This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2008 proceedings.

A Novel Quantitative Approach For Measuring Network Security

Mohammad Salim Ahmed[†]
salimahmed@utdallas.edu

Ehab Al-Shaer[‡]
ehab@cs.depaul.edu

Latifur Khan[†]
lkhan@utdallas.edu

***Abstract*—Evaluation of network security is an essential step in securing any network. This evaluation can help security professionals in making optimal decisions about how to design security countermeasures, to choose between alternative security architectures, and to systematically modify security configurations in order to improve security. However, the security of a network depends on a number of dynamically changing factors such as emergence of new vulnerabilities and threats, policy structure and network traffic. Identifying, quantifying and validating these factors using security metrics is a major challenge in this area. In this paper, we propose a novel security metric framework that identifies and quantifies objectively the most significant security risk factors, which include existing vulnerabilities, historical trend of vulnerability of the remotely accessible services, prediction of potential vulnerabilities for any general network service and their estimated severity and finally policy resistance to attack propagation within the network.**

We then describe our rigorous validation experiments using real-life vulnerability data of the past 6 years from *National Vulnerability Database (NVD)* [10] to show the high accuracy and confidence of the proposed metrics. Some previous works have considered vulnerabilities using code analysis. However, as far as we know, this is the first work to study and analyze these metrics for network security evaluation using publicly available vulnerability information and security policy configuration.[1]

I. Introduction

Each network can be regarded as a collection of systems that provide various services to its clients or users. When considering security, the measurement approach must be able to evaluate each of these systems individually. Apart from service vulnerabilities, global security policy configuration of the network defines how deeply security breaches will affect the network.

In this paper, we present a new framework for network security policy evaluation that can quantitatively measure the security of a network based on two critical risk aspects - the risk of having a successful attack and the risk of this attack being propagated within the network. All these risk factors and their scope can be seen in Figure 1. We have, therefore, modeled our framework as a combination of two parts. The first part measures the security level of the services based on vulnerability analysis. The analysis considers the presence of existing vulnerabilities, the dormant risk based on previous history of vulnerabilities and future predictions. Our second part measures the security of the network from policy perspective. The degree of penetration or impact of successful attacks is considered in this measure. From the service risk components, we get the threat likelihood and at the same time the attack propagation provides us with the risk impact to the network of interest. When the risk impact is combined with the cost of the damage, we get the contribution of network policies in the total risk. The threat likelihood and the risk impact obtained from the two parts are significant indications of total network risk.

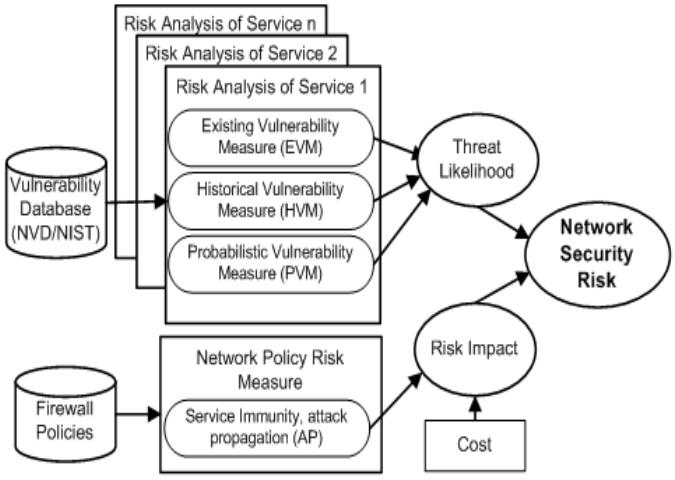


Fig. 1. Network risk measurement framework

The proposed evaluation metrics can help in comparing security policies with each other to determine which policy is more secure. It is also possible to judge the effect of a change to the policy by comparing the security metrics before and after the change. This framework will also allow networks to be evaluated and hardened automatically by constantly monitoring dynamic changes in the network and service vulnerabilities. Using the National Vulnerability Database published by NIST, we performed extensive evaluation of our proposed model. The results show high level of accuracy in the evaluation.

Our framework is novel as the existing tools and previous research works only scan or analyze a given network to find out allowed traffic patterns and existence of vulnerabilities, and do not predict the future security state of the network. Works that do try to predict future vulnerabilities are only beginning to emerge [4]. But this work requires inside knowledge of the studied software. Our prediction model is general and can work using only publicly available data.

