Intrusion Activity Projection for Cyber Situational Awareness

Shanchieh J. Yang[1], Stephen Byers[1],
Jared Holsopple[2], Brian Argauer[1], Daniel Fava[1]
[1]Department of Computer Engineering
Rochester Institute of Technology
[2]Center of Multisource Information Fusion
State University of New York at Buffalo

***Abstract*—Previous works in the area of network security have emphasized the creation of Intrusion Detection Systems (IDSs) to flag malicious network traffic and computer usage. Raw IDS data may be correlated and form attack tracks, each of which consists of ordered collections of alerts belonging to a single multi-stage attack. Assessing an attack track in its early stage may reveal the attacker's capability and behavior trends, leading to projections of future intrusion activities. Behavior trends are captured via Variable Length Markov Models (VLMM) without predetermined attack plans. A virtual terrain schema is developed to model network and system configurations, and used to estimate critical elements and vulnerabilities exposed to each attacker given his/her progress. Experimental results show promises for these proactive measures in ensuring continuous and critical cyber operations.**

I. Introduction

The pervasion of computer networks has amplified the importance of cyber security ranging from personal life to homeland security. Information and cyber security are multifaceted and entail the provision of user accounts and passwords to protect data, the encryption of communication mediums, the imposition of network access rules through firewalls, etc. In addition to these preventive methods, cyber security relies heavily on Intrusion Detection Systems (IDSs), which work by performing anomaly detection or pattern recognition on network traffic and/or host activities. Many IDS solutions have been proposed, and taxonomies can be found in [1] and [2]

As network complexity and size grow, the number of IDSs deployed and alerts also grow and often overwhelm human analysts in critical times [3], [4]. Several methods for creating comprehensive alert reports have been proposed as potential solutions to this problem. Some of these efforts were in the area of alert aggregation [5], [6] or alert correlation [7], [8], [9], [10], [11]. Alert correlation essentially finds IDS alerts that are related and organizes them into ordered collections, often called attack tracks. These attack tracks, borrowing the notion from object tracking, may be viewed as observed 'virtual trajectories' of cyber attacks. This work discusses ways to project the progression of cyber attacks, enabling analysts to proactively prevent plausible future intrusion activities.

Three earlier studies have attempted to project likely future attack actions. Qin and Lee [12] proposed to adaptively update Bayesian networks and used them to model and predict attack plans of actions. It is unclear, however, how the various attack plans can be created and matched efficiently with real-time observations. Li *et al.* [13] utilized a data mining technique to extract the sequential relationships from observed attack actions in training data. Likely sequences were used to assess potential threats in real time by matching with newly observed alert sequences. Their work, however, consider the sequential order of *all* alerts, regardless whether they belonged to the same multistage attack. No result was published to suggest how well Li *et al.* 's technique predicts next likely actions. Holsopple *et al.* [14] proposed to separate the threat projection process into two sub-tasks: analyzing the attack methods versus analyzing the network topology. The outcomes of the two sub-tasks were then combined to determine the threatened entities in the network.

The challenge of projecting attack actions comes from the fact that not only cyber attacks are diverse and constantly changing, but so are the network and system configurations. Relying on a set of rigid attack plans or assessing all aspects at once may lead to inconclusive or, worse yet, misleading projection results. This work discusses various methods that examine different aspects of cyber attacks and aims at providing better situational awareness by showing the analysts where and how the attacks might progress in the near future. Section II will first illustrate the cyber intrusion projection problem. Section III discusses a graph-based virtual terrain model, representing network and system configurations, and its uses for threat projection. Section IV presents our approach on capturing and projecting attacker's behavior trends. Section V concludes the paper.

II. Elements for Cyber Intrusion Projection

The cyber intrusion projection system envisioned here is a real-time system that shows the analysts plausible futures as attack actions are observed. An alert correlation system is assumed to take the observed attack actions, in the form of IDS alerts, and produces ordered collection of alerts belonging to the same multistage attack. These ordered collection of alerts, called attack tracks, will be the basis for projecting the next plausible attack actions the network might see. This work does not intend to predict new attack methods that have not been observed before. The methodology described here aims at identifying plausible futures of *ongoing* attacks.

