance comparisons of the IDS methods

Figure 5.10 shows the average combined probability mass of the three intrusion detection systems of the proposed multiple simultaneous threats detection system used in this research. GEP demonstrates a clear superiority in threats detection over the Dempster-Shafer and Extended Dempster-Shafer Data fusion techniques.

5.5.1 Limitations of the Proposed MSTD Model

1. As the experiment was conducted in a controlled environment, there was no discussion of consonant, consistent and disjoint evidence assignments in the

threat results. In fact, this was one of the major advantages of applying the set cover theory.

2. The proposed MSTDS was developed using a combination of Dempster-Shafer, Extended-Dempster Shafer, and GEP inference approaches. Because the focus was on the mathematical components of these technologies, details on the following methods applied in this research might not be adequately described. These include:

    a. Set cover theory could be implemented in a number of different ways. Discussion on the implementation aspects was beyond the scope of this thesis.

    b. The Dempster-Shafer theory has been modified by various researchers who have developed the following probability combining rules and evidential processing methods: [73]

        - The Discount and Combine method
        - Yager's modified Dempster's rule
        - Inagaki's unified combination rule
        - Zhang's centre combination rule
        - Dubois and Prade's disjunctive consensus rule
        - Mixing or averaging
        - Convolutive X-averaging
        - Smets' rule

- The qualitative combination rule
- DST
- Yen's rule
- Envelope imposition, and horizontal x-averaging

c. Similarly, Generalised Evidential Processing (GEP) has different rules for combining probability masses.

d. The calculations of probability mass and weights for more than four simultaneous threats using the Dempster Shafer theory become a highly complex process. Fortunately, GEP does not impose such restrictions, although GEP is an extension of the Dempster Shafer and Bayesian inference approaches. As a result, there is a need in future to conduct research to test the detection of more than 4 simultaneous threats using GEP and other inference techniques that do not exhibit the above limitations.

6. Conclusions

In this thesis, a novel multiple simultaneous threats detection system (MSTDS) is proposed. It is based on a hybrid multisensor data fusion model that involves a set cover based intermediate filtering layer, followed by a data fusion layer comprising four independent intrusion detection systems (IDS). Each of these IDS runs an intrusion detection engine based on one of the component data fusion models. The component data fusion models include Bayesian, Dempster-Shafer, Extended Dempster-Shafer, and Generalised Evidential Processing (GEP). Through empirical experimentation, it was confirmed that the MSTDS significantly increased the precision of threats detection. In particularly, Dempster-Shafer inference produced a 56% detection rate while Extended Dempster-Shafer and GEP had 80% and 95% detection rates, respectively. On average, the proposed MSTDS increased the detection rate by 39%, which was an increase from 56% to 95% accuracy. The false positive rate also went down from 33% to 6%. Here, the detection rate was calculated by dividing the number of detected alerts by the number of observed alerts, while the false positive rate was obtained by dividing the number of real alerts by the number observed alerts, respectively. In summary, there was a net improvement of 27% in decreasing the number of false positive alarms, which is significant for practical instrusion detection in real time and distributed systems environment like UNIX (cf. Tables 5.5 and 5.8).

Through experimentation, it was also discovered that GEP achieved better performance than both Dempster-Shafer and Extended Dempster-Shafer based on the values of combined evidential/probability masses returned by each of the four intrusion detection systems in the data fusion layer. The combined probability mass of the GEP was 0.56, while for Dempster-Shafer and Extended Dempster-Shafer it was 0.11 and 0.47, respectively. GEP increased the combined probability mass by 45%, which in turn increased the overall efficiency of the proposed MSTDS (cf. Tables 5.5 and 5.8).

Set cover used as the middle-tier layer for filtering threats data reduces the amount of redundancy while providing better groupings of types of threats for subsequent processing. By obtaining pair-disjoint subsets of threats in this layer, excessive computation both in terms of time and CPU cycles downstream could be avoided. Particularly, set cover reduced the threats data (from 2,274 to 295 substrings) to a level where it became possible to detect more than two simultaneous threats with less computational effort, a performance that would be almost impossible with existing threat detection approaches that are based on simply using Bayesian and Dempster-Shafer approaches. Set cover was also capable of determining the cost effectiveness when selecting a computer node among the set of individual IDS for processing the subsets. Thus, it played a critical role in improving the precision of detection while simultaneously minimising the amount of numerical calculations for the overall MSTDS.

6.1 Future Work

1) In this thesis, data fusion approaches that are based on Dempster-Shafer, Extended Dempster-Shafer and Generalized Evidential Processing (GEP) have been studied and applied. However, other data fusion techniques including heuristic-based methods, kinematic and attributive techniques, knowledge-based approaches might further improve the precision and efficiency of detection using multiple simultaneous threats detection system. Understanding the relative merits of these less explored methods in multisensor data fusion will be one direction of future work [25].

2) Related to data fusion is the essential task of calculating the combined probability mass based on inputs coming from the component IDS involved in the MSTDS. There are varied ways of combining these individual probabilities, many of which have not been used and compared before in multiple simultaneous threats detection research. These methods include: [73]

    - The discount and combine method
    - Yager's modified Dempster's rule
    - Inagaki's unified combination rule
    - Zhang's centre combination rule
    - Dubois and Prade's disjunctive consensus rule
    - mixing or averaging
    - convolutive x-averaging

- Smets' rule
- the qualitative combination rule
- DST
- Yen's rule
- Envelope imposition, and horizontal x-averaging

3) While an empirical experimental setup was used in this research, large scale evaluation of the proposed multiple simultaneous threat detection system in operational scenario will be a major focus of future research.

4) In this thesis, up to four intrusion detection systems – MARS, Sniffers, Snoop and Wireshark, were used as collectors of threat data. Overcoming this restriction by considering beyond four intrusion detection systems and researching advanced techniques for controlling the increase in computation will be another focus of future work. To achieve this, future results will be compared with the performance trend line illustrated in Figure 5.9 of Section 5.4 on Discussions.

5) While set cover theory was applied for filtering threats data in the middle tier prior to multisensor data fusion, other techniques like the Kalman Filter could also be applicable. Studying and comparing the pros and cons of other filtering approaches for the middle tier constitutes another area of future research.

6) The calculation of more than four simultaneous threats' probability masses and weights by using the Dempster Shafer theory could become highly

complex. Generalised Evidential Processing (GEP) did not have to impose such restrictions. In future work, research will be carried out to study the detection of more than 4 simultaneous threats using GEP and other inference engines that do not have the above restriction. Appropriate modification of the MSTDS architecture might be necessary in solutions to this restriction.

7) In order to further improve the precision of simultaneous threats detection, a weighted version of the Generalised Evidential Processing (GEP) model will be studied.

Bibliography

1. Azzedine Bendjebbour, Yves Delignon, et al., 2001, 'Multisensor Image Segmentation Using Dempster-Shafer Fusion in Markov Fields Context', IEEE Transaction on GeoScience and Remote Sensing, Volume 39 Issue 8, August page(s) 1789-1798.
2. Ahsan Habib, Hefeeda Mohamed and Bharat Bhargava, 2003, 'Detecting service violations and DoS attacks'. In NDSS Conference Proceedings. Internet Society, page(s) 177-189.
3. Uwe Aickelin, 2002, 'An Indirect Genetic Algorithm for Set Covering Problems', Journal of the Operational Research Society, 53(10), page(s) 1118-1126.
4. Alexei Makarenko and Hugh F. Durrant-Whyte, 2006, 'Decentralized Bayesian algorithms for active sensor networks'. Information Fusion 7(4), page(s) 418-433.
5. Ambareen Siraj et al. , 2004, 'Intrusion Sensor Data Fusion in an Intelligent Intrusion Detection System Architecture', Proc. of International Conference on System Sciences, page(s) 281-306.
6. Anind K. Dey, 2000, 'Providing Architecture Support for Building Context-Aware Applications', PhD thesis, November 2000, Georgia Institute of Technology, Page(s) 1-170.
7. Midori Asaka, Yasujiro Taguchi and Tatsutoshi Goto, 1999. 'Local Attack Detection and Intrusion Route Tracing', IEICE Trans. on Commun. Volume E-82-B, Issue11, page(s) 1826-1833.
8. Ben Grocholsky, Alexei Makarenko and Hugh F. Durrant-Whyte, 2003, 'Information-theoretic coordinated control of multiple sensor platforms', ICRA 2003, page(s) 1521-1526.
9. J. Braun 2000, 'Dempster-Shafer theory and Bayesian reasoning in

multisensor data fusion', Sensor Fusion: Architectures, Algorithms and Applications IV; Proceedings of SPIE 4051, page(s) 255–266.

10. J. Burroughs , F. Wilson and V. Cybenko, 2002, 'Analysis of Distributed Intrusion Detection Systems Using Bayesian Methods', Proc. of the Performance, Computing, and Communications Conference, page(s) 329-334.

11. CERT/CC advisory w32/blaster worm, Aug. 2003. Accessed on 12 Feb 2009.

12. CISCO. Using CAR during DoS attacks. http://www.cisco.com/warp/public/63/car_rate_limit_icmp.html. Accessed on 12 Feb 2009, page(s) 439-446.

13. Cory F. Cohen, 2002, CERT advisory CA-2002-17 Apache web server chunk handling vulnerability, http://www.cert.org/advisories/CA-2002-17.html, page(s) 298-307.

14. Cory F. Cohen, 2002, CERT advisory CA-2002-18 'Open SSH vulnerabilities in challenge response handling', http://www.cert.org/advisories/CA-2002-18.html. Accessed 24 July 2002, page(s) 231-242.

15. Computer Emergency Response Team (CERT), 1997, CERT advisory CA-1992-09 'AI anonymous FTP vulnerability', September 1997. http://www.cert.org/advisories/CA-1992-09.html. Accessed on 13 August 2002.

16. Computer Emergency Response Team (CERT), 2001, CERT advisory CA-2001-35 'Recent activity against secure shell daemons', December 2001. http://www.cert.org/advisories/CA-2001-35.html. Accessed on 24 July 2002.

17. David P. Williamson, IBM Research Report. 'Lecture Notes on Approximation Algorithms', Fall 1998, page(s) 7-18.

18. Computer Security Institute, 2002, 'Cyber crime bleeds U.S. corporations, survey shows', April. 2002. http://www.gocsi.com/press/20020407.html. Accessed on 16 January 2003.

19. M. Cooper and M. Miller, 1998, 'Information gain in object recognition via sensor fusion', Proc. of the International Conference on Multisource-Multisensor Information Fusion (Fusion '98), volume 1, page(s) 143–148.

20. F. Cuppens et al., 2002, 'Correlation in an Intrusion Process', Internet Security Communication Workshop (SECI'02), page(s) 1-20.

21. David Hall, 1992, *Mathematical Techniques in Multisensor Data Fusion*. Artech House, Norwood, Massachusetts. ACMDL.

22. Dong Yu and Deborah Frincke, 2004, 'A Novel Framework for Alert Correlation and Understanding', Lecture Notes in Computer Science, volume 3089, Proc. of International Conference on Applied Cryptography and Network Security (ACNS), page(s) 452-466.

23. Daniel Palmer et al., 2005, 'Swarm Reasoning', Proc. of Swarm Intelligence Symposium, IEEE, page(s) 294–301.

24. Danyliw Roman and Allen Householder 2001, CERT advisory CA-2001-19 'Code Red' worm exploiting buffer overflow in IIS indexing service DLL, August 2001. http://www.cert.org/advisories/CA-2001-19.html. Accessed on 13 August 2002.

25. David L. Hall and Sonya A. H. McMullen, 2004, *Mathematical Techniques in Multisensor Data Fusion* Second Edition, ISBN 1-58053-335-3, 2004 RTECH House, INC, MA 02062.

26. Diego Zamboni, 1996, 'A security analysis integration tool', Proc. of the Systems Administration, Networking and Security Conference, Washington, D.C., May 1996, page(s) 1-13.

27. Don Koks and Subhash Challa, 2005, An Introduction to Bayesian and Dempster-Shafer Data Fusion, DSTO Systems Sciences Laboratory, page(s) 1-52.

28. Dong Yu and Deborah Frincke, 2005, 'Alert Confidence Fusion in Intrusion Detection Systems with Extended Dempster-Shafer Theory' in The 43rd Annual ACM Southeast Conference 2005, page(s) 142-147.

29. Edward Waltz and James Llinas, 1990, 'Multisensor Data Fusion'. Boston: Artech House. Artech House, Boston, page(s) 1-464.

30. Enrique H. Ruspini, John D. Lawrance and Thomas M. Strat, 1990, 'Understanding Evidence Reasoning', Technical Note 501, December 1990, Artificial Intelligent Center, Computer and Engineering Sciences Division, SRI International, Menlo Park, California 94025, USA.

31. Ferguson and Senie, 2000, 'RFC2827 network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing', ACMDL.

32. Gottlob Frege, 2005, 'On Sense and Reference'. http://wikisource.org/wiki/On Sense and Reference. Accessed on Sep 21, 2009.