[†]Department of Computer Science, The University of Texas at Dallas.

[‡]School of Computer Science, Telecommunications and Information Systems, DePaul University.

[1]This research was supported in part by National Science Foundation under Grant No. CT- 0716723. Any opinions, findings, conclusions or recommendations stated in this material are those of the authors and do not necessarily reflect the views of the funding sources.

The organization of the paper is as follows. First, we discuss related works in Sect. II. Then, we discuss the service risk analysis part and the network policy risk of our security metric in Sect. III and Sect. IV respectively. We then present our experiments and results in Sect. V and conclusions in Sect. VI.

II. Related Work

Keeping the network secure is of utmost importance to any organization. Many organizational standards have evolved to evaluate the security of an organization [12]. Proffessional organizations as well as tools like Nessus, Retina and others that perform vulnerability analysis also exist. There has also been some research in the security policy evaluation and verification. A. Atzeni et al. discuss the importance of security metrics in [6]. Evaluation of VPN, Firewall and Firewall security policies include [2], [3], [7], [8]. Attack graph [5] is another technique that is used to assess the risks associated with network exploits.

There has been some research focusing on the attack surface of a network. Mandhata et al. in [9] have tried to find the attack surface from the attackability of a system. In [11] Pamula propose a security metric based on the weakest adversary (i.e. the least amount of effort required to make an attack successful). In [4], Alhazmi et al. present their results on the prediction of vulnerabilities. Sahinoglu et al. propose a framework in [13] for calculating risk. They consider present vulnerabilities in terms of threat represented as probability of exploiting a vulnerability and also the lack of counter-measures. But all these work do not represent the total picture as they predominantly try to find existing risk and do not address how risky the system will be in the near future or how policy structure or network traffic would impact on security.

In our previous work, published as a short paper [1], we performed a preliminary investigation of measuring the existing vulnerability and explored some historical trends. This work was still limited in analysis and scope as many other evaluation measures that we have discussed in this paper were left unexplored.

III. Network Service Risk Analysis

In this section, we describe and discuss in detail the calculation method of our vulnerability analysis for service risk measurement. This analysis comprises of Existing Vulnerability Measure, Historical Vulnerability Measure and Probabilistic Vulnerability Measure.

Existing vulnerability is important for networks in which the services are left unpatched or where there are no known patches available. Also, when a vulnerability is discovered, it takes time before a patch is introduced for it. During that time the network and services are vulnerable to outside attack. The *Existing Vulnerability Measure (EVM)* measures this risk toward the services within the network.

The EVM is a measure of the severity of the vulnerabilities present in the network. It has been studied and formalized in our previous work [1]. There are a multitude of commercial and open source network vulnerability scanning softwares for finding vulnerabilities present in the network for such measurement purposes.

Therefore, we can use EVM as part of the proposed framework based on our previous work or other similar works. However, finding the risk to fully patched services based on historical trend and future prediction is one of the major challenges where we contribute in this paper.

A. Historical Vulnerability Measure

The *Historical Vulnerability Measure (HVM)* measures how vulnerability prone a given service has been in the past, based on the vulnerability history report of the service. Considering both the frequency and recency of the vulnerabilities, we combine the severity scores of past vulnerabilities so that a service with a high frequency of vulnerabilities in the near past has a high HVM.

We first need to determine the HVM of individual services. We divide the vulnerabilities of service S, into three groups – $HV_H(S)$, $HV_M(S)$ and $HV_L(S)$ for vulnerabilities that pose high, medium and low risks. We assign progressively high to low weight to them. During evaluation, the vulnerabilities discovered a long time ago should carry smaller weight, because with time these would be analyzed and patched. Our analysis of vulnerabilities found the service vulnerability to be more dependent on recent vulnerabilities. And this dependency is not linear. Hence, we apply an exponential decay function of vulnerability age.

$$DV(v_i) = SS(v_i).e^{-\beta Age(v_i)} \tag{1}$$

Here the parameter β controls how fast the factor decays with age. In computing the HVM of individual services, we sum up these decayed scores in each class, and take their weighted sum. Finally, we take its natural logarithm to bring it to a more manageable magnitude. Mathematically,

$$HVM(S) = \ln\left(1 + \sum_{X\in\{H,M,L\}} w_X . \sum_{v_i \in HV_X(S)} DV(v_i)\right) \tag{2}$$