Recognizing the many aspects involved in projecting future attack actions, this work leverages the concepts put forth by Holsopple *et al.* [14] and the Capability-Opportunity-Intent (COI) model for threat assessment [15]. In addition, we conjecture that behavior trends is also a critical element, as hackers might execute/import malicious codes in series, which exhibit patterns.

As a result, there are four basic elements for projecting cyber attack actions:

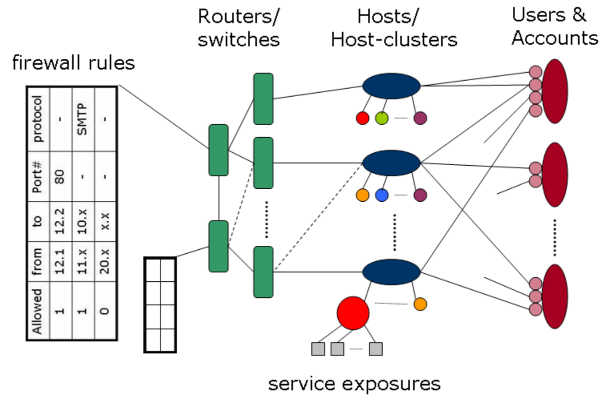
- **Capability**: The intrusion methods the attacker has use is indicative to the types of vulnerabilities he is capable of exploiting.
- **Opportunity**: Given the already compromised entities or privileges by a given attack, originally hidden entities or vulnerabilities may be exposed and give opportunities to the attacker.
- **Intent**: The intent of a cyber attacker can be quite diverse, perhaps making it impossible to estimate. Instead of assessing the true intent, cyber intrusion projecting may examine the criticality of network entities and operations to determine the worst-case intent of the attacker.
- **Behavior trend**: The patterns exhibited in the observed cyber attack actions. The pattern may exist in attack methods, types of services or OS attacked, subnets visited, protocols exploited, etc.

A cyber intrusion projection system may address one or more elements above. In the case where individual elements are addressed via different algorithms, an all-encompassing method may be required to combine the estimations. The combination method is beyond the scope of this paper. The next two sections describe the proposed approaches and how they address the four elements above.

III. Virtual Terrain Assisted Assessment

A reasonably secured network typically has multiple access domains, where direct access to internal and often critical domains or subnets is prohibited. Serious cyber attacks, thus, need to exploit different system vulnerabilities and progress through multiple domains. Reasoning on the progress made by a cyber attack shall benefit from a virtual terrain model that represents the logical accessibility from one access domain to another. Most importantly, the virtual terrain should model the system and network configurations, including their vulnerabilities that may be exposed as the attacker compromises one or more systems in the network.

Vidalis and Jones [16] proposed the use of a vulnerability tree to identify the types of attacks an attacker could perform to accomplish a goal. Their model requires a separate vulnerability tree for each possible goal, which could be potentially numerous. Philips and Swiler [17] and Liu and Man [18] suggested the use of a Bayesian network to model the vulnerabilities. Their model assumes acyclic graphs, which implies that bi-directional connections between hosts must be modeled in separate acyclic graphs. Massicote *et al.* [19] discussed ways of introducing contextual information to cross examine reported IDS alerts, therefore reducing false positives. Their experiences suggested that contextual information may be derived by utilizing Snort [20], Nessus [21] and Bugtraq [22]. This work, developed independently of Massicote's work, shares some similar ideas, yet provides additional network connectivity and privilege information for situation assessment and threat projection.



Allowed	from	to	Port#	protocol
1	12.1	12.2	80	-
1	11.x	10.x	-	SMTP
0	20.x	x.x	-	-

Fig. 1. A graphical representation of the virtual terrain model.

