Alert Correlation with Abstract Incident Modeling in a Multi-Sensor Environment[1]

Ambareen Siraj[†] and Rayford B. Vaughn[††],

Department of Computer Science
Tennessee Tech University

Department of Computer Science and Engineering
Mississippi State University

Summary
In response to proliferated attacks on enterprise systems today, many practitioners employ multiple, diverse sensors for increased information assurance because a single sensor cannot detect all types of attacks. A multi-sensor environment is characterized by deployment of a homogeneous and/or heterogeneous suite of sensors to monitor different entities in the corresponding environment. These multiple sensors may employ different strategies based on the model they use, the data source they monitor and the techniques they employ. Essentially, the primary advantage of using multiple sensors is to improve the detection rate and the coverage within the system. In multi-sensor environments, the sensors can collaborate with or complement each other to provide increased assurance of information. Although it makes good engineering sense to employ multiple sensors in a secure environment, however, managing data from these sensors is critically important. In this paper, we address the alert correlation aspect of sensor alert fusion in a multi-sensor environment. Here we describe the use of a causal knowledge-based inference technique with Fuzzy Cognitive Modeling to discover causal relationships in sensor data.

Key words:
Alert correlation, sensor alert fusion, fuzzy cognitive modeling, network security.

1. Introduction

In response to proliferated attacks on enterprise systems today, many practitioners employ multiple, diverse sensors for increased information assurance because a single sensor cannot detect all types of attacks. A *multi-sensor environment* is characterized by deployment of a homogeneous and/or heterogeneous suite of sensors to monitor different entities in the corresponding environment. These multiple sensors may employ different strategies based on the model they use, the data source they monitor and the techniques they employ. Essentially, the primary advantage of using multiple sensors is to improve the detection rate and the coverage within the system. In multi-sensor environments, the sensors can collaborate with or complement each other to provide increased assurance of information.

Although it makes good engineering sense to employ multiple sensors in a secure environment, however, managing data from these sensors is critically important as the workload for the security administrator is increased in many folds and large alert volume from different sensors can potentially overwhelm the security administrator - making analysis of such alerts extremely difficult. These factors necessitate fusion of the sensor alerts to provide sophisticated reasoning capabilities outside the sensors' core functions. Potential advantages of sensor alert fusion in a single or multi-sensor environment include elimination or reduction of the need for manual analysis of reported data; compression or reduction of alert volume; and identification of context by associating alerts from different sensors.

In this paper, we address the alert correlation aspect of sensor alert fusion in a multi-sensor environment. Here we describe the use of a causal knowledge-based inference technique with *Fuzzy Cognitive Modeling* to discover causal relationships in sensor data. The following sections will provide necessary background information, outline our technical approach, report on experimental results on a benchmark dataset and lastly conclude.

[1] This work was supported by NSF Cyber Trust Program Grant No: SCI-0430354, NSA IASP Grant No: H98230-04-1-0205, Office of Naval Research Grant number N00014-01-1-0678, and the Department of Computer Science and Engineering, Center for Computer Security Research at Mississippi State University. Parts of this work have appeared in Proceedings: *IEEE International Conference on Intelligence and Security Informatics, 2005.*

Manuscript received August 5, 2007
Manuscript revised August 20, 2007

2. Related Background

Research in the area of alert fusion/alert aggregation/alert clustering/alert correlation has emerged in last few years and primarily concerns information modeling and high level reasoning. Among them, the ones that are most relevant to our work are the following [16].

Julisch introduces attribute generalization in alarm (i.e., alert) clustering as a method to support *root cause discovery* [4]. This work outlines a semi-automatic approach for reducing false positives in alarms by identifying the root causes with clustering of alerts by abstraction and then eliminating the root causes to reduce alarm overload. Ning et al. proposes an alert correlation model based on prerequisites and consequences of intrusion [9]. With knowledge of prerequisites and consequences, the correlation model can correlate related alerts by matching the consequences of previous alerts with prerequisites of later ones and then *hyper alert correlation graphs* are used to represent the alerts. In the prerequisite-consequence model, the authors conduct reasoning with predicate logic where predicates are used as basic constructs to represent the prerequisites and consequences of attacks. This approach requires extensive modeling of attacks in terms of specifying prerequisite and consequence of each alert type in the sensor report. Yu and Frincke propose a model for *Alert Correlation and Understanding (ACU)* based on *Hidden Colored Petri-Nets (HPCN)* [18]. HPCNs model agents, resources, actions, and functions of a system with transition and observation probabilities. To perform correlation, a model based on prerequisites and consequences is generated using domain knowledge. Training of model is required to best fit the model parameters. Qin and Lee generate high level aggregated alerts from low level sensor data and then conduct causal analysis based on a statistical technique, known as the *Granger Causality Test*, to discover new patterns of attack relationships [13]. Although this approach does not require apriori knowledge of attacks behavior, it still requires some human intervention in background alert identification used in the statistical technique employed.

Fuzzy Cognitive Maps (FCMs) originated from the combination and synergism of fuzzy logic and neural networks. Researchers have used FCMs for many tasks in several different domains. Use of FCMs was first reported in our previous work [15] for fusing alert information in an intrusion detection environment to assess network health. Fuzzy Intrusion Recognition Engine, a network based IDS, also use FCMs in detecting attacks from features extracted from network traffic [17].

In this research, a new alert correlation technique has been described that uses fuzzy cognitive modeling with generalization to correlate alerts that are linked in multi-staged attacks. We have developed an abstract incident model for alert correlation with generalized security events to deal with scalability issues in sensor fusion. By focusing on the effects of the intrusions, such an abstract incident model captures the essence of typical or commonly occurring techniques used the attackers in multi-staged attacks and correlates alerts, even though intermediate alerts are missing in the sensor reports.

3. Alert Correlation with Abstract Incident Model

Alert correlation involves discovering causal relationships between alerts such that alerts that are associated in multi-staged attacks can be linked together. With the premise that every cause is bound to have an effect – whether the effect is critical or non-critical, we view the alerts generated by the sensors as causes with the potential to generate various impacts or effects in systems. Different alerts in sensor reports relate to different actions of the attackers which may have different objectives [16]. The effects generated can potentially be coupled together in a causal chain to reveal the possible correlations between the alerts that initiate them.