In order to evaluate the aggregated HVM of a system, we take all the services exposed to the network, and combine their HVMs. If the set of such services in a system A is $SERVICES(A)$, then the aggregated measure, $AHVM(A)$ is calculated as

$$AHVM(A) = \ln\left(\sum_{s_i \in SERVICES(A)} e^{HVM(s_i)}\right) \tag{3}$$

This equation is designed to be dominated by the highest HVM of the services exposed by the policy. We take the exponential average of all the HVMs so that the score will be at least equal to the highest HVM, and will increase with the HVM's of the other services. If arithmetic average was taken, then the risk of the most vulnerable services would have been undermined. Our formalization using the exponential average is validated through our conducted experiments.

B. Probabilistic Vulnerability Measure

The *Probabilistic Vulnerability Measure (PVM)* combines the probability of a vulnerability being discovered in the next period of time and the expected severity of that vulnerability to give an indication of the risk faced by the network in the near future.

Using the vulnerability history of a service, we can calculate the probability of at least one new vulnerability being published in a given period of time as well as the probability distribution of the severities of the vulnerabilities affecting the service. From this

probability distribution, we can compute the expected severity of the vulnerabilities exposed in the next period of time.

We define a new measure, *Expected Risk (ER)* for a service as the product of the probability of at least one new vulnerability affecting the service in the next period of time and the expected severity of the vulnerabilities. Given a score of X for the ER of a service, we can then say that the service has a 100% probability of getting at least one new vulnerability of severity X in the next period of time.

First, we construct the list of the interarrival times between the previous vulnerabilities affecting each service. Then we compute the probability of the interarrival time being less than or equal to a given period of time, T. We find P_{s_i}, the probability that d_{s_i}, the number of days before the next vulnerability of the service s_i is exposed, is less than or equal to a given time interval, T. The value of T that results in the best accuracy is determined experimentally from the vulnerability database used. To compute the expected severity, we build the probability distribution of the severities. Using X as the random variable corresponding to the severities, we can define the expected risk, ER, of a service s_i as in Eqn. 4.

$$ER(s_i) = P_{s_i} \times E[X_{s_i}] \tag{4}$$

Where $E[X_{s_i}]$ is the expected value of X for service s_i. We compute PVM for all services S in the network as in Eqn. 5

$$PVM\,(S) = \ln \sum_{s_i \in S} e^{ER(s_i)} \tag{5}$$

Exponential Distribution is the first method that we have analyzed to calculate PVM. We can fit the interarrival times to an exponential distribution, and then we can find the required probability from the Cumulative Distribution Function (CDF). If λ is the mean interarrival time of service S_i, then the interarrival times of service S_i, d_{S_i}, will be distributed exponentially with the following CDF:

$$P_{s_i} = P\,(d_{S_i} \leq T) = F_{d_{S_i}}\,(T) = 1 - e^{-\lambda T} \tag{6}$$

The next method, that we have analyzed is *Empirical Distribution*. Here, the frequency distribution is used to construct a Cumulative Distribution Function (CDF). Let $f_i\,(x)$ be the number of times the value x occurs in the interarrival time data of the service S_i. Then, the empirical CDF of the interarrival time, d_{S_i}, will be:

$$P_{s_i} = P\,(d_{S_i} \leq T) = F_{d_{S_i}}\,(T) = \frac{\sum_{x \leq T} f_i\,(x)}{\sum f_i\,(x)} \tag{7}$$

IV. Network Policy Risk Analysis

The network policies determine the extent to which a network will be exposed to outside world. The degree to which a policy allows an attack to spread within the network is given by the *Attack propagation (AP)* metric. This measure assesses how difficult it is for an attacker to propagate an attack through the network, using service vulnerabilities as well as security policy vulnerabilities. This measure complements the other three measures introduced earlier, to form a comprehensive metric that can help analyze specific areas to improve the security of a given network where different security policies may interact.

For the purpose of this measure, we define a general network having N hosts and protected by k firewalls in terms of the following sets:
$D = \{d \in N : d\ is\ a\ host\ running\ a\ service\ that\ is\ directly\ reachable\ from\ outside\ the\ network\}$
$S_n = \{s \in S :\ host\ n\ runs\ service\ s\}$
$P = \{p_s : p_s\ is\ the\ combined\ vulnerability\ measure\ for\ service\ s \in S\}$
Here, p_s will be a decreasing function of both EVM and HVM of service s and has the range $(0, 1)$.