A. A graph-based virtual terrain model

A virtual terrain modeling approach is proposed in [23]. The virtual terrain model is composed of three key component sets: hosts or host-clusters (H), switches or routers (R), and users (U). Figure 1 is a graphical representation of the three-tier definition and the associations between the switches and hosts and those between the users and hosts. Hosts, which are used to model regular user machines, uniformly configured host-clusters, and various servers, are interconnected via a tree of switches/routers. The hosts are the leaf nodes for this part of the graph. Note that common practice would have an enterprise network configured as a spanning-tree even though there are physical loops and redundant paths, as represented by the dashed lines shown in Figure 1. Each user node may have one or more accounts, which are depicted by the small circles shown next to the user nodes. Some accounts are local to a single host node, and some are associated with multiple hosts. The accounts and the hosts form a bipartite graph.

Also shown in Figure 1 are the key attributes associated with the hosts, switches, and users. Each switch has a list of firewall rules defining the allowed or banned protocols and IPs. Each host also contains a list of banned port-numbers, protocols, IPs, or a combination of the three (not shown in the figure). Remote services and local applications running on each host node are specified, and each service is defined with a set of 'exposures.' The exposures correspond to the known alert descriptions (or vulnerabilities) associated with each service. Note that information such as remote services and service vulnerability exposures may be obtained from automatic scanning tool such as Nessus [21] and NMap [24], and can reference to databases such as the National Vulnerabilities Database (NVD) [25] and the Common Vulnerabilities and Exposures (CVE) Dictionary [26].

B. Assessing capability and opportunity using virtual terrain

It is difficult, perhaps impossible, to estimate an attacker's exact capability in exploiting system vulnerabilities. An extremely conservative estimate is to assume that all attackers are able to execute all exploitation methods. A more optimistic approach is to assume that an attacker is able to exploit services that he/she has successfully attacked before. The latter approach allows a significant reduction in projecting possible vulnerabilities each attacker can exploit and is used for this work.

The key tasks in determining the optimistic capability set for each attack track are (1) finding the services associated with the executed attacks and (2) determine the executed attacks that are actually successful. Note that the capability is determined at the service level, *e.g.,* SMTP and DNS, but not at the specific exploit level. This is to reflect that attackers typically know various methods to exploit the same service; including only the specific executed attack methods will be overly optimistic. Determining successful attacks are also important because unsuccessful attacks do not constitute sufficient evidences for the attacker's capability.

The two tasks are accomplished by mapping the correlated alerts to the service exposures of the corresponding target defined in the virtual terrain. First, for each attack track t_j, the successful attacks are filtered by examining the firewall rules along the path from the source to the target, and the exposures sets in the targeted hosts. Note that IDSs and alert correlators may have filtered out false positive alerts, but the proposed system does not make such assumption. Only the service exposures associated with successful attack actions will be asserted for each t_j. The service instances $R(t_j)$ can then be identified through the asserted exposures. The other service instances in the network that are vulnerable to the same attacks can be found via the threatened service type $S(t_j) = \{s | s = S(r), \forall r \in R(t_j)\}$.

Capability alone is not suitable to provide estimates of future attack actions. The next question is how to narrow down to the hosts that are 'exposed' to each attack, *i.e.,* the opportunities. Note that if traffic is allowed to freely move within the network, regardless of subnet domain, protocol, or port numbers, any host node is virtually an immediate neighbor of all other nodes. That is, the virtual terrain is equivalent to a complete graph of host nodes. In which case, all nodes running service instances belonging to $S(t_j)$ will be threatened. In reality, networks are reasonably secured with firewall rules, permission, and banned lists defined to segregate the access domains. By examining the firewall rules of the switches and the permission lists of the hosts that are accessible from the attacked hosts, the assessment algorithm finds the logical entities that are exposed to each attack. Note that the entities can be hosts or users. The discussion in this proposal shall focus on hosts.

For each attack track, the opportunity assessment will instantiate a trajectory graph from the virtual terrain model to represent its progress. This trajectory graph will be dynamically updated when a new alert is reported to correlate with the attack track. The search for exposed entities can be found by taking advantage of specific graph properties. For example, the hosts interconnected via a tree of switches can be found in $O(\log(n))$ time where n is the number of switches. One proposed effort will be to develop an efficient algorithm that finds exposed nodes from a set of attacked nodes when they are interconnected with firewall-rule defined paths within a general mesh.

The current algorithm assigns a 'threat score' to each host that is susceptible to the demonstrated capability and exposed to the current progress of each attack track. These threat scores, ranging between 0 and 1, represent the relative belief that an entity will be attacked soon. A threat score of 1 means that the entity is already compromised. No optimal methodology has been identified to determine how the threat scores should be assigned in order to maximize the projection accuracy. A heuristic scoring scheme is currently implemented to provide the analysts with quantitative references in projecting threatened entities in the network.

A prototype of the virtual terrain model and the associated algorithms for cyber intrusion projection has been implemented and tested. Note that most research work on cyber security has focused on intrusion detection and alert correlation; therefore, most datasets available are composed of uncorrelated alerts with little or no ground truth in terms of network configuration or attack tracks. Examples of such datasets include the ones from the MIT Lincoln Lab [27], [28], KDD Cup 99 [29], and Defcon [30]. Two mock-up networks are, therefore, developed and experimented with 15 random attack sequences each. Table I shows the experimental results for the two networks. Network 1 has four subnets and each subnet has access to two dedicated servers and four shared centralized servers. This is to represent the case with segmented departments. Network 2 has 3 subnets, each of which can access only 1 dedicated server but share most others. The subnets in Network 2 are hidden behind layers of tightly controlled servers, including a server farm of 10.

	#Servers	#Subnets	$AvgCS$	$AvgAR$
Network 1	12	4	71.5%	86.2%
Network 2	19	3	89.6%	52.7%

TABLE I
EXPERIMENTAL RESULTS FOR PROJECTING USING VIRTUAL TERRAIN.

The Average Compromising Score ($AvgCS$) shown in Table I is the average threat scores of entities that are about to be compromised next. Intuitively, a good projection scheme will give high threat score to entities just before they are compromised. Therefore, the higher the $AvgCS$, the better the projection accuracy. Reasonably good $AvgCS$ is shown for both networks. Network 1 sees a lower $AvgCS$ because the hosts and servers in each subnet are all 1 server away from the Internet and, thus, are easily susceptible to attacks. This makes it harder to differentiate between more and less severely threatened entities. Network 2, on the other hand, sees close to 90% $AvgCS$. This exceptional performance is primarily due to the tightly configured server and subnet access of Network 2; only few vulnerable paths are avaiable to attack internal hosts and servers, and the paths can be quite different from one target to another.

The Average Assessee Reduction ($AvgAR$) in Table I shows the opposite trend as $AvgCS$ does. The metric $AvgAR$ represents the average percentage of entities the system has reduced for the analysts to focus on. The implemented system only shows analysts the entities that have a threat score no less than 0.5. In other words, for experiments done on Network 1, the analysts only need to focus on 13.8% of the entities they would have to examine without the proposed system. Network 2 sees a relatively less reduction of 52.7%. Network 1 sees a better reduction because the subnets are segmented only 1 server away from the Internet; so the reduction of assessee is already high even at the very early stage of an attack.

In general, examining the attacker capability and opportunity, *i.e.,* exposed vulnerabilities, is effective to project most cyber attack actions, particularly for well-managed and secured networks.

C. Assessing intent using virtual terrain

As discuss in Section II, this work does not intend to estimate the true ‘intent’ of attackers. Instead, the proposed approach will estimate the impact of the cyber attacks on the critical entities in the network. More specifically, the use of virtual terrain will help identify the targeted network entities that are critical to network operations or due to the associated data content.

Similar to the process described in Section III-B, each attack track is mapped onto the virtual terrain model by asserting the service exposure that match each alert. Once mapped, the sequences of alerts are transformed into sequences of complex objects, which are associated with network services, hosts, and users. When assessing the impact of the attacks, a heuristic scoring scheme is developed to quantify the relative severity of damage on each network component. This work defines the impact scores ($I_X(x)$) for an entity x as a function of its subcomponents' criticality with respect to x ($c(\cdot, x)$) and a score representing its subcomponents' exposure to current observed attack actions (α_x). An entity can be a host, a service, a user, a subnet or the entire network.

$$I_X(x) = \frac{\sum_{r \in R(x)} c(r,x) \cdot \alpha_r}{\sum_{r \in R(x)} c(r,x)}; \qquad (1)$$

Note that an entity can be a subcomponent of several other entities in the network for assessment purposes. For example, a host may be a subcomponent of a subnet but also a subcomponent of a overall network service.

All variables shown in (1) is normaized to between 0 and 1. Each component's criticality value may be assigned by the analysts as it represent, for example, whether a service is important to the mission of the network. The exposure scores, on the other hand, is iteratively derived from the service vulnerability scores at the bottom level of virtual terrain. A service vulnerability score is asserted if an observed alert is mapped to the corresponding service vulnerability. The service vulnerability scores can be either pre-assigned by the security analysts or derived from different vulnerability scoring systems, such as Microsoft's proprietary scoring system [31], US-CERT [32], SANS [33], and the Common Vulnerability Scoring System (CVSS) [34].

The impact scores can be evaluated for any network entity with respect to any observed attack track, and can be continuously updated as new alerts are reported. At a given stage of an attack, not all attacked services are fully compromised. The proposed scheme will ‘project’ the impact score by calculating according to (1) except that all vulnerabilities of attacked services are assumed to be asserted. This allows the analysts to project the critical impact caused by an attack to each network entity. An example 13-step attack is performed on a third mock-up network that is larger and more secured than the two networks described in Section III-B. Figure 2 shows how the actual impact score and the projected impact score for the Administrator (I_U(Admin)) evolve as the alerts are reported.

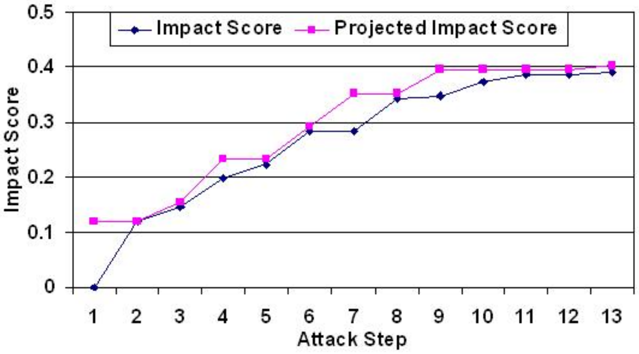


Fig. 2. Actual and projected I_U(Admin) as a 13-step attack progresses.

As can be seen in Figure 2, the actual impact score more or less follows the projected impact score. Similar trends were also observed for many other multistage attacks. A benefit of adopting this impact based estimation is that it is not restricted to the attacker demonstrated capability, and will not be misled by attacker's decoy attacks because the projection is based on what will lead to most critical impact on the network, not based on attacker behavior.

IV. Behavior extraction and projection via sequence modeling

In addition to benefiting from virtual terrain which models capability, opportunity, and intent, projecting attack actions can take advantage of analyzing attacker behavior exhibited in individual attack tracks. Recognizing that cyber attack behavior can be diverse and changing, this work does not use specific attack plans, nor attempt to construct attack plans.

Furthermore, no network specific information is required. Instead, each ordered collection of alerts are converted to a sequence of symbols using only the information contained in the alerts. Depending on the alert field or the combination of fields used, each attack track may be converted to a different sequence of symbols. In current work, the symbols are defined in terms of the attack description (Ω_d), the type or category of attacks (Ω_c), the network protocol used for the attack (Ω_p), and the subnet where the target resides (Ω_t).

With respect to a symbol space Ω, each attack sequence is now transformed to a sequence $s = \{x_1, x_2, x_3, \cdots\}$. The sequential order of symbols in historical data is used to build a model that characterize behavior patterns. The pattern may exist from one symbol to the next, from two consecutive symbols to the next, and so on. This is the intuition behind the use of Variable Length Markov Models (VLMM). Other models, such as Hidden Markov Model, are also possible, but the complexity may be too high to be realized in real-time handling hundreds or thousands or alerts per second.

Markov models may consider sequential relationships in various orders. Consider an ongoing attack with k observed actions $\{x_1, \ldots, x_k\}$. A model of order o gives the probabilities $p_o(x) = P\{x|x_{k-o+1}, \ldots, x_k\}$, $\forall x \in \Omega$. The probabilities may be obtained via historical attack sequences. A sequence of length n will contribute to the building of o^{th} order models for $1 \leq o \leq n$. More specifically, a sequence of length n will provide one sample to the n^{th} order model, two samples to the $(n-1)^{\text{th}}$ order model, ..., and n samples to the 1st order model. A suffix tree is used to record the samples and to store the models of different orders. The suffix tree structure allows one to find a symbol x_{n+1} in $O(n)$ time given a sequence or a context of length n.

Markov Models of different orders can be blended into a VLMM. Let $p_o(x)$ be the probability of an event x happening at a certain context according to a model of order o. A blended probability for the event x can be computed as follows:

$$p(x) = \sum_{o=-1}^{N} w_o \times p_o(x) \qquad (2)$$

where N is the maximum length of a context, w_o is the weight associated with the o^{th} order model, and $\sum_{o=-1}^{N} w_o = 1$. Note that contexts should be penalized by their rarity and rewarded by their specificity. Examples of the weight functions can be found in [35]. Notice that the summation in Eq.(2) starts at -1. The minus-one order model assigns all characters a probability of $1/|\Omega|$ to prevent the *zero frequency problem* [35]. The zero order model holds the frequency count of all $x \in \Omega$ in the training set.

The VLMM model allows us to discover patterns within attack sequences without explicitly defining attack plans. In fact, an ongoing attack sequence may match to patterns from numerous different types of attack sequences before it. VLMM combines the probabilities associated with all matched patterns and produce a best guess.

Experiments have been conducted for the VLMM approach using a dataset created through scripted multi-stage attacks performed on a VMWare network. This dataset contains a total of 1482 attack tracks comprising 10425 alerts. Correlated alerts are sent to the system in the order of their time stamps. Figures 3 and 4 show the moving average prediction accuracy achieved by the VLMM approach when attack description (Ω_d), attack category (Ω_c), network protocol (Ω_p), and destination subnet (Ω_t) are used to define the symbol space. The prediction accuracy is the percentage of occurring symbols fall within a prediction set. The prediction set comprises symbols with the highest probabilities according to the VLMM model. The number of symbols contained in the prediction set varies, with their cumulative probability being no less than 95%. Surprisingly, this 95% cumulative probability correspond to only a small percentage of symbols - less than 6 out of 51 symbols in the case of Ω_d.

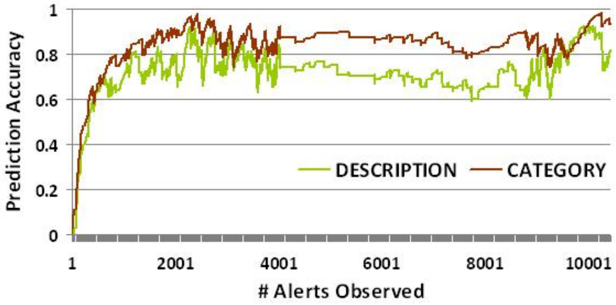


Fig. 3. Prediction accuracy using VLMM as alerts are observed.

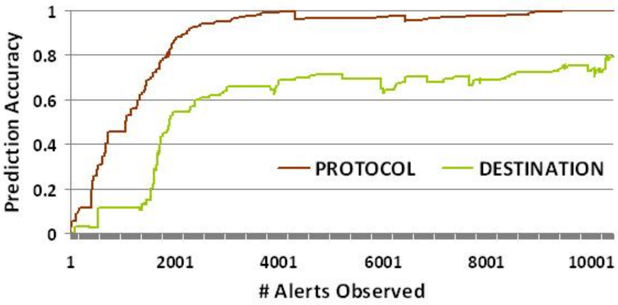


Fig. 4. Prediction accuracy using VLMM as alerts are observed.

In the beginning of the simulation, not all the symbols have occurred and the VLMM model does not have sufficient data to build an accurate model. After the transient period, which ends around 1000∼2000 alerts depending on the symbol definition used, the prediction accuracy fluctuates around 90%, 75%, 98%, and 70% for the cases of using Ω_d, Ω_c, Ω_p, and Ω_t, respectively. Overall, the prediction accuracy is better than expected considering the number of possible symbols each definition can have. The use of attack category is of particular interest since it gives analysts a reasonable scope of projection without worrying the specific vulnerability the attacker will be exploiting. Combining the behavior based projection of attack methods, protocol, and target subnet with the virtual terrain information shall enhance the prediction accuracy even further since infeasible combinations on the given network will be eliminated.

V. Conclusion

Defending against multistage cyber attacks has become a top priority for government, business, as well as individuals. This work proposes to project next actions of ongoing cyber attacks. This proactive measure aims to assist network security analysts with an enhanced cyber situational view of plausible futures. Recognizing the diverse and constant changing nature of cyber attacks and network configurations, this work decomposes the cyber intrusion projection problem into four assessment elements: capability, opportunity, intent, and behavior. A set of algorithms are proposed to analyze these elements independently. More specifically, a graph-based virtual terrain model is utilized to analyze the attacker's capability, exposed vulnerability, and critical target entities, and a VLMM model is used to extract and project behavior patterns. The proposed algorithms are all implemented and able to process alerts in real time. The promising results and observations warrant the potential of our approaches utilizing network-based virtual terrain and behavioral-based VLMMs. An ongoing work is to develop an intelligent combination of the various algorithms under different scenarios, particularly in the presence of stealthy and decoy attacks.

Acknowledgment

The authors would like to thank John Salerno, George Tadda and Michael Hinman for their valuable inputs. This work is funded through the National Center for Multisource Information Fusion (NCMIF) grant under the technical supervision of AFRL/IFEA.

References

[1] S. Axelsson. Intrusion detection systems: A survey and taxonomy. Technical Report 99-15, Chalmers University, 2000.

[2] H. Debar, M. Dacier, and A. Wespi. Towards a taxonomy of intrusion-detection systems. *The International Journal of Computer and Telecommunications Networking*, 31:805–822, 1999.

[3] Tim Bass. Intrusion detection systems and multisensor data fusion. *Communications of the ACM*, 43:99–105, 2000.

[4] J. Goodall, W. Lutters, and A. Komlodi. The work of intrusion detection: rethinking the role of security analysts. In *Proceedings of the Americas Conference on Information Systems*, 2004.

[5] F. Cuppens. Managing alerts in a multi-intrusion detection environment. In *Proceedings of the* 17^{th} *Anual Computer Security Applications Conference*, pages 22–31, 2001.

[6] O. Dain and R. Cunningham. Fusing a heterogeneous alert stream into scenarios. In *Proceedings of ACM Workshop on data mining for security applications*, 2001.

[7] Samuel T. King, Z. Morley Mao, Dominic G. Lucchetti, and Peter M. Chen. Enriching intrusion alerts through multi-host causality. In *Proceedings of the 2005 Network and Distributed System Security Symposium (NDSS'05)*, February 2005.

[8] Peng Ning, Yun Cui, and Douglas Reeves. Analyzing intensive intrusion alerts via correlation. In *Proceedings of the 9th ACM Conference on Computer & Communications Security*, pages 245–254, 2002.

[9] Adam Stotz and Moises Sudit. INformation Fusion Engine for Real-time Decision making (INFERD): A perceptual system for cyber attack tracking. In *Proceedings of 10th International Conference on Information Fusion*, July 2007.

[10] Alfonso Valdes and Keith Skinner. Probabilistic alert correlation. In *Recent Advances in Intrusion Detection (RAID 2001)*, number 2212 in Lecture Notes in Computer Science. Springer-Verlag, 2001.

[11] F. Valeur, G. Vigna, C. Kruegel, and R. A. Kemmerer. Comprehensive approach to intrusion detection alert correlation. *IEEE Transactions on Dependable and Secure Computing*, 1(3):146–169, July-Sep 2004.

[12] Xinzhou Qin and Wenke Lee. Attack plan recognition and prediction using causal networks. In *Proceedings of the* 20^{th} *Annual Computer Security Applications Conference*, pages 370–379, 2004.

[13] Z. Li, J. Lei, L. Wang, and D. Li. Assessing attack threat by the probability of following attacks. In *Proceedings of the International Conference on Networking, Architecture, and Storage*, pages 91–100, 2007.

[14] Jared Holsopple, Shanchieh Jay Yang, and Moises Sudit. TANDI: Threat assessment for networked data and information. In *Proceedings of SPIE Defense and Security Symposium: Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications Conference*, volume 6242, April 2006.

[15] Alan Steinberg. Open interaction network model for recognizing and predicting threat events. In *Proceedings of Information, Decision and Control (IDC) '07*, pages 285–290, Febuary 2007.

[16] S. Vidalis and A. Jones. Using vulnerability trees for decision making in threat assessment. Technical Report CS-03-2, University of Glamorgan, School of Computing, June 2003.

[17] Cynthia Phillips and Laura Painton Swiler. A graph-based system for network-vulnerability analysis. In *Proceedings of the 1998 workshop on New security paradigms*, pages 71–79, New York, NY, USA, 1998. ACM Press.

[18] Yu Liu and Hong Man. Network vulnerability assessment using Bayesian networks. In *Proceedings of Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security*, volume 5812, pages 61–71, March 2005.

[19] Frdric Massicotte, Mathieu Couture, Lionel Briand, and Yvan Labiche. Context-based intrusion detection using Snort, Nessus and Bugtraq databases. In *Proceedings of the Third Annual Conference on Privacy, Security and Trust, Fredericton*, October 2005.

[20] Sourcefire. Snort: an open source network intrusion prevention and detection system. `http://www.snort.org`, 2007.

[21] Tenable Network Security, Inc. Nessus vulnerability scanner. `http://www.nessus.org/`.

[22] SecurityFocus. Bugtraq vulnerability database. `http://www.securityfocus.org/bid`, 2006.

[23] Jared Holsopple, Brian Argauer, and Shanchieh Jay Yang. Virtual terrain: a common representation of a computer network. In *Proceedings of SPIE Defense and Security Symposium: Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security Conference*, volume 6973, March 2008.

[24] Insecure.com. Nmap (Network Mapper): a free open source utility for network exploration or security auditing. `http://insecure.org/nmap`.

[25] National Institute of Standards and Computer Security Division Technology. National Vulnerabilities Database (NVD). `http://nvd.nist.gov/nvd.cfm`.

[26] Mitre. Common Vulnerabilities and Exposures CVE dictionary. `http://cve.mitre.org/`.

[27] Kristopher Kendall. A database of computer attacks for the evaluation of intrusion detection systemdetection systems. Master's thesis, Massachusetts Institute of Technology, 1999.

[28] MIT Lincoln Laboratory. Cyber attack with background traffic data set (1998, 1999, 2000). `http://www.ll.mit.edu/IST/ideval/index.html`.

[29] KDD Cup. KDD Cup data, 1999. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[30] DEFCON conference. DEFCON Capture the Flag (CTF) contest. http://www.defcon.org/.

[31] Microsoft Corporation. Microsoft security response center security bulletin severity rating system. `http://www.microsoft.com/technet/security/bulletin/ rating.mspx`.

[32] United States Computer Emergency Readiness Team (US-CERT). US-CERT vulnerability note field descriptions. `http://www.kb.cert.org/vuls/html/fieldhelp`.

[33] SANS Institute. SANS critical vulnerability analysis archive. `http://www.sans.org/newsletters/cva/`.

[34] Peter Mell, Karen Scarfone, and Sasha Romanosky. A complete guide to the Common Vulnerability Scoring System (CVSS) version 2.0. `http://www.first.org/cvss/cvss-guide.html`, 2007.

[35] Timothy C. Bell; John G. Cleary; Ian H. Witten. *Text Compression*. Prentice Hall, 1990.