4-18-2006

TANDI: Threat Assessment of Network Data and Information

Jared Holsopple
Rochester Institute of Technology

Shanchieh Jay Yang
Rochester Institute of Technology

Moises Sudit
University at Buffalo

TANDI: Threat Assessment of Network Data and Information

Jared Holsopple[a], Shanchieh Jay Yang[a], and Moises Sudit[b]

[a]Rochester Institute of Technology, 83 Lomb Memorial Dr., Rochester, NY, USA
[b]University at Buffalo, 409 Bell Hall, Buffalo, NY, USA

ABSTRACT

Current practice for combating cyber attacks typically use Intrusion Detection Sensors (IDSs) to passively detect and block multi-stage attacks. This work leverages Level-2 fusion that correlates IDS alerts belonging to the same attacker, and proposes a threat assessment algorithm to predict potential future attacker actions. The algorithm, TANDI, reduces the problem complexity by separating the models of the attacker's capability and opportunity, and fuse the two to determine the attacker's intent. Unlike traditional Bayesian-based approaches, which require assigning a large number of edge probabilities, the proposed Level-3 fusion procedure uses only 4 parameters. TANDI has been implemented and tested with randomly created attack sequences. The results demonstrate that TANDI predicts future attack actions accurately as long as the attack is not part of a coordinated attack and contains no insider threats. In the presence of abnormal attack events, TANDI will alarm the network analyst for further analysis. The attempt to evaluate a threat assessment algorithm via simulation is the first in the literature, and shall open up a new avenue in the area of high level fusion.

Keywords: Information fusion, Threat assessment, Impact assessment, Cyber attacks

1. INTRODUCTION

Cyber attacks typically occur over multiple machines in order to compromise important data or to impair network operations. Typical defense mechanisms for these attacks use intrusion detection sensors (IDSs) to detect suspicious activities on the network and apply rule-based policies to block these attacks from further damaging the network. Much research has been devoted to design and update the IDSs for automatic generation of alerts for the suspicious activities. More recent research in this field aims at correlating and grouping IDS alerts to identify the types of attacks.[1–3] Under current practice, network administrators are still mostly required to manually sort through the alerts to determine what has been compromised and thus the overall threat to the entire network. This work takes recent research one step further to investigate automatic estimation of the next targets in a multi-stage attack. This shall allow the network analysts to determine proactive actions for future attacker actions, instead of passively reacting to attacks that are already detected.

IDS alerts vary by the type of IDS used. Examples of IDSs are Snort,[4] Dragon,[5] and DAIWatch.[6] Alerts generated by these IDSs typically include reconnaissance actions, such as 'ping' or a port scan, and intrusion attempts, such as logon failures or service exploits. Since a cyber attack may occur in multiple stages over multiple machines, a good computer security tool should be able to correlate alerts generated due to the same attack and assess the threat associated with the attack. Unfortunately, typical tools in use today provide merely an interface to group and categorize alerts based on attack types or machine/subnet addresses. While experienced network analysts may be able to identify the attacker's target in a reasonable amount of time in some cases, it is desirable to have an automatic process running restlessly to monitor and predict the potential attacks before they happen. This requires correlating IDS alerts based on the attack's progression across the network, perhaps over different machines or subnets such as reconnaissance on multiple subnets followed by service exploit on one of the FTP servers in one of the subnets. Very little has been reported on how IDS alerts may be correlated, not to mention assessing the threats associated with correlated alerts. An example tool that correlates IDS alerts is called ECCARS[2] and is under revision at the time of writing this paper. The proposed threat assessment algorithm will leverage the work of ECCARS.

This work is funded by AFRL/IFEA. For further inquiry, send correspondence to S. Jay Yang (E-mail: jay.yang@rit.edu, Telephone: 1 585 475 6434).

The key challenge to assess potential threats due to cyber attacks lies in determining and modeling of the potential courses of action taken by the attackers. The course of action taken by an attacker depends on the attacker's capability, the opportunity or the vulnerability of the network, and the intent of the attacker.[7] Combinations of all possibilities can easily produce a large and perhaps impractical number of attack models to generate and maintain. In particular, it is undeniable that new attacks are being invented every day in the cyber domain due to the improved knowledge of the attackers and the ever increasing variety and complexity of software. In the absence of a perfect model, this work proposes to decompose the modeling of the capability of and the opportunity seen by attackers, allowing the models to be developed and revised accurately and in a timely fashion. A high level fusion scheme is developed to fuse the information provided by the two models for determining the intent of the attacker in real time. In addition to modeling complexity, the other challenge of this work is the lack of formal evaluation method for threat assessment schemes. To our knowledge, there exists no prior work on how to assess the validity and the performance of a threat assessment algorithm for any application domain. This work will provide a framework with several performance metrics serving to analyze threat assessment algorithms.

The rest of the paper is organized as following. Section 2 defines the threat assessment problem and the related work, followed by a discussion on the proposed framework and implementation of TANDI in Section 3. In Section 4, we will present our simulation results highlighting the use and the performance of TANDI.

2. PROBLEM STATEMENT AND RELATED WORK

Assessing the threats caused by cyber attacks based on IDS alerts is a multisensor data fusion problem. Multisensor data fusion was categorized by the Joint Director's Laboratory (JDL) into five levels[8]: Level 0 - signal refinement, Level 1 - object refinement, Level 2 - situation assessment, Level 3 - impact assessment, and Level 4 - process refinement. More recently, the fusion process refinement was suggested to be excluded in the definition of data fusion, because it is decision and reaction but not data fusion.[9] Threat assessment belongs to Level 3 impact assessment, which estimates the potential damage and threats that may be caused by current attack situations, which would be determined by Level 2 fusion. In other words, Level 2 fusion correlates IDS alerts and determines the network data and information that are currently compromised or have been attempted to be compromised; based on such, the threat assessment algorithm will identify those that are likely to be the next network entities to be under attack.

To be more specific on the type of information to fuse for threat assessment, one may step back and consider the six basic questions: How, Where, When, What, Why, and Who. Among them, the IDS alerts provide information to indicate (or at least imply) the **How**: the methods the attackers used to penetrate the network, the **Where**: the machines or subnets compromised (or attempted to be compromised) by the attackers, and the **When**: the time the attacks took place. The **What** is defined, in this work, as the network data and information, such as the existence of a machine, the root privilege, or Oracle database, the attackers may target on in each stage of the attack. The determination of the **Why** and the **Who** may require forensic analysis by experts and is out of the scope of this paper. Note that the 'When' can be indicative to the attacker's behavior and used to project the time of the next attack. However, we have not found consensus among subject matter experts (SMEs) on how to use such information. Hence, this work will only consider the order of which the attacks took place but not the exact time of the attacks. In summary, this work focuses on the threat assessment problem that predicts the potentially threatened 'What' based on the fused information of the 'How' and 'Where' of the already compromised entities in a network. Note that the key challenges of the threat assessment problem in the cyber domain is its modeling complexity and the ever changing methods of attack. Section 3 will illustrate how the proposed framework simplify the process by separating the modeling of the 'What,' 'How,' and 'Where.'

An important aspect of the threat assessment problem is to determine which network entity is more vulnerable. To some degree, threat assessment is similar to vulnerability analysis,[10,11] since both provide indications on how attackers may compromise a network. They are, however, different in that threat assessment needs to determine which network entity is more threatened than the others in real time based on the ongoing IDS alerts, instead of an offline analysis of which part of the network is more vulnerable assuming all types of attacks.

Among others, Bayesian networks and Hidden Markov Model have been recommended to be used for threat assessment.[12] They are reasonable choices because the most likely course of action may be deduced with

probabilistic inferencing from one action to the next. In the Bayesian network case, a course of action may be defined to represent the types of attacks the hacker could perform or the information the hacker could compromise during the attack. Though no literature is found using Bayesian networks for cyber security threat assessment, there are a few papers analyzing network vulnerability using such an approach. Phillips and Swiler[10] suggest that the course of action can be generated based on the topology of the network, the services running on each machine, configuration, user groups, attacker profile and other network characteristics. Each edge is assigned a probability that represents the probability of success. These probabilities then can be used as edge weights to compute the most likely path of the attack. Liu and Man[13] also develop a Bayesian vulnerability assessment technique using attack graphs structurally similar to that of Phillips and Swiler, but their graphs only contain root and user privileges. Note that different privileges allow access to different files, some privileges (such as network administrator) may be more important to the integrity of the network. Our threat assessment algorithm takes this into account and distinguish between the different privileges. Both work on vulnerability analysis[10, 11] motivate the design of our threat assessment algorithm.

While the above Bayesian-based approaches are comprehensive ideas, it may not be realistic for large and complex enterprise networks. Even if the course of action can be generated for such a network, assigning the probabilities is not a trivial task. The probabilities could be assigned in two ways. First, one or more SMEs could assign the probabilities manually.[12] However, with a large number of nodes, this could be a very time consuming and potentially inaccurate and high-maintenance task. The other way to assign the probabilities is by training based on historical data sets.[3, 12, 14] Unfortunately, the non-stationary nature of cyber attacks, i.e., old attacks will soon be obsolete and new attacks are being invented every day, presents a fundamental flaw to this approach. The historical data set may never be representative for the future courses of action. These observations suggest that, while the Bayesian network, at first glance, seems to be a good technique to assess threats in the cyber domain, the assignment of probabilities makes it not a viable choice.

While Bayesian network approach has been used for vulnerability analysis in the cyber domain, we have not found literature using Hidden Markov Models (HMM's) for cyber security. HMM's, however, have been used to detect, track, and predict terrorist activities.[12, 15, 16] In their models, each state corresponds to a specific sequence of events (instead of each node being an event in a Bayesian network). Given the actual detected sequence of events, a graph matching algorithm is used to identify the most likely current state of the HMM, and, consequently, to predict future activity via the transition probabilities. Similar to the Bayesian network case, the assignment of the transition probabilities may not be feasible in the cyber domain. Moreover, having each state representing a sequence of events, the HMM's may result in even more complex attack graphs than those using the Bayesian network approach.

Other more deterministic approaches have been proposed. For example, Vidalis and Jones[11] use vulnerability trees to model attacks where the root of the tree is the goal of the attack and the child nodes together define a course of action. The possible exploitation of each vulnerability is modeled by the educational complexity of the attacker, which measures how advanced the attacker must be to exploit the vulnerability. This procedure may not be feasible for systems that have a large number of attacker goals. A different feature tree will need to be developed for each goal, so the generation of these feature trees for a large number of goals could be potentially tedious and error-prone.

Changwen and You[17] use a decision making matrix to determine the threat of enemy vehicles and missiles in a military application. Their technique incorporates multiple attributes such as altitude, velocity, and attack angle. Fuzzy membership functions are defined to calculate the overall threat. These membership functions must be manually created, and may not be suitable in the cyber domain where unknown attacks are possible and being invented every day.

Aside from the above theoretical approaches, there exists a commercially available 'pseudo' threat assessment tool provided by Cisco Systems Inc. in their netForensics application.[18] This tool examines alerts on a per machine basis and classifies the alerts on a scale of 1 to 5, with 5 being the most severe. Based on the number of alerts in each scale, the tool calculates the overall threat score of a machine based on the following formula.

$$ThreatScore = \sum_{n=1}^{5} (AlertCount_n) * (2^{n-1} - 1). \quad (1)$$

A risk factor then can be calculated based on this $ThreatScore$ and user-defined '$Popularity$,' '$SystemValue$,' and '$Exposure$.' This approach is essentially used to rank and present to the network analyst the number and the levels of attacks that have happened to the computers in the network. It does not provide any prediction of future attacks nor correlating alerts across machines or subnets. Nevertheless, this seems to be the only commercially available tool that claims to assess threats induced by cyber attacks.

3. TANDI: THREAT ASSESSMENT OF NETWORK DATA AND INFORMATION

The complexity of modeling possible attacks on large scale networks with a variety of operating systems and software packages makes threat assessment a challenging task. Bayesian-based approaches or probability inferencing may not be viable solutions due to the non-stationary nature of cyber attacks. A framework for Threat Assessment of Network Data and Information (TANDI) is, therefore, developed to reduce the modeling complexity and to avoid assigning a large number of edge probabilities (or weights) with 'logical inferencing.'

3.1. The Framework

Recall Section 2 where the threat assessment problem is defined as fusing the 'How' and the 'Where' to predict the 'What.' The key feature of TANDI is its separation of the modeling of the 'How,' the 'Where,' and the 'What' aspects of cyber attacks. For the ease of illustration, the three models shall be referred to as the **attack sequence** ('How'), the **logical topology** ('Where'), and the **information graph** ('What'). Each of the three models may be represented by a directed graph where a traversal over nodes connected by directed edges reflects a potential course of attacks, as shown in Figure 1. For example, an internal machine A with two and only two incoming edges ($B \rightarrow A$) and ($C \rightarrow A$) can be compromised if and only if at least one of the two external machines, B or C, has been compromised. This is what we meant by 'logical inferencing,' where there is no need to differentiate the likelihood each parent node may result in the child node being asserted. Note that the logical topology depends on the network infrastructure and its organization; hence, it can be developed as being independent of the attack sequence the attacker may take. Likewise, the development of the attack sequence does not need to account for the specific network topology. This separation eases not only the development but also the error tracking and maintenance of individual models. The information graph, meanwhile, does depend on the network organization and the types of attacks one can use. A procedure has been developed to automatically generate the information graph based on the logical topology and the attack sequence. The procedure will be illustrated later in Section 3.2.

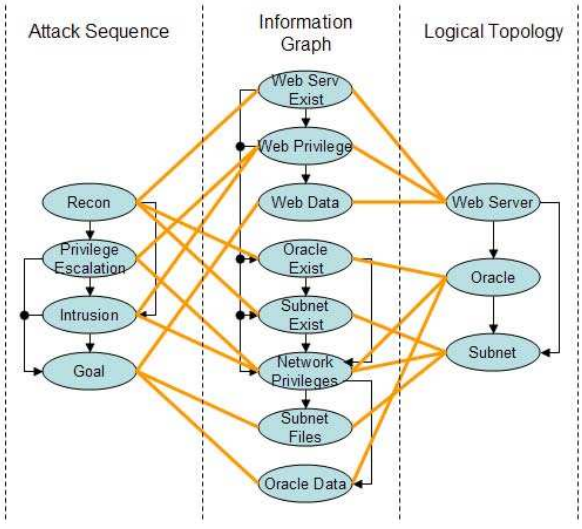

Figure 1. An example showing the attack sequence, the information graph, the logical topology, and how they are interconnected.

Each node in the logical topology and the attack sequence graphs are to be evaluated in real time as cyber attacks occur. The 'asserted' nodes reflect that the corresponding machines have been compromised or attack methods have been used. Consider this real-time situation assessment that provides indication of the attacker's capability (the types of attack methods he or she knows), and that of the attacker's opportunity (the machines or subnets he or she has compromised). By fusing the two, one may predict the intent of the attacker's next target, i.e., the nodes with high threat scores in the information graph. Following this intuition, TANDI performs situation and threat assessments on a per attack basis; that is, IDS alerts belonging to different attack will be evaluated separately. Grouping alerts to different attacks is the task of an underlying alert-correlation engine, such as ECCARS.[2] This restriction may be relaxed and, in fact, will allow identifying coordinated attacks and is under investigation at writing of this paper.

The fusion of the logical topology and the attack sequence information for predicting the threatened network entity may be represented with the use of the undirected edges connecting the nodes in the three model graphs, as shown in Figure 1. A network entity is connected to a machine or a subnet if the machine or the subnet contains or can access the entity. Similarly, a network entity is connected to an attack node if such type of attack can compromise the entity. The nodes in the logical topology and attack sequence can also be aggregate nodes that represent a group of computers (e.g. a subnet) or a set of alerts. These connections dictate how TANDI performs situation and threat assessments. For situation assessment, the information nodes that are associated with at least one of the asserted attack nodes and at least one of the asserted topology nodes are considered asserted or, equivalently, compromised. For example, considering Figure 1, if an IDS alert aggregated by the 'Intrusion' node occurs on the 'Web Server', TANDI will determine that the 'Web Server Privileges' is compromised by the attack because of the undirected edges between the models. Once determining the asserted information nodes, all the successor nodes of them are potential network entities that will be under attack next. The proposed threat assessment algorithm will fuse the information provided from the attack sequence model and the logical topology model, to determine a threat score for each of the successor nodes. These threat scores then can be presented to the network analyst for possible proactive and preventive actions. Note that there are many ways to determine the threat scores for the information node, and all of them should encompass the structure of the three directed graphs as well as the connections between the models. Section 3.3 will present our proposed threat assessment fusion algorithm, which we believe to be adequate and serve the purpose of better understanding the threat assessment problem in the cyber domain.

3.2. The Information Graph and Automatic Generation

The information graph represents the relationships between network entities such as privileges, databases, proprietary files, etc. By analyzing the structure of modern machines and the typical attack sequences, an inherent three level hierarchy of network entities was observed: an attacker needs to (1) know the existence of the machine or subnet, (2) obtain appropriate access privilege, and (3) access the target information or files. Based on this observation and inspiration from the work by Phillips and Swiler[10] and that by Liu and Man,[13] a template representing an information subgraph for a typical machine* is developed and shown in Figure 2. This template is cloned with changes for each of the nodes in the logical topology graph. Together, the clones form the entire information graph.

Each of the entities in each clone is associated with the corresponding machine or subnet, as well as the attacks that can compromise the entity. Recall the undirected edges in Figure 1, and note that these edges represent the associativity of the cloned information nodes to the nodes in the logical topology and those in the attack sequence graph. During the process of the cloning, the IDS alerts belonging to the associated attacks but cannot occur for the services running on the corresponding machines will be removed. This is done to increase the accuracy of the threat and impact assessment.

Note the 'Incoming Computer' node and the 'Outgoing Computer' node in the template. These two nodes are used to build the connection between network entities associated with different machines. For any directed edge $A \rightarrow B$ in the logical topology, the information nodes representing the existence of and the privileges at

*The information subgraph template can be extended and used to model network user accounts and files across computers. The discussion of this extension is omitted in this paper due to space limitation.

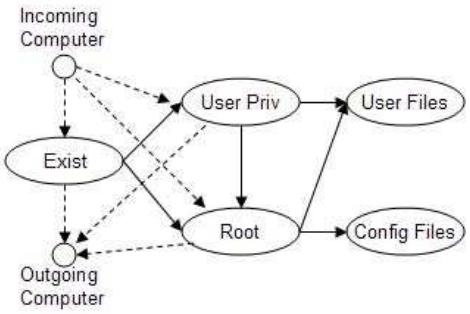

Figure 2. Information Graph Template defined for each computer or group of computers.

machine A will be connected to those of machine B (also with directed edges). These connections complete the automatic generation of the information graph.

3.3. Fusion for Threat Assessment: The Feature Tree

Recall that the proposed threat assessment algorithm examines the successor nodes of those that have been asserted due to previous actions of a given attack. Denote the set of information nodes that have been compromised no later than the jth event occurring in attack A as $I^*(e_j^A)$, and their successor nodes as $I(I^*(e_j^A))$. Note that, in this definition, the set $I(I^*(e_j^A))$ will exclude those that are also in the set $I^*(e_j^A)$. Immediately after the jth event of attack A, a **threat score**, $0 \leq t_i(e_j^A) < 1$ will be determined for every node in the set $I(I^*(e_j^A))$. All nodes in $I^*(e_j^A)$ are assigned a threat score of one, indicating that they have been compromised. Nodes that are not in either of these two sets (i.e. nodes that are two or more hops from all nodes in $I^*(e_j^A)$) are assigned a threat score of zero. Note that, by restricting assessing threat scores for the successors of already compromised nodes, one can detect abnormality, such as insider threats and coordinated attack.

To determine the threat scores for the nodes in $I(I^*(e_j^A))$ upon the occurrence of the jth event of attack A, a feature tree is evaluated for each of these nodes. Figure 3 shows the feature tree used for the 'Web Privilege' node drawn in Figure 1. Note that the alerts shown in this feature tree are those associated with 'Privilege Escalation' and 'Intrusion' alerts that can happen on a web server. The structure of this feature tree determines the fusion rule, and is followed for all the nodes in the information graph. The feature trees for different information node differ in the alerts and the machines that are connected as leaf nodes.

In the current implementation, a 'Sum-of-SumSets' operation is used. The Sum-of-SumSet operation uses only four pre-determined weights, which is a substantial reduction from the probability inferencing approach where one needs to determine the weights for all edges. The four weights: λ_{A^*}, λ_{M^*}, $\lambda_{A(A^*)}$, and $\lambda_{M(M^*)}$ correspond to the partial threat scores due to the already asserted attack nodes (A^*), the already asserted machines (M^*), the successor nodes of A^*, ($A(A^*)$), and the successors of M^*, ($M(M^*)$), respectively. The Sum-of-SumSet operation basically sums up the weights of the leaf nodes, i.e., the alerts and the machines, but only adds once for each weight. In other words, if two machines leaf nodes are both asserted, only one λ_{M^*} will be added to the overall threat score for the corresponding information node. This operation ensures that the threat score is less than one for all nodes that could be compromised next as long as $\lambda_{A^*} + \lambda_{M^*} + \lambda_{A(A^*)} + \lambda_{M(M^*)} < 1$.

We provide a few remarks to this Sum-of-SumSet operation. First, a drawback of this approach is that the threat score does not distinguish network entities that are associated with more asserted attack nodes or machines from those associated with less. An extension is currently under investigation to resolve this issue. Second, intuition has that $\lambda_{A^*} > \lambda_{A(A^*)}$ and $\lambda_{M^*} > \lambda_{M(M^*)}$ should define the relationships between the weights. This intuition stems from the assumptions that a computer that has been attacked is likely to be attacked again and that an alert that has occurred is likely to occur again. We test this intuition in Section 4.2.2. Finally, the algorithm relies on the correctness of the models and templates developed by SMEs, which is expected as any threat assessment algorithm may claim. As will be discussed in the next section, the current version of TANDI will provide indication for any abnormality due to possible modeling flaws.

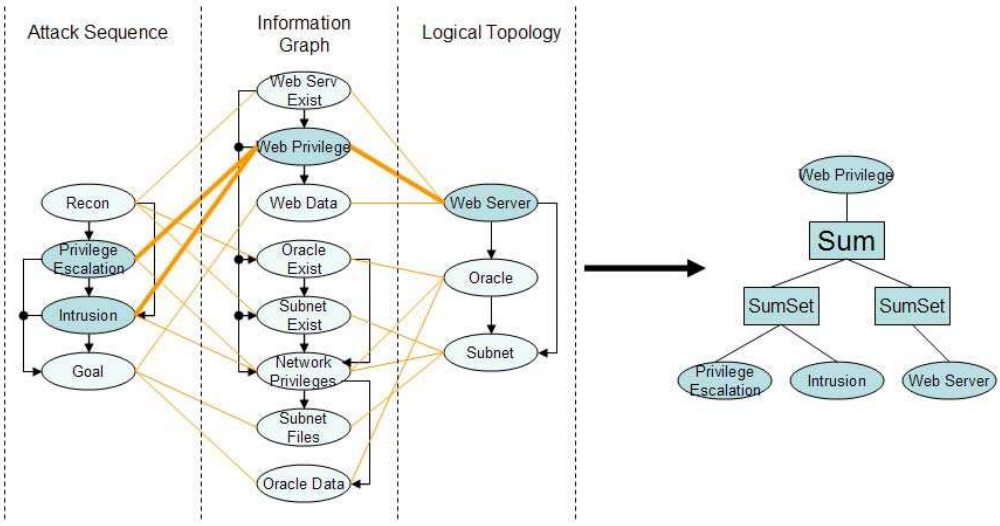

Figure 3. An example feature tree for threat assessment. Note how the relationships between the three graphs correspond to the feature tree.

4. ALGORITHM ANALYSIS VIA SIMULATION

4.1. Simulation Framework

TANDI has been implemented and tested with randomly generated attack sequences on topologies with distinct structures. The implementation includes a GUI and will generate the threat scores in real time as attack events being detected. Recognizing the lacking of simulation work for threat assessment algorithms in the literature, a proposed framework is presented below. This framework may be extended for any threat assessment algorithm in any application domain, as long as the algorithm fulfills the following requirement:

> The algorithm assigns a **threat score**, $0 \leq t_i(e_j^A) \leq 1$, to the ith entity upon the occurrence of the jth event of an attack A, where the threat scores satisfy the following adjectives, which qualitatively describe the threat level of an entity:
>
> - **Compromised**: $t_i(e_j^A) = 1$
> - **Threatened**: $0 < t_i(e_j^A) < 1$
> - **Unthreatened**: $t_i(e_j^A) = 0$

In addition to the threat scores, TANDI records and reports the following metrics.

- **Percent Threatened**: The percentage of non-compromised entities that have a threat score greater than or equal to a threshold, β, due to a specific attack. A high percent threatened metric indicates that the corresponding attacker has the opportunity to compromise many entities, and the analyst may want to focus on treating that attack.
- **Abnormality**: An abnormality is recorded when an entity is compromised with a prior threat score of 'zero.' The number (or percentage) of abnormalities is indicative to the analyst possible flaws in at least one of the following aspects:
  - Sensor readings: the abnormality could be due to a false positive or undetected event.
  - Event correlator: the correlator responsible for level 2 fusion could have falsely correlated (or uncorrelated) events to an attack. A mis-correlation at this level could be due to a stealthy coordinated attack by multiple attackers.

  - Threat assessment model: incomplete or inaccurate models used by the threat assessment algorithm.
  - Insider threat: An algorithm, such as TANDI, is susceptible to not detecting insider threats since it assumes that attacks will originate from outside of the organizational network.

- **Criticality**: A criticality value may be assigned by network analysts to each of the network entity, as an indicator of its importance. The criticality may be used in conjunction with the threat scores to allow the network analyst examining entities that have a high criticality value but, perhaps, a lower threat score.

The above metrics are used for network analyst to make better decisions in anticipation of potential future attacks. An important aspect in designing a threat assessment algorithm, meanwhile, is how to evaluate the performance of the threat assessment algorithm. An ideal threat assessment algorithm with perfect models should generate threat scores that accurately depict the sequence of attack events. In other words, the compromised entity should have the highest threat score with respect to the other threatened entities one step before it is compromised. Based on this intuition, we define the **normalized compromising score**, $\hat{t}_i^*(A)$, as the normalized threat score for entity i one event prior to it being compromised by attack A. That is,

$$\hat{t}_i^*(A) = \{\frac{t_i(e_j^A)}{\max_{k \in I(I^*(e_j^A))} t_k(e_j^A)} \quad | \quad t_i(e_{j+1}^A) = 1, t_i(e_j^A) < 1\}. \tag{2}$$

Note that the normalized threat score is defined only for entities that are compromised. Also, the case of $\max_{k \in I(I^*(e_j^A))} t_k(e_j^A) = 0$, i.e., no uncompromised entity are threatened, will be filtered by TANDI's abnormality tracker.

As attack events happen over time, $\hat{t}_i^*(A)$ will be tracked for each entity and for each generated attack sequence. The average of this tracked variable will be indicative to the accuracy of a threat assessment algorithm. Note that, this is the first attempt, to our knowledge, to define a metric for evaluating threat assessment algorithms. Other metrics may exhibit the performance of threat assessment algorithms from similar or different perspectives. Below are a few examples.

- Average percentile of the compromising threat score.
- Frequency at which the entities with the top five threat scores are compromised next.
- Frequency at which the entities in the top percentile of threat scores are compromised next.

4.2. Simulation Results

TANDI was simulated using three variants of the network topology shown in Figure 4. Note that a more complex network may be used for simulation, but these variants serve the present goal of better understanding threat assessment for cyber attacks. Due to liability, loss of reputation, and competition issues cyber attack data is not publicly available,[19] so simulations are currently limited to artificially created data.

- **Network 1** assumes that all four subnets are completely segmented from each other. The only connection the four subnets have is that they all have one server connecting to the Internet.
- **Network 2** is the same as the first topology, except that all four external servers are fully connected to each other.
- **Network 3** is the same as the second topology, except that all internal computers in all subnets are fully connected.

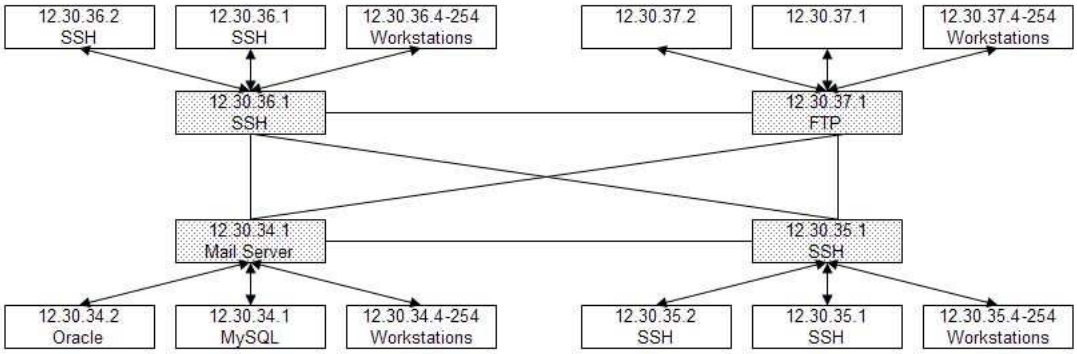

Figure 4. The network topology used for simulations. Blocks with a dotted background are external computers while all others are internal. The corresponding IP address(es) and services running are indicated.

4.2.1. Example Attack and Threat Scores

Figure 5 illustrates an example simulated attack and the evaluated threat scores at each stage of the attack for a subset of network entities in the information graph. Network 1 is used for this example, and the weights used for threat assessment are given in the caption of the figure. The bold number for each entity indicates the threat score one step before the entity was compromised. Note that there is no abnormality in this attack, and TANDI identifies the next potential target with the highest threat score, except for 'SSH Server 35.2-Exist' and 'SSH Server 35.2-User' in Step 2. Note that the average normalized compromising score is 0.74 - indicating TANDI did accurately predict the next compromised entities.

Sequence Number	**0**		**1**		**2**		**3**		**4**	
Attack	Initial		ICMP (Ping)		imap exploit partial body overflow attempt		SSH Password Accepted		DIRECTORY TRAVERSAL	
Computer			12.30.35.1		12.30.35.1		12.30.35.2		12.30.35.2	
	Act	Norm	Act	Norm	Act	Norm	Act	Norm	Act	Norm
SSH Server 35.1-Exist	**0.10**	**1.00**	X	X						
SSH Server 35.1-Root	0.00	0.00	**0.30**	**1.00**	X	X				
SSH Server 35.1-UserFiles	0.00	0.00	0.00	0.00	0.35	1.00	0.35	1.00	0.35	1.00
SSH Server 35.2-Exist	0.00	0.00	0.10	0.33	**0.10**	**0.29**	X	X		
SSH Server 35.2-User	0.00	0.00	0.10	0.33	**0.15**	**0.43**	X	X		
SSH Server 35.2-UserFiles	0.00	0.00	0.00	0.00	0.00	0.00	**0.35**	**1.00**	X	X

Average Normalized Compromised Score: 0.74

Figure 5. Example threat scores during the progression of a cyber attack with the weights of $\lambda_{A^*} = 0.2$, $\lambda_{A(A^*)} = 0.05$, $\lambda_{M^*} = 0.3$, $\lambda_{M(M^*)} = 0.1$. Both the actual and normalized threat scores are shown for each entity. Nodes compromised are indicated with an 'X'.

4.2.2. The Effect of Weights

Recall the weights described in Section 3.3. Ten randomly generated attacks containing no abnormality are created to test the following hypotheses by varying the weights used by TANDI for all three networks.

- **Hypothesis 1**: An attack method that has already attempted is more likely to occur again than a new attack. Therefore, the weights should be set up such that $\lambda_{A^*} > \lambda_{A(A^*)}$.
- **Hypothesis 2**: A computer that has been attacked is more likely to be attacked again than a different computer. Therefore, the weights should be set up such that $\lambda_{M^*} > \lambda_{M(M^*)}$.

The randomly created ten attacks contains realistic alerts from Snort and Dragon IDSs as well as system logs. This set of tests is used to determine how the assignment of the weights affects the average of the normalized compromising scores. The average of the normalized compromising scores are shown in Figure 6 for different topologies with different weight combinations.

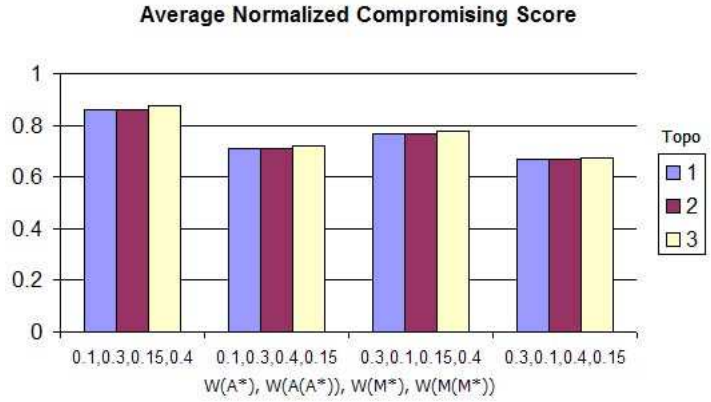

Figure 6. Average of the normalized compromising scores for a simulation of normal attacks.

While TANDI did provide acceptable to good average normalized compromising scores, between 0.67 to 0.86, for the use of different weight combinations, our hypotheses seem to be refuted based on these results: higher $\lambda_{A(A^*)}$ and $\lambda_{M(M^*)}$ actually exhibit better performance. An in-depth analysis of the attacks provides interesting insights towards network dependent threat assessment. Most of the attacks in this test set focus on subnets 34 and 37, which contain servers with different services running. Since different services are running on different computers, a previously executed exploit could not be used on the successor machine. Therefore, a high value of λ_{A^*} does not help to predict the next attack event. Similarly, because the test network has each computer run a single service, the number of successive attack events for the same computer is not as many as originally expected. If, however, the threat assessment is performed on the scale of subnets, i.e., network IDSs are used instead of host-based IDSs, one should expect a relatively long sequence of attack events appearing on the same subnet - the original hypothesis 2 should still hold. This analysis leads to the following two revised hypotheses, which will need to be tested over a larger set of data on many different networks.

- **Revised Hypothesis 1**: In the case where networks contain highly similar computers, a weight assignment of $\lambda_{A^*} > \lambda_{A(A^*)}$ should outperform $\lambda_{A^*} < \lambda_{A(A^*)}$. In the case where networks have a wide array of services running across different computers, the opposite weight assignment should yield a better threat prediction.
- **Revised Hypothesis 2**: In the case where mainly network-based IDSs are used to detect attacks, $\lambda_{M^*} > \lambda_{M(M^*)}$ should outperform $\lambda_{M^*} < \lambda_{M(M^*)}$. In the case where host-based IDSs and extensive system logs are used to detect attacks, the opposite weight assignment should yield a better threat prediction.

4.2.3. Abnormalities/Insider Threats

Two more sets of data are generated to test how TANDI handles abnormalities. These two attack sets will be compared against the baseline Set 0 with no abnormality - the one used for the analysis in the previous subsection. Similar to those in Set 0, attacks from Set 1 will always start by attacking an external machine, but a network entity that does not belongs to $I(I^*(e_j^A))$ may be compromised next in the middle of the attack. Set 2 takes one step further and contains attacks that start at an internal machine, representing insider threats. The weights used here are $\{\lambda_{A^*}, \lambda_{A(A^*)}, \lambda_{M^*}, \lambda_{M(M^*)}\} = \{0.3, 0.1, 0.4, 0.15\}$.

Figure 7 shows that TANDI predicts very poorly with abnormalities, and even worse with the presence of insider threats. These results are expected because TANDI assumes that attacks will follow the possible courses

of action defined by the logical topology. The abnormalities happen when the hackers take unexpected actions or if there are IDS failures. The current implementation of TANDI will alarm the network analyst for the abnormalities, whenever a network entity is recorded a normalized compromising score of 0. This indicator will allow the network analyst to examine the potential source of insider threats, to identify coordinated attacks, or to revise the logical topology model and the IDS setup.

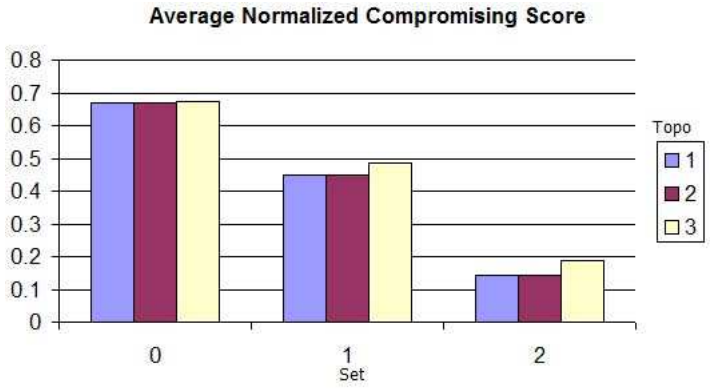

Figure 7. Average of the normalized compromising scores for the set of normal attacks (set 0), a set of attacks with abnormalities (set 1), and a set of insider threats (set 2).

It should also be noted from Figure 7 that Network 3 consistently exhibits better performance than the other two networks. This is due to the high connectivity of topology 3, thus created fewer outlets for abnormalities. While this may indicate that a highly connected topology provides a better assessment using TANDI, the percent threatened for topology 3 (between 55%-64%) is always higher than that of topologies 1 and 2 (between 24%-47%) with $\beta = 0$. The network analyst would interpret the percent threatened metric as the opportunity of the hacker. Since a highly segmented network consistently yielded lower percent threatened results, a highly segmented network provided for less opportunity to the hacker.

5. CONCLUSION AND FUTURE WORK

To maintain the integrity of critical network information and operation, a novel threat assessment scheme, TANDI, is proposed to predict future attacker actions. TANDI reduces the modeling complexity by separating the modeling of attacker's opportunity (logical topology), capability (attack sequence), and intent (information graph). A level 3 fusion rule is introduced to fuse information provided by the logical topology and the attack sequence, so as to determine the threat scores of network entities in real time. TANDI has been implemented and tested via simulation with randomly generated attack sequences. The evaluation framework is the first, to our knowledge, proposed for evaluating a threat assessment algorithm. The results demonstrate that TANDI predicts accurately the threatened entities for attacks containing no insider threats. In the presence of insider threats, coordinated attacks, sensor failure, or incomplete modeling, TANDI will provide real-time indicators to these abnormalities. The use of TANDI should allow network analyst to have a better anticipation in reaction to cyber attacks.

The framework of TANDI may be applied to other application domains. It is expected that TANDI will be particular efficient for situations where there are large number of unprofiled attackers, who continuously update their attack strategies - the characteristics of, e.g., cyber attacks and asymmetric warfare. TANDI is also currently under investigation for improvement. In particular, a better fusion rule is needed to *appropriately* adjust the threat score of an entity if the attacker is more capable and/or sees more opportunity to compromise it. Moreover, a dynamic assignment of weights based on the services running and the historical pattern of the attacker may lead to more accurate threat assessment. An improved TANDI shall be leveraged to distinguish various abnormalities and help identify coordinated attacks and insider threats. Continuous effort will also be

placed on seeking larger simulation data sets and more representative networks. It is our believe that a rigorous evaluation framework will be essential to the development of a threat assessment algorithm.

ACKNOWLEDGMENTS

The authors would like to thank John Salerno, Doug Boulware, and George Tadda at AFRL for motivating the work and providing valuable input towards the design of TANDI, Adam Stotz and Rich Giomundo at the University of Buffalo for offering their expert knowledge on cyber attacks and ECCARS, and Jason Kistner and Michael Kuhl for sharing their experience on developing a simulator for cyber attacks.

REFERENCES

1. S. Noel, E. Robertson, and S. Jajodia, "Correlating intrusion events and building attack scenarios through attack graph distances," in *Proceedings of ACSAC*, December 2004.
2. M. Sudit, A. Stotz, and M. Holender, "Situational awareness of a coordinated cyber attack," in *Proceedings of SPIE*, pp. 114–129, March 2005.
3. O. Dain and R. K. Cunningham, "Fusing a heterogeneous alert stream into scenarios," in *Proceedings of ACM Worksop on Data Mining and Security*, December 2001.
4. Sourcefire, "Snort: an open source network intrusion prevention and detection system." `http://www.snort.org`.
5. Enterasys, "Enterasys intrusion defense." `http://www.enterasys.com/products/ids/`.
6. O. I. Assurance, "Daiwatch home page." `http://www.daiwatch.com/`.
7. E. Little, G. Rogova, and A. Bourry-Brisset, "Theoretical foundations of threat ontology for data fusion applications," Tech. Rep. TR-2005 -269, DRDC-Valcartier, November 2005.
8. D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," in *Proceedings of the IEEE*, **85**, pp. 6–23, January 1997.
9. J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz, and F. White, "Revisions and extensions to the jdl data fusion model II," in *Proceedings of The 7th International Conference on Information Fusion*, pp. 1218–1230, June 2004.
10. C. Phillips and L. P. Swiler, "A graph-based system for network-vulnerability analysis," in *Proceedings of the 1998 workshop on New security paradigms*, pp. 71–79, ACM Press, (New York, NY, USA), 1998.
11. S. Vidalis and A. Jones, "Using vulnerability trees for decision making in threat assessment," Tech. Rep. CS-03-2, University of Glamorgan, School of Computing, June 2003.
12. J. Allanach, H. Tu, S. Singh, P. Willett, and K. Pattipati, "Modeling threats," *IEEE Potentials* **23**(3), pp. 18–21, 2004.
13. Y. Liu and H. Man, "Network vulnerability assessment using bayesian networks," in *Proceedings of Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security*, **5812**, pp. 61–71, March 2005.
14. N. Ye, Y. Zhang, and C. M. Borror, "Robustness of the markov-chain model for cyber-attack detection," *IEEE Transactions on Reliability* **53**, pp. 116–123, Mar. 2004.
15. J. Allanach, H. Tu, S. Singh, P. Willett, and K. Pattipati, "Detecting, tracking and counteracting terrorist networks via hidden markov models," in *Proceedings of IEEE Aerospace Conference*, pp. 3246–3257, Mar. 2004.
16. K. N. Ross and R. D. Chaney, "Hidden markov models for threat prediction," in *Proceedings of SPIE*, **4051**, pp. 300–311, 2000.
17. Q. Changwen and H. You, "A method of threat assessment using multiple attribute decision making," in *Proceedings of 6th International Conference on Signal Processing*, **2**, pp. 1091– 1095, August 2002.
18. Cisco Systems Inc., *netForensics: Report Guide*, April 2003. `http://www.cisco.com/application/pdf/en/us/guest/products/ps5209/c1626/ccmigration_09186a008019d567.pdf`.
19. C. Taylor, A. Krings, and J. Alves-Foss, "Risk analysis and probabilistic survivability assessment (RAPSA): An assessment approach for power substation hardening," in *Proceedings of ACM Workshop on Scientific Aspects of Cyber Terrorism*, (Washington, D.C.), November 2002.