1) The Attack Immunity of a Service: For our analysis, we define a measure, I_s, that assesses the attack immunity of a given service, s, to vulnerabilities based on that service's EVM and HVM. I_s is directly calculated from the combined vulnerability measure of a service s, p_s as:

$$I_s = -\ln(p_s) \tag{8}$$

I_s has a range of $[0, \infty)$, and is used to measure the ease with which an attacker can propagate an attack from one host to the other using service s. Thus, if host a can connect to host b using service s exclusively, then I_s measures the immunity of host b to an attack initiating from host a. Assuming that the combined vulnerability measure is independent for different services, we can calculate a combined vulnerability measure $p_{s_{mn}}$ and define the combined attack immunity $I_{s_{mn}}$ as:

$$I_{s_{mn}} = -\ln(p_{s_{mn}}) \times PL \tag{9}$$

Here PL is the protection level. For protection using firewall, this level is 1, and for protection using IDS, this is a value between 0 and 1 that is equal to the false negative rate of the IDS.

After measuring the immunities, we map the network of interest to a Service Connectivity Graph (SCG). The SCG is a directed graph that represents each host in the network by a vertex, and a directed edge from m to n for each pair $m, n \in N$ indicates connectivity from m to n through a service.

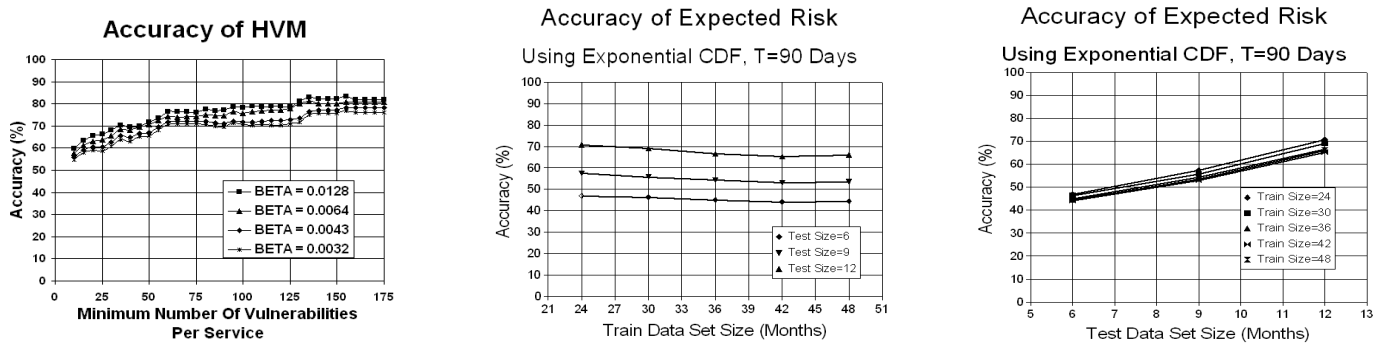
2) Calculating the Attack Propagation: To assess the security of a network, for each node d in D, we want to find how difficult it is for an attacker to compromise all the hosts within reach from d. To do that, we build a minimum spanning tree for d for its segment of the SCG. We define Service Breach Effect (SBE_d) to be the weight of the tree rooted at d in D. It actually denotes the damage possible through d. SBE_d is calculated as in Eqn. 10

$$SBE_d = \sum_{n \in N} \Big(\prod_{m \in nodes\ \ from\ d\ to\ m} p_{s_{dm}} \Big) \times Cost_n \tag{10}$$

Here $Cost_n$ indicates the cost of damage when host n is compromised and N is the set of hosts present in the spanning tree rooted at host d. Finally, the attack propagation metric of the network is:

$$AP(D) = \sum_{d \in D} P(d) \times SBE_d \tag{11}$$

Where, $P(d)$ denotes the probability of the existence of a vulnerability in host d. The equation is formulated such that it provides us with the expected cost as a result of attack propagation within the network.



(a) (b) (c)

Fig. 2. (a) Accuracy of the HVM for different values of β and minimum number of vulnerabilities. (b) Results of ER validation with training data set size vs. accuracy for different test data set sizes. (c) Results of ER validation with test data set size Vs. accuracy for different training data set sizes.