We use cognitive modeling with *Fuzzy cognitive maps (FCM)s* to represent these different cause and events in the system and the nature of relationships between them. *Fuzzy cognitive modeling* [16] offers a straightforward structural representation of causal knowledge and allows what-if kinds of reasoning for causal analysis of data. Proposed by Kosko, FCMs model the world as concepts and causal relations between concepts in a structured collection [5,6,7]. *Concepts* (nodes) in an FCM are events that originate in the system and whose values change over time. The causality links between concepts are represented by directed *edges* that denote how much one concept impacts the other(s). The concepts, as well as the edges, in the FCMs can be crisp or fuzzy. For a detailed description of FCMs and their workings readers are referred to [1,5,6,7].

For alert correlation, we employ *abstract incident modeling* where reasoning is based upon generalized events rather than specific/exact events in the environment. Problems with specific/exact knowledge modeling have been discussed in [16]. Such abstract incident modeling with generalized events captures the essence of typical or commonly occurring techniques used by the attackers in

multi-staged attacks focusing on the effects of the intrusions.

The abstract incident model shown in Figure 1. that show how different attacks in a system facilitate other attacks - all being part of coordinated multi-staged attacks. In an abstract incident model, an event can cause other events to occur or it can occur because of other events occurring in the system. Primarily, there can be two types of events activated in the system: *Cause Events (CEvents)* and *Effect Events (EEvents)*. The difference between the two types of events is that, as the names reflect, *CEvents* essentially contribute to *EEvents* or *EEvents* are activated by *CEvents*. The events in the abstract incident model of Figure 1. are described below:

- The events at far most left of the model (colored blue) are considered *CEvents*, which are generated as a result of alerts seen in the sensor reports and correspond to possible actions taken by the intruder to achieve some goal.
- The events at middle of the model (colored yellow) are considered:
  - *EEvents*, when they are generated as combined effects of the *CEvents* that correspond to the sensor alerts and the *CEvents* that correspond to existing risks in systems. *EEvents* are security incidents indicating a possible security violation in the system.
  - *CEvents* when they contribute to generation of risks of security incidents.
- The events at far right of the model (colored green) are considered:
  - *EEvents*, when they are generated as effect of the *CEvents* that correspond to some security incidents that have occurred in the system. Although not shown here, other external factors like, vulnerabilities or threats can also contribute to the activation of these events.

  CEvents, when as risks they contribute to generate other security incidents in the system.

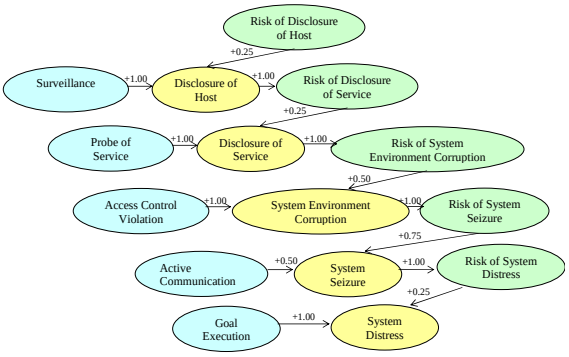

Fig. 1 An Abstract FCM Incident Model for Multi-Staged Attacks in General

The leftmost *CEvents* in the abstract incident model of Figure 1. correspond to the generalized alert types in the attack name generalization hierarchy of Figure 2., which shows how specific attack names are generalized into different abstract categories of attacks at different abstraction levels. Some of the abstract concepts used in this hierarchy are adapted from [12]. In this generalization hierarchy, commonality of alerts is considered based on the nature of impact of the attacks that generate the alerts, i.e., it focuses on what the attacker achieves by executing the attacks on systems.

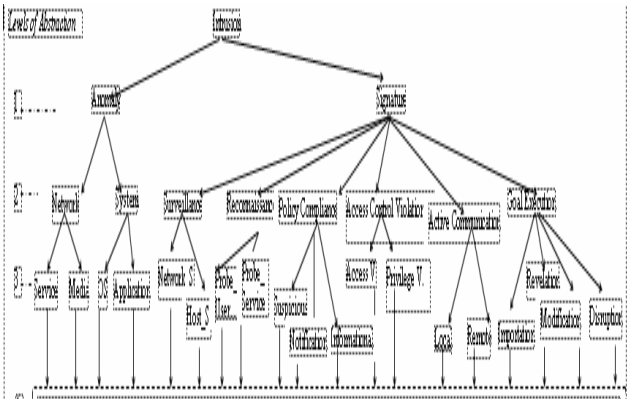

Fig. 2. Generalization Hierarchy for Attack Names

The following is a description of the subset of concept nodes in the hierarchy that corresponds to the leftmost *CEvents* in the abstract incident model of Figure 1.

- Surveillance: Designates alerts that are attributed to general activities which collect information about networks or systems. Surveillance activities are considered non-critical threats to the system but they may be used as preludes to conducting specific attacks on systems. For example, alerts for *IPSweep and Ping.*
- Reconnaissance: Designates alerts that can be attributed to activities that collect specific information about networks or systems. Reconnaissance activities are considered non-critical threats to the system but they may be used to conduct further malicious acts which may cause harm to systems. *Probe_of_Service* is a reconnaissance activity that designates alerts that are targeted to a particular system to specifically obtain information about specific services supported by the system (for example, alerts for Port Scan, Ping of Service).
- Access Control Violation: Designates alerts attributed to intrusive activities that compromise the system security perimeter. Examples are exploitation or manipulation of weak/insecure/inadequate system features or configuration/implementation errors to

gain access to a system. Access control violation activities pose an immediate critical threat to the system and may cause further impacts on the system. For example, alerts for *Dictionary* and *Fdformat*.

- Active Communication: Designates alerts that are attributed to general suspicious activities which open a communication channel between systems that may be used to transfer files to and from those systems. Active communication activities are considered to pose an immediate critical threat to the system and may be used for further attacks. For example, alerts for *command shell* or alerts for *remote shell*.
- Goal Execution: Designates alerts that indicate malicious attacks that unconditionally conflict with the security policy and have the potential to cause a system to behave in an unwarranted way. These activities are considered the most critical threats to a system. For example, alerts for *Mstream_Zombie* attack or Tripwire *integrity* alerts or alerts for *Syn Flood* attack.

The middle *EEvents* in the abstract incident model of Figure 1. correspond to various security incidents and are described below:

- Disclosure_of_Host (DHS): This event occurs when there is evidence that a resource's identity is exposed or disclosed to outside users. *Surveillance CEvent* triggers this event in system. The knowledge about the existence of the resource can be used by the intruder in further probing to gain additional information to continue with additional attacks.
- Disclosure of Service (DSV): This event occurs when there is evidence that existence of a particular service resided on a particular resource is revealed and when there is pre-existing risk of such disclosure present. *Probe_of_Service CEvent* triggers this event in the system. The knowledge of the existence of a particular service that has known vulnerabilities/weaknesses can be exploited by the intruder for further attacks.
- System_Environment_Corruption (SEC): This event occurs when there is evidence of activity that may result in unauthorized access to resources such that the resource's security perimeter is breached and when there is pre-existing risk of such activity present. *Access_Control_Violation CEvent* triggers this event in system. Once the security perimeter is breached, an intruder can take necessary measures to attack the resource itself or to use the compromised resource to launch further attacks against other resources in the network.
- System Seizure (SSZ): This event occurs when there is evidence of an unauthorized communication channel with the resource in question indicating total control over the resource and when there is pre-existing risk of such activity present. *Active_Communication CEvent* triggers this event in system. With transferring necessary files or tools, the intruder can proceed to attack the resource itself or use the compromised resource to launch attacks against other resources in the network.
- System Distress (SDT): This event occurs when there is evidence of definitive malicious attack and when there is pre-existing risk of such an activity present. *Goal_Execution CEvent* triggers this event in the system.

All these events are linked together by cause and effect relationships in the abstract incident model of Figure 1. For a DHS incident, there is no predecessor incident in the correlation chain and all other DSV, SEC, SSZ and SDT incidents are considered its successors. For a DSV incident, the predecessor incident in the correlation chain is DHS and SEC, SSZ and SDT incidents are considered its successors. For an SEC incident, DHS and DSV incidents are considered its predecessors and SSZ and SDT incidents are considered its successors in the correlation chain. For an SSZ incident, DHS, DSV and SEC incidents are considered its predecessors and an SDT incident is considered its successor. For an SDT incident, all other DHS, DSV, SEC, and SSZ incidents are considered its predecessors and it has no successor incident in the correlation chain.

The following properties hold for inference with the abstract incident model:

- Suppose, I_P denotes a predecessor incident and I_S denotes a successor incident in a correlation scenario found for a resource R_i. If the earliest occurrence time of alerts contributing to I_P is $t_{start\text{-}ip}$, latest occurrence time of alerts contributing to I_S is $t_{end\text{-}is}$, occurrence time of an alert contributing to I_P is t_{ip}, and occurrence time of an alert contributing to I_S is t_{is}, then the following must be true for the alerts to be correlated:
  - *with predecessor incident*
    - $t_{is} => t_{start\text{-}ip}$
  - *with successor incident*
    - $t_{ip} <= t_{end\text{-}is}$
- For I_P and I_S to be correlated, they must occur between the same pair of hosts with one of them being the resource in question. (An exception is a System_Distress incident, which may involve the resource in question with any other host. This is because once a multi-staged attack proceeds to the SDT level, alerts can designate definitive attacks directed to the host in question (e.g.,

DDOS_shaft_handler_to_agent[2] alert) or alerts can designate attacks originating from the host in question (e.g., *Mstream_Zombie_Respons*[3] alert*)*. Nevertheless, in both cases, the host in question is considered as the target of the attack.)

For abstract alert correlation, evidence in the sensor reports (i.e., sensor generated alerts) are initially generalized to abstract alert types (Figure 2.) and then mapped to the leftmost *CEvents* as shown in Figure 1. For example, if there is an alert that indicates a *sadmind* buffer overflow, then instead of generating a specific event like *SadmindAmslverifyOverflow*, abstract alerts are used to activate more generalized *CEvents* like *Access_Control_Violation*.

Note that the same *CEvent* will also be generated for similar types of alerts such as *StatdOverfllow* [4] or *SolarisLPDOverflow* [5] . Instead of generating specific events like *VulnerableToSadmind*, more generalized *EEvents* like *Disclosure_of_Service* are activated. Note that this same *EEvent* can also replace other specific events like *VulnerableToStatd,* or *VulnerableToSolarisLPD.*

Different actions of an attacker, targeted at a particular host, activate different incidents for that resource [16]. The extent to which such an incident occurs depends not only on the evidence of the corresponding action taken by the attacker as found in the sensor generated report, but also on the existing risk of such an incident taking place (Figure 1.), i.e., the determination of what has happened jointly depends on what was reported to have happened (i.e., current evidence of the incident) and what could have happened (i.e., the possibility of the incident). For example, the *EEvent,* System_Environment_Corruption (SEC) primarily depends on the sensor reporting of the *CEvent, Access_Control_Violation* (alert impact designated by an FCM edge of +1.00). This type of action is not always successful and therefore, sensor notification of this alert does not guarantee that such an incident actually took place. The abstract incident model deals with this uncertainty by taking additional information into account, i.e., the pre-existing risk of such an incident happening for the particular resource in question as shown by the middle *CEvents* in Figure 1. It shows this risk impact designated by the FCM edge of +0.50 for the incident System_Environment_Corruption. Note the difference between alert and risk impact. This is because security administrators tend to pay more attention to the report of the alert itself than to its existing risk. Sometimes when alerts such as - *rsh, Telnet XDisplay, Ftp_Put* (which generate Active_Communication *CEvent*) are issued by sensors, the existing risk (or possibility) of such an incident occurring impacts the incident more than the alerts do since such alerts are not always indicative of actual malicious activities. Hence impacts of such *CEvents* are less than the impacts of the associated risks. It should be noted that such risk computation can also incorporate other characteristic features of the resource itself such as the presence of known vulnerabilities in the host that can be exploited to cause security incidents. For example, if a resource is known to have the *sadmind* service running, thus making it vulnerable to a buffer overflow type of attack, this would increase the risk of the incident System_Environment_Corruption for that resource.

All alerts contributing to *CEvents* of the abstract incident model can be correlated as part of a general multi-staged attack scenario denoted by the FCM model. Figure 3. shows the steps for such abstract alert correlation:

```
for each host x in X (X: host list for the protected environment)
{
    get all alerts into A that involve any communication with x
    for each alert a in A
    {
      generate CEvent for the alert type
    }
    get list of each host y in Y that are in communication with x
    for all alerts that involve communication between x and y
    {
        generate EEvents (incidents and risks) and correlate alerts
              check for isolated alerts
              check for isolated non-critical incidents
        identify x as compromised
    }
}
for each host h in H (H: hosts reported with correlated alerts)
{
    for each incident
            compute Incident Strength
    compute total Incident Association Strength
}
```

Fig. 3. Abstract Alert Correlation Steps

It should be noted that a correlated scenario is unique between the pair of hosts involved in the communication. Multiple scenarios can be activated for one victim host because it is feasible for a host to become the target of a coordinated attack launched from multiple attack sources. In that case, the coordinated scenarios are reported differently depending on the source of the attacks and the nature of the attacks. When correlating multiple alerts for multi-staged attacks scenarios, isolated incidents (i.e., if

[2] This is an alert generated by Snort denoting that a DDoS Shaft handler is directing a DDoS Shaft agent (compromised host) to launch an attack (http://www.snort.org).

[3] This is an alert generated by RealSecure denoting that an mstream agent/zombie (compromised host) is responding to an mstream handler/master (http://xforce.iss.net/xforce/search.php)

[4] *StatdOverflow*: An attack that exploits vulnerability associated with Solaris system's *statd* program that provides network status monitoring and crash and recovery functions.

[5] *SolarisLPDOverflow*: An attack that exploits vulnerability associated with *Solaris BSD print protocol* daemon.

alert correlation results in only one type of incident) are disregarded intuitively. An exception is the System_Distress incident, which may be generated due to isolated attacks such as *smurf* or *syn_flood*, and therefore even if isolated, it is reported because of its critical nature.

With FCM modeling of system events, presence of all predecessor events in the abstract incident model is not necessary to infer all subsequent events. For example, if a sensor does not report an Access_Control_Violation *CEvent* (which is possible because "false negatives" are a common problem for sensors), a System_Environment_Corruption *EEvent* is still activated to some extent. This is because once Disclosure_of_Service *EEvent* gets activated in the system as a result of the Probe_of_Service *CEvent*, it activates Risk_of_System_Environment_Corruption immediately. The Risk_of_System_Environment_Corruption *EEvent* consequently activates the System_Environment_ Corruption *EEvent* to some extent (not in full because some of the evidence of the incident is missing). This eventually causes other subsequent *EEvents* to activate partially as further inference takes place. As the situation builds and more associated alerts are reported, resulting *EEvents* become stronger. Therefore, alert correlation is able to progress to a partial extent with missing alerts in the sensor reports. Thus, use of abstract incident modeling allows to replace multiple explicit attack models and helps with scalability and uncertainty issues in alert correlation.

Along with correlating alerts, we also report security incidents that occur for the resources[6] [16]. The extent to which a particular security incident occurs designates its *incident strength*. In accordance with FCM inference [7], the strength of a successor incident I_s activated for a host or resource R_i at t_{n+1} time for each predecessor incidents I_p with impact e_{pi}, can be represented by the following:

$$S(I_s)(t_{n+1}) = \left[\frac{\sum_{p=1}^{n} (I_p)(t_n) * e_{pi}(t_n)}{\sum_{p=1}^{n} e_{pi}(t_n)} \right]$$

It should be noted that when a particular host becomes the target of different coordinated attacks launched from different sources, for the computation of the incident strengths, the individual incidents that occur for the resource are taken into consideration to compute overall impact of the incidents on the resource, irrespective of their context. The incident model combines the strengths of the different incidents activated for a particular resource in order to measure the extent of its incident association.

[6] We use the term resource to denote protected hosts or systems in the network.

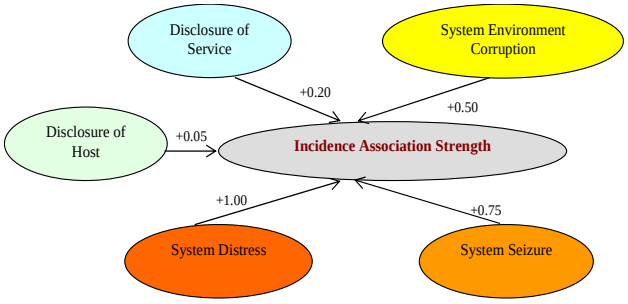

Fig. 4. Combining Evidence of Security Incidents

Figure 4. shows how the evidence of different incidents activated for a resource contribute to the overall incident association of the resource with different impacts. The degree of impact depends on the nature of the incident and the designated security policy [16]. At any time, the *Incident Association Strength* (IAS) of a particular resource collectively represents the effects of all the security incidents activated for the resource at that time. Therefore, the IAS of a resource R_i at time t_{n+1} for each contributing incidents I_k with impact e_{ki}, can be represented as the following:

$$IAS\ (R_i)(t_{n+1}) = \left[\frac{\sum_{k=1}^{n} (I_k)(t_n) * e_{ki}(t_n)}{\sum_{k=1}^{n} e_{ki}(t_n)} \right]$$

With abstract incident modeling, alerts are correlated to find incidents in a coordinated attack scenario. Along with incidents found for each resource, their Incident Association Strengths (IAS)s are also reported. It should be pointed out that IAS can be considered a confidence score given by the incident model to represent the degree of concern for a particular resource's involvement in correlated security incidents resulting from multi-staged attacks.

4. Experiments and Results

For experimentation, MIT Lincoln's Lab's DARPA (Defense Advanced Research Projects Agency) 2000 Intrusion Detection Evaluation (IDEVAL) Scenario Specific dataset [8] was used as the test data because it is a well known benchmark dataset that contains simulated multi-staged attack scenarios in a protected environment. The fact that the ground truth required for validation purposes cannot be known for real world traffic, has inspired us to use this simulated attack traffic for which ground truth is known.

In the Lincoln Lab DARPA (LLD) experiment, the attack traffic includes a series of attacks carried out over multiple

networks and audit sessions by an attacker who probes hosts in the network, successfully breaks into some of them to prepare for and finally launch Distributed Denial of Service (DDoS) attacks against an off-site government website. The attack experiments were conducted over three segments of a simulation network: a network inside an Air Force base, an internet outside an Air Force base and the demilitarized zone (DMZ) that connects the outside network to the inside network [8]. There are two attack scenarios in the LLD attack traffic:

- one scenario includes DDoS attacks carried out by a novice attacker (LLDOS 1.0) who compromises three hosts individually to launch the attack against an outside host;
- another scenario includes DDoS attacks carried out by a more sophisticated attacker (LLDOS 2.0.2) who compromises one host and then fans out from it.

In general, attacks in LLDOS 2.0.2 are stealthier that those in LLDOS 1.0. Overall, there are four *tcpdump* files containing the attack traffic: LLDOS 1.0 Inside Zone; LLDOS 1.0 DMZ; LLDOS 2.0.2 Inside Zone; and LLDOS 2.0.2 DMZ.

For evaluating our technique, the LLD attack traffic itself was not sufficient because we required sensor alert reports that would result from monitoring this attack traffic. Therefore, for the purpose of our research, we needed to generate these sensor alert reports as part of the research. In this regard, two intrusion detection systems (IDS)s or sensors, RealSecure (Version 7.0) [3] (a commercial signature based network sensor) and Snort (Version 2.3.3) [14] (an open source lightweight signature based network sensor) were used. Apart from the fact that these sensors are among the most widely used sensors today, RealSecure was selected because other researchers have also used this IDS for similar purposes [10] and Snort was selected because it is a freely available IDS.

To generate the sensor alert reports, Snort was configured to execute in full coverage (with all available attack signatures active) and installed in the security lab of the Center of Computer Security Research (CCSR), Mississippi State University (MSU), to monitor the simulated network traffic containing the LLD attacks. Since RealSecure is not designed to monitor offline *tcpdump* data, the *tcpdump* files had to be replayed in a live network using *tcpreplay*, a *tcpdump* file utility program offered by Open Source Technology Group [11]. We elected to execute RealSecure with the "Attack Detector" policy, instead of the "Attacks and Audits" policy (which is equivalent to executing Snort with all the default rules), for the following reasons:

- Executing RealSecure and Snort with an equivalent policy generated almost the same sets of alerts from both sensors since they both sniffed the same attack traffic. Consequently, we found the same results by analyzing the sensor reports individually. Apart from showing that our technique performs consistently, it is not interesting to compare and contrast the results. Differences in sensor scope coverage resulted in different sensor reports and we wanted to evaluate our approach in detecting these differences.
- In real situations, it is likely that one will need to make use of sensors with different coverage. From a security administrator's point of view, provided with two identical sensors, it makes more sense to use one of them with full coverage and one with focused coverage. This is because full coverage generates a very large amount of alerts in sensor report (for example, RealSecure generated more than 39K alerts for the LLDOS 1.0 inside zone attack traffic alone), which not only included clear attacks but also include all audits for any kind of notable activities. On the other hand, for a better understanding of the big picture, sometimes it is beneficial to analyze all activities to trace the malicious ones to their roots or to link them together.

Evaluating the incident model's performance in building an overall security view by analyzing integrated alerts reported by multiple sensors (in our case, RealSecure and Snort), involved multi-sensor data generation consisting of:

- simulation of attacks in a live network with *tcpreplay* and LLD attack traffic;
- installation of both sensors such that they would monitor the same traffic simultaneously and generate alerts independently; and finally
- integration of the individual sensor alert reports.

Both RealSecure and Snort were installed in the same host to monitor the LLD attack traffic. To integrate the sensor alerts, the following alert features were extracted from the independent sensor reports: Source, Target, Time, and Attack name. An additional feature identified the sensor and the individual alerts uniquely.

For experimentation, the generalization hierarchy shown in Figure 4. was used to generalize the attack names in the sensor alert reports into abstract alert types. Since the sensors (Real Secure and Snort) used in the experiments were both signature-based or misuse sensors, alert abstraction was limited up to level 2 of the generalization hierarchy. The low-level alerts reported by RealSecure were generalized with the help of attack signature descriptions provided by ISS, Inc.'s X-Force database, a very comprehensive threats and vulnerabilities database

(http://xforce.iss.net/), and generalization of the Snort alerts were conducted using attack signature descriptions provided by Sourcefire, Inc. (http://www.snort.org/). In addition, security experts were consulted for their comments/suggestions on the generalization. If interested, please contact the authors for complete categorization of the attack names reported by RealSecure and Snort.

The alert correlation experiment was conducted on the individual RealSecure and Snort alert reports (RealSecure-MSU and Snort-MSU sensor report) and the integrated multi sensor (MultiSensor-MSU report) separately. While the individual sensor reports were used to evaluate and compare the alert incident model's alert correlation performance for intra-sensor fusion (i.e., between alerts generated by a single sensor), the multi-sensor report was used to evaluate its performance for inter-sensor fusion (i.e., between alerts generated by different sensors). In this paper, we focus on the results with the multi-sensor report due to space constraints.

Evaluating a high-level reasoning process, like alert correlation with abstract incident modeling (abstract alert correlation), is not trivial as it involves many subjective and qualitative factors. To show the alert correlation capability of the incident model, the following correlation performance metrics are used in this dissertation, as suggested by Qin and Lee [13]:

- True Causality Rate (TCR): is measured by the ratio of the number of correctly correlated alerts[7] for a scenario to the total number of actual causal relationships[8] that are involved in the scenario.
- False Causality Rate (FCR): is measured by the ratio of the number of incorrectly correlated alerts for a scenario to the total number of correlated alerts reported for the scenario.

It should be pointed out that TCR essentially measures the detection rate for alert correlation and can be used as an indicator of how completely the incident model is able to correlate alerts. FCR measures the false positive rate for alert correlation and provides insight into how correctly the incident model is able to correlate alerts. Table 1. denotes the alert correlation results for the multi-sensor report in terms of the metrics described earlier. It should be noted that this only concerns the correlated alerts reported by the incident model. The abbreviations used are the following:

SA: Sensor reported Alerts, CR: Causal Relationships, CA: Correlated Alerts, CCA: Correctly Correlated Alerts, ICA: Incorrectly Correlated Alerts, MA: Missed Alerts, TCR: True Causality Rate, FCR: False Causality Rate.

Tab. 1 Correlation Performance for the MultiSensor-MSU Report

Dataset	SA	CR= CCA+ MA	CA= CCA+ ICA	CCA	ICA	MA	TCR= CCA/CR	FCR= ICA/CA
LLDOS 1.0 Inside Zone	1353	91	86	75	11	16	82.42%	12.79%
LLDOS 1.0 DMZ	3932	161	136	122	14	39	75.78%	10.29%
LLDOS 2.0.2 Inside Zone	920	26	22	22	0	4	84.61%	0%
LLDOS 2.0.2 DMZ	1508	9	10	9	1	0	100%	10%

With alert correlation on the multi-sensor data, the following results were expected:

- Alerts incorrectly correlated in intra-sensor fusion (analyzing the individual sensor reports) would also be incorrectly correlated in inter-sensor fusion (analyzing the multi-sensor report). This is because evidence present in individual sensor reports that leads to incorrect correlation, remain present when integrated. However, there should not be additional false positives. The only exception would be if incident situations are discovered when unrelated evidence found in individual sensor reports are linked together in the multi-sensor report to collectively discover a seemingly coordinated attack scenario.
- Alerts missed in intra-sensor fusion would also be missed with inter-sensor fusion. However, there should not be any additional false negatives unless there is related evidence found in individual sensor reports that cannot seemingly be linked together in the multi-sensor report to support a coordinated attack scenario.

As experiment was conducted on the multi-sensor report, the following was found:

- As expected, the incident model incorrectly correlated the same set of alerts for inter-sensor fusion that were incorrectly correlated for intra-sensor fusion (in case of RealSecure-MSU and Snort-MSU sensor reports). That is, the number of incorrectly correlated alerts (ICA) in the integrated multi-sensor reports (column 6 of Table 1.) are the sum of the number of incorrectly correlated alerts (ICA) in the RealSecure-MSU sensor report and the number of incorrectly correlated alerts (ICA) in the Snort-MSU sensor report.
- Unexpectedly, there were additional missed alerts (shown in the last column of Table 2.) beyond those missed for RealSecure-MSU and Snort-MSU sensor reports. That is, the number of missed alerts (MA) in the integrated multi-sensor reports (column 7 of Table

7 Correlated alerts: Alerts that are reported l to be part of coordinated attacks.

8 Causal relationships: Alert data that are part of coordinated attacks against target hosts.

1.) is greater than the sum of the number of missed alerts (MA) in the RealSecure-MSU sensor report and the number of missed alerts (MA) in the Snort-MSU sensor report. Table 2. shows the additional missed alerts.

Tab. 2. Comparison of Correlated Alerts found for RealSecure-MSU Report, Snort-MSU Report and MultiSensor-MSU Report

Dataset	Causal Alerts (CA) for Real Secure-MSU	CA for Snort-MSU	Expected CA for MultiSensor MSU= CA for RealSecure-MSU + CA for Snort- MSU	Actual CA for Multi Sensor MSU	Additional Missed Alerts= Expected CA - Actual CA
LLDOS 1.0 Inside Zone	16	71	87	86	1
LLDOS 1.0 DMZ	23	116	139	136	3
LLDOS 2.0.2 Inside Zone	8	15	23	22	1
LLDOS 2.0.2 DMZ	4	6	10	10	0

It should be noted that the discrepancy in the number of causal or correlated alerts for the RealSecure-MSU sensor report (column 2) and the Snort-MSU sensor report (column 3) is due to the fact that the sensors were executed with different security policies. In the case of LLDOS 1.0 Inside Zone dataset, the one additional missed alert was a *Sadmind_Buffer_Overflow* alert, reported by RealSecure and generated for the host *locke:* 172.016.112.010. This alert activated a System_Environment_Corruption incident for this host. However, since this incident occurred after its successor System_Seizure incident (activated due to alerts reported by Snort) and not before it, our model found the incidents unrelated. In the cases of LLDOS 1.0 DMZ and LLDOS 2.0.2 Inside Zone datasets, the following alerts were missed by the incident model for the same reason:

- For LLDOS 1.0 DMZ dataset, three *Sadmind_Buffer_Overflow* alerts reported by RealSecure for the hosts *plato:* 172.016.114.010, *smith:* 172.016.114.020 and *solomon:* 172.016.114.030;
- For LLDOS 2.0.2 Inside Zone dataset, one *Sadmind_Buffer_Overflow* alert reported by RealSecure for the host *pascal:* 172.016.112.050.

For comparison purposes, Table 3. shows a list of all hosts reported by the fusion model after analyzing the RealSecure-MSU Report, Snort-MSU Report, and MultiSensor Report. The listings in <BOLD> indicate the actual attacked hosts in the LLD experiments. As can be seen the incident model successfully report the attacked hosts with comparatively high incident association strengths (IAS), indicating their involvement in multi-staged attacks. The IAS reported for multi-sensor report is higher or equal to the individual sensor reports. This is because is some cases incidents were found to be linked together with inter-sensor fusion that were missing with intra-sensor fusion. Due to space constraints, here we only mention a very few from analysis of the multi sensor report.

Tab. 3. Comparison of Incident Situation Discovered after Analyzing RealSecure-MSU Report, Snort-MSU Report, and MultiSensor-MSU Report

Dataset	Host	IAS found for RealSecure-MSU Report	IAS found for Snort-MSU Report	IAS found for Multi Sensor-MSU Report
LLDOS 1.0 Inside Zone	**172.016.112.010**	70.8	100.0	100.0
	172.016.112.050	70.8	100.0	100.0
	172.016.115.020	70.8	100.0	100.0
	172.016.112.194		15.44	15.44
	172.016.114.050		34.37	34.37
LLDOS 1.0 DMZ	**172.016.112.010**	14.0	60.0	60.0
	172.016.112.050	34.0	60.0	60.0
	172.016.115.020	34.0	60.0	60.0
	172.016.114.010	34.0	60.0	60.0
	172.016.114.020	34.0	60.0	60.0
	172.016.114.030	34.0	60.0	60.0
	172.016.112.194		15.44	15.44
	172.016.114.001	32.03		32.03
	172.016.114.050	32.03	34.37	72.8
LLDOS 2.0.2 Inside Zone	**172.016.112.050**	48.7	72.8	72.8
	172.016.115.020	54.19	72.8	79.23
LLDOS 2.0.2 DMZ	172.016.114.001	32.03		32.03
	172.016.114.050	32.03		32.03
	172.016.115.020	18.67	34.37	40.51

In the case of LLDOS 1.0 DMZ dataset, for one of the host *marx*: 172.016.114.050, the incident model correlated SEC and SSZ incidents from evidence reported only by Snort with a SDT incident from evidence reported only by RealSecure. Figure 5. shows the incident situation for this host. Although these alerts came from different sensors (i.e., RealSecure and Snort), the incident model linked them together because the alerts corresponded to sequential incidents in a multi-staged attack such as shown in Figure 1. This correlation is justifiable because we are interested in the ultimate impact of security incidents on a target and it is feasible for a target to be attacked from multiple sources. Therefore, such correlation is needed for comprehensive security analysis. In this case (Figure 6.), as a result of this correlation, the IAS reported was higher (72.8%) as compared to what were found analyzing the RealSecure report (32.03%) and the Snort report (34.37%).

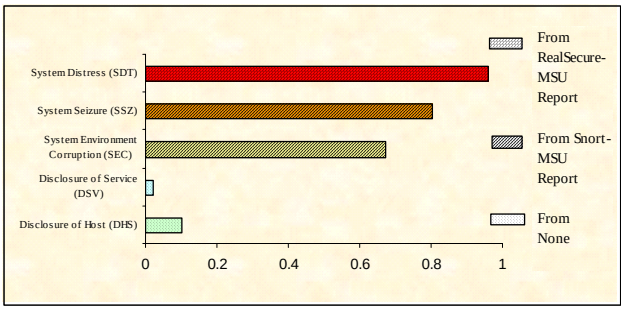

Fig. 5. Incident Situation for Host *marx:* 172.015.114.050 analyzing the MultiSensor-MSU Report

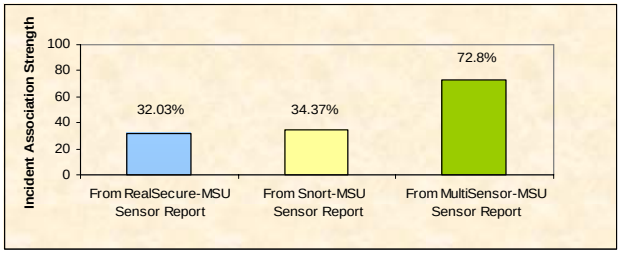

Fig. 6. Comparison of IAS reported for Host *marx:* 172.015.114.050 for RealSecure-MSU, Snort-MSU and MultiSensor-MSU reports

In the case of LLDOS 1.0 DMZ dataset, for the hosts under attack, the incident model was able to correlate DHS, DSV and SEC incidents inferred from evidence reported by RealSecure and Snort with the SSZ incident inferred from evidence reported only by Snort. Since the Snort-MSU report also had evidence of all related incidents, the IAS reported for the MultiSensor-MSU report is the same as that for the Snort-MSU sensor report.

In the case of LLDOS 1.0 Inside Zone and DMZ datasets, the incident situation for the host *mill*: 172.016.115.020, is found to be the same as that in the Snort-MSU sensor report. This is because the multi-sensor report for LLDOS 1.0 Inside Zone and DMZ datasets did not provide any additional evidence other than what already existed in the Snort-MSU sensor report. However interestingly, the IAS is reported higher in the case of LLDOS 2.0.2 Inside Zone and DMZ datasets of the multi-sensor report than for the respective datasets of the individual sensor reports. The following explains why.

Tab. 4. Comparison of Incident Situation for Host *mill* analyzing LLDOS 2.0.2 Inside Zone Dataset of RealSecure-MSU Report, Snort-MSU Report, and MultiSensor-MSU Report

Dataset	Analysis based on Sensor Report	DHS	DSV	SEC	SSZ	SDT	IAS
LLDOS 2.0.2 Inside Zone	RealSecure	1.0	0.2	0.733	0.44	0.888	54.19%
	Snort	0.1	0.02	0.673	0.8	0.961	72.80%
	Multi-Sensor	1.0	0.2	0.733	0.84	0.968	79.23%

Table 4. shows the incident situation for host *mill* in the case of LLDOS 2.0.2 Inside Zone dataset of RealSecure-MSU, Snort-MSU and MultiSensor-MSU reports. The yellow cells denote incidents activated based only on pre-existing risks and without any evidence from sensor reports. For example, none of the sensors reported any evidence of a DSV incident for this host. The blue cells in the last row indicate that in these cases, the abstract incident model complemented failure of one sensor in reporting an alert for a certain type of incident with another sensor reporting an alert for similar type of incident. The green cells in last row indicate both sensors reporting evidence of the same incident. For the multi-sensor report (last row in Table 4.), the reported strengths of incidents DHS, DSV and SEC, are the same as the maximum of the corresponding incidents' strengths reported for the individual sensor reports. This is because of the same evidence support and existing risk conditions. However, for the multi-sensor report, the reported strengths of incidents SSZ and SDT are higher than the corresponding incidents' strengths reported for the individual sensor reports. In these cases, although the supporting evidence was the same, the existing risks were higher in the case of the multi-sensor report (since the predecessor cases of the incidents activated to a higher degree). Therefore, the successor incidents (SSZ and SDT) were also activated to a higher degree and as a result, the overall IAS was reported higher (79.23%) than those found from analyzing the individual sensor reports.

In the experiments conducted for abstract alert correlation, we found that the incident model was able to correlate alerts that were generated as part of a coordinated attack scenario. While correlating alerts or finding causal relationship between alerts, it also reported on security incidents that had occurred for the hosts involved in the attacks. The extent of incident activation depended on evidence supporting the incident and the risk or the possibility of the incident occurring. That is, a high incident value indicated the presence of both evidence and risk for the incident and a low incident value indicated the absence of either the evidence or the risk. For each host reported under attack, an overall degree of concern for incident association was also reported. For example, a high incident association strength reported for a host indicated that one or more highly critical security incidents had occurred for the host and a low incident association strength reported for a host indicated that one or more less critical security incidents had occurred for the host. Thus incident association strengths and incident strengths provided the security administrator with an insight into the extent of concern for hosts involved in multi-staged attacks carried out by attacker.

5. Conclusion and Future Work

The main advantage of our alert correlation technique with abstract incident modeling has been shown to link together alerts that are involved in multi-staged coordinated attacks by considering both evidence of attacks present in the sensor reports and the possible occurrence of such attacks.

The abstract incident model allowed inference to progress even though evidence of attacks was missing in the sensor reports. Our correlation technique has been shown to properly derive quantitative assessments of the protected resources' involvement in multi-staged attacks. The level of such involvement provided insight into the criticality of coordinated attacks targeted towards a resource. In addition, the correlation technique has been shown to further reduce alert volume by reporting only correlated alerts.

In this research, we have used cognitive models with FCMs whose structures have been defined by human experts. However, the models are intuitive and generic and require little or no specialized knowledge. In the future, we will explore the use of adaptive FCMs, where the FCMs can self-learn and self-train like neural networks with minimal involvement of the human expert.

A critical assumption in this research is that meaningful generalization hierarchies have been defined for the alert features and that the sensor reported attacks are appropriately categorized into the developed attack generalization hierarchy or taxonomy. Defining such generalization hierarchies is a knowledge engineering task that has no single best way to be done. The generalization hierarchies used in this research are simply shown as examples to demonstrate the usefulness of our model.

In this research, we have focused on what has happened to a protected resource from evidence provided by sensor reports. In the future, we want to extend this work to predict an attacker's future plans such that we are able to report what might or is about to happen to a protected resource. This has the potential to warn the security administrator in advance and aid in preventing such attacks.

Another issue that is worth future investigation is the collaboration between multiple information sources to provide a more holistic view of security situations. Data from vulnerability scanners, honey pots, and performance monitoring systems can be utilized in this respect. Also, we would like to investigate incorporation of dynamic generalization hierarchy for alert feature abstraction.

It should be noted that the DARPA data is not intended to be conclusive examination of the effectiveness of our approach, but rather to provide a sense of how well and how accurate our approach works. Since our model has not been tested on a live system, a potential future research effort will be to experiment with real-time traffic in both distributed and cluster environment and with larger datasets.

References

[1] D. Brubaker, "Fuzzy Cognitive Maps," *EDN Access*, Apr. 1996.

[2] M. Caudill, "Using Neural Nets: Fuzzy Cognitive Maps", *AI Expert*, vol. 6, 1990, pp. 49-53.

[3] Internet Security Systems, "RealSecure Network 10/100," http://www.iss.net/products_services/enterprise_protection/rsnetwork/sensor.php (current Aug. 2004).

[4] K. Julisch, "Mining Alarm Clusters to Improve Alarm Handling Efficiency," *Proceedings:* 17th Annual Computer Security Applications Conference (ACSAC'01), New Orleans, LA, Dec. 2001.

[5] B. Kosko, "Fuzzy Cognitive Maps," *International Journal of Man-Machine Studies*, vol. 24, 1986, pp. 65-75.

[6] B. Kosko, Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence, Prentice Hall, Englewood Cliffs, NJ, 1992.

[7] B. Kosko, *Fuzzy Engineering*, Prentice Hall, Upper Saddle River , NJ, 1997.

[8] M.I.T Lincoln Laboratory, "2000 DARPA Intrusion Detection Scenario Specific Data Sets," http://www.ll.mit.edu/IST/ideval/data/2000/2000_data_index.html (current Jul. 2007)

[9] P. Ning, Y. Cui, and D. Reeves, "Constructing Attack Scenarios through Correlation of Intrusion Alerts," *Proceedings: ACM Conference on Computer & Communications Security*, Washington D.C., WA, Nov. 2002.

[10] P. Ning, D. Reeves, and Y. Cui, *Correlating Alerts using Prerequisites of Intrusions*, technical report TR-2001-13, Department of Computer Science, North Carolina State University , 2001.

[11] Open Source Technology Group, Inc., "Tcpreplay: Pcap editing and replay tools for *NIX," http://tcpreplay.sourceforge.net/ (current Aug. 2005).

[12] P. A. Porras, M. W. Fong, and A. Valdes, "A Mission-Impact-Based Approach to INFOSEC Alarm Correlation," *Proceedings: Recent Advances in Intrusion Detection*, Zurich, Switzerland. Oct. 2002.

[13] X. Qin and W. Lee, "Statistical Causality Analysis of INFOSEC Alert Data," *Proceedings: Recent Advances in Intrusion Detection*, Pittsburgh, PA, Sep. 2003.

[14] M. Roesch, "Snort-Lightweight Intrusion Detection for Networks," http://www.snort.org/docs/lisapaper.txt (current Jul. 2007).

[15] A Siraj, S. M. Bridges, and R. B. Vaughn, "Fuzzy Cognitive Maps for Decision Support in an Intelligent Intrusion Detection System," *Proceedings: International Fuzzy Systems Association/ North American Fuzzy Information Processing Society*

(IFSA/NAFIPS) Conference on Soft Computing, Vancouver, Canada, Jul. 2001.

[16] A. Siraj, and R. B. Vaughn, "A Cognitive Model for Alert Correlation in a Distributed Environment", Proceedings: *IEEE International Conference on Intelligence and Security Informatics (ISI 2005)*, Lecture Notes in Computer Science, Springer-Verlag, Volume 3495/2005.

[17] J. Q. Xin, J. E. Dickerson, and J. A. Dickerson, "Fuzzy Feature Extraction and Visualization for Intrusion Detection," *Proceedings: FUZZ-IEEE*, St. Louis, MO, 2003.

[18] D. Yu and D. Frincke, "A Novel Framework for Alert Correlation and Understanding," *Proceedings: International Conference on Applied Cryptography and Network Security* (ACNS), Yellow Mountain, China, 2004.

Ambareen Siraj obtained her PhD in Computer Science from Mississippi State University in the United States. She is currently an Assistant Professor of Computer Science at Tennessee Tech University. Her research interests include information assurance and security, artificial intelligence and software engineering

Rayford B. Vaughn, Jr. received his Ph.D. from Kansas State University in 1988 and is currently the Billy J. Ball Professor of Computer Science and Engineering at Mississippi State University. He teaches and conducts research in the Software Engineering and Information Security. Prior to joining the University, he completed a twenty-six year career in the Army where he commanded the Army's largest software development organization and created the Pentagon agency that today centrally manages all Pentagon IT support. Dr. Vaughn has over 100 publications to his credit and is an active contributor to software engineering and information security conferences and journals. He is actively engaged in high performance computing intrusion detection system research at Mississippi State University and established the MSU Center of Computer Security Research in 2001. In 2004, Dr. Vaughn was named a Mississippi State University Eminent Scholar and in 2005 he was given the "Most Outstanding Academic Award" by the National Colloquium on Information Systems Security Education. Today, Dr. Vaughn is the elected representative of all the principal investigators on the NSF Scholarship for Service program and member of the Interagency Coordinating Committee overseeing the SFS program. He maintains an active relationship with NSA as a part of the DOD Information Assurance Scholarship Program that MSU has been funded by since 2001.