33. Huadong Wu, Mel. Siegel, Rainer Stiefelhagen and Jie Yang, 2002, 'Sensor fusion using Dempster-Shafer theory'. Proc. of IEEE Instrumentation and Measurement Technology Conference, Anchorage, AK, USA, page(s) 1-6.

34. D. Hall and A. Garga, 1999, 'Pitfalls in data fusion (and how to avoid them)', Proc. of the Second International Conference on Information Fusion (Fusion '99), volume 1, page(s) 429–436.

35. Hervaldo S. Carvalho, Wendi B. Heinzelman, Amy L. Murphy, J. Claudionor and N. Coelho, Proc. of the Sixth International Conference on Information Fusion 'General Data Fusion Architecture', July 11, 2003, page(s) 1465-1472.

36. http://www.cert.org/advisories/CA-2003-20.html. Accessed on 25 June 2008.

37. http://www.cs.sunysb.edu/~algorith/files/set-cover.shtml. Accessed on 09 Aug 2008.

38. http://www-math.mit.edu/~goemans/18434/setcover-tamara.pdf. Accessed on 22 September 2007.

39. Hugh F. Durrant-Whyte, 2006, 'Data fusion in sensor networks'. Proc. of Advanced Video and Signal Based Surveillance, page(s) 29-39.

40. Ingham Kenneth and Stephanie Forrest, 2002, 'A History and Survey of Network Firewalls' ACMDL, page(s) 1-42.

41. Daniel Burroughs, Linda. F. Wilson and George V., 2002, 'Analysis of Distributed Intrusion Detection Systems Using Bayesian Methods', IPCCC 2002, April 2002, page(s) 329-334.

42. Joanna Kulik, Wendi Heinzelman and Harry Balakrishnan, 2002, 'Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks', Wireless Networks, Volume 8, 2002, page(s) 169-185.

43. James R. Boston, 2000, 'A Signal Detection System Based on Dempster-Shafer Theory and Comparison to Fuzzy Detection', IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Volume 30, Issue 1, page(s) 45-51.

44. J. C. Bezdek and N. R. Pal, 1993, Fuzzification of self organizing feature map, Proc. Application of Fuzzy Logic Technology, SPIE Volume 2061, Boston, USA, page(s) 142-162.

45. Jeff Hawkins, 2004, *On Intelligence*, New York: Times Books.

46. Jie Yang and Alex Waibel, 1996, 'A Real-Time Face Tracker', Proceedings of WACV, page(s) 142-147.

47. Kevin Tomsovic and Banister Baer, 1998, 'Fuzzy information approaches to equipment condition monitoring and diagnosis', Electric Power Applications of Fuzzy Systems, IEEE Press, page(s) 59–84.

48. Kapil Kumar Gupta, 2010, Intrusion Detection in Networks and Applications, World Scientific, Singapore, volume 7, issue 1, pages 35-49.

49. M. Kokar, M. Bedworth and C. Frankel, 2000, 'A reference model for data fusion systems', Sensor Fusion: Architectures, Algorithms and Applications IV; Proceedings of SPIE 4051, page(s) 191–202.

50. M. L. Krieg, 2003. 'A Bayesian belief network approach to multi-sensor kinematic and attribute tracking', In Proc. of the Sixth International Conference on Information Fusion 'General Data Fusion Architecture', July 11, 2003, page(s) 17-24.

51. Lawrence A. Klein, 1999, 'Sensor and Data Fusion Concepts and Applications' (second edition), SPIE Optical Engineering Press, ISBN 0-8194-3231-8.

52. W. Lee W and W. Fan et al., 2002, 'Toward Cost-Sensitive Modelling for Intrusion Detection and Response', Journal of Computer Security, Volume 10, Numbers 1, page(s) 5-22.

53. Ma Bing, 2001 'Parametric and Non Parametric Approaches for Multisensor Data Fusion', PhD thesis, University Of Michigan, page(s) 1-212.

54. D. Mituzas, 2002, 'Apache worm in the wild', June 2002. Posted on the BUGTRAQ mailing list. http://online.securityfocus.com/archive/1/279529. Accessed on July 24 2002.

55. P. Mockapetris, 1987, 'Domain names—concepts and facilities', Nov. 1987. RFC 1034. ftp://ftp.isi.edu/in-notes/rfc1034.txt. Accessed on March 20 2003.

56. Nathalie Francois, 2000, 'A New Advanced Multitechnique Data Fusion Algorithm for NDT', http://www.ndt.net/article/wcndt00/papers/idn316/idn316.html. Accessed on 23 May 2009.

57. P. Ning, D. Xu, C. Healey and R. Amant, 2004, 'Building Attack Scenarios through Integration of Complementary Alert Correlation Methods', The 11th Annual Network and Distributed System Security Symposium, page(s) 97-111.

58. Rainer Stiefelhagen, Jie Yang and Alex Waibel, 2001, 'Estimating Focus of Attention Based on Gaze and Sound', Proc. of Workshop on Perceptive User Interfaces PUI 2001, Orlando, Florida, USA, page(s) 1-138.

59. R. Rehman, 2003, 'Intrusion Detection System with SNORT', http://www.snort.org/. Accessed on Aug 18 2009.

60. S Terry Brugger, 2004, 'Data Mining for Network Intrusion Detection' – p.8/55 www.bruggerink.com/~zow/papers/dmnid_qualpres.pdf. Accessed on Jan 12 2010.

61. Siaterlis Christos and Maglaris Basil, 2004, 'Towards Multisensor Data Fusion for DoS detection', Proc. of the 2004 ACM symposium on Applied Computing, page(s) 1-8.

62. E. H. Spafford, 1989, 'The internet worm incident'. In ESEC '89. 2nd European Software Engineering Conference Proceedings, 11-15 Sept. 1989, Coventry, UK (Berlin, West Germany, West Germany, 1989), C. Ghezzi and J. McDermid, Eds., Springer-Verlag, page(s) 446–68.

63. E. H. Spafford, 1989, The Internet worm incident', Tech. Rep. Purdue Technical Report CSD-TR-933, Department of Computer Science, Purdue University, West Lafayette, IN 47907-2004, 1991. LEM OS, R. Counting the cost of slammer, Jan. 2003. http://news.com.com/2100-1001-982955.html?tag=mainstry. Accessed on 10 February 2003.

64. Tim Bass, 2000, 'Intrusion detection systems and multisensor data fusion', Communications of the ACM, volume 43, Issue 4, page(s) 99-105.

65. Tim Bass and Dave Gruber, 2005, 'A glimpse into the future of id'. Usenix. 18 Aug 2005. http://www.usenix.org/publications/login/1999-9/features/future.html. Accessed on Oct 27 2009.

66. Tim Bass et al., 1999, 'E-Mail Bombs and Countermeasures, Cyber Attacks on Availability and Brand Integrity', IEEE Network Magazine, volume 12, Issue 2, page(s) 10-17.

67. V. Chatzigiannakis, A. Lenis, C. Siaterlis, M. Grammatikou, D. Kalogeras, S. Papavassiliou and V. Maglaris, 2002, 'Distributed Network Monitoring and anomaly Detection as a Grid Application' published in www.gridcc.org.

68. Vladimir I. Gorodetski, Oleg Karsayev, Igor V. Kotenko and Alexey Khabalov, 2002, 'Software Development Kit for Multi-agent Systems Design and Implementation'. In B.Dunin-Keplicz and E.Nawareski (Eds.) 'From Theory to Practice in Multi-agent Systems'. Lecture Notes in Artificial Intelligence, volume 2296, Springer Verlag, page(s) 121-130.

69. Vitorino Ramos and Ajith Abraham, 2004, 'ANTIDS Self-Organized

Ant-Based Clustering Model for Intrusion Detection System', WSTST 2005, page(s) 977-986.

70. A. Wespi et al., 2004, 'Recent Advances in Intrusion Detection', Proc. of the Symposium on Recent Advances in Intrusion Detection (RAID 1999), page(s) 234-243.

71. Xiao-gang Wang, Wen-han Qian; E. Pagello and Pei Ren-qing, 1996, 'Multisensor Fusion and Integration for Intelligent Systems' IEEE/SICE/RSJ International Conference on Volume 35 , Issue 1 , 8-11 Dec 1996, page(s) 166–173.

72. http://en.wikipedia.org/wiki/Principle_of_maximum_entropy, Accessed on 4 October 2009.

73. Kari Sentz and Scott Ferson, 2002, 'Combination of Evidence in Dempster-Shafer Theory', Systems Science and Industrial Engineering Department, Thomas J. Watson School of Engineering and Applied Science, Binghamton University, NY, page(s) 1-94.