V. Experiments and Results

In our experiments, we used the *National Vulnerability Database (NVD)* published by *National Institute of Science and Technology (NIST)* [10]. The NVD provides a rich array of information that makes it the vulnerability database of choice. All the vulnerabilities are stored using the standard *CVE (Common Vulnerabilities and Exposures)* name. For each vulnerability, the NVD provides the products and versions affected, descriptions, impacts, cross-references, solutions, loss types, vulnerability types, the severity class and score, etc. We have used the database snapshot updated at 04/05/2007. For each vulnerability in the database, NVD provides $CVSS$ [14] scores in the range 1 to 10.

A. Validation of HVM

We conducted an experiment to evaluate the HVM score according to Eqn. 2 with the hypothesis that if service A has a higher HVM than service B, then in the next period of time, service A will display a higher number of vulnerabilities than B.

In our experiment, we used vulnerability data up to 06/30/2006 to compute the HVM of the services, and used the rest of the data to validate the result. We varied β so that the decay function falls to 0.1 in 0.5, 1, 1.5 and 2 years respectively, and observed the best accuracy in the first case. Here, we first chose services with at least 10 vulnerabilities in their lifetimes, and gradually increased this lower limit to 100 and observed that the accuracy increases with the lower limit. As expected of a historical measure, better results have been found when more history is available for the services and observed the maximum accuracy of 83.33%. The graph in Fig. 2(a) presents the results of this experiment.

B. Validation of Expected Risk (ER)

The experiment for the validation of *Expected Risk, ER*, is divided into a number of parts. First, we conducted experiments to evaluate the different ways of calculating the probability. Our general approach was to partition the data into training and test data sets, compute the quantities of interest from the training data sets and validate the computed quantity using the test data sets. If these computed values were close enough, we regarded the measurement as accurate. Here, we obtained the most accurate and stable results using exponential CDF.

The data used in the experiment for exponential CDF was the inter-arrival times for the vulnerability exposures for the services in the database. We varied the length of the training data set and test data set for both the probability and Expected Severity. In this case, we considered only the data with $T = 90$.

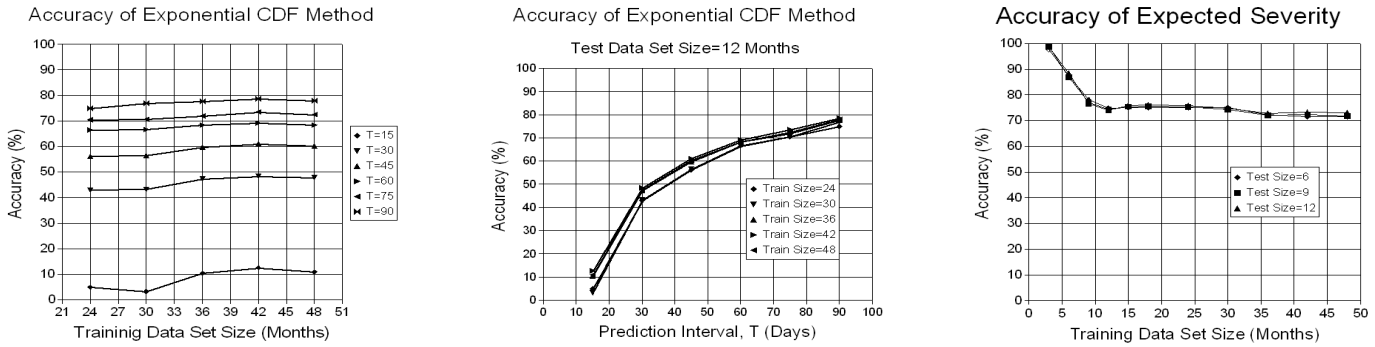
In the experiment for exponential CDF, we constructed an exponential distribution for the interarrival time data using Eqn. 6. For each training set, we varied the value of T from 15 days to 90 days in 15 day increments, and ran validation for each value of T with the test data set. Finally we took the average of the accuracies for each combination of training data set size and test data set size.

a) Results: The results of the ER experiment are presented in Figs. 2(b) and 2(c). In Fig. 2(b), we present the accuracy along the Y axis as a percentage against the training data set sizes. In Fig. 2(c) we present the same accuracies but with respect to test set sizes. Fig. 3(a) presents the accuracy of the computed probabilities using Exponential CDF method. For the test data set size of 12 months, we observed the highest accuracy of 78.62% with the 95% confidence interval being $[72.37, 84.87]$ for the training data set size of 42 months and prediction interval of 90 days. We present the results of the experiment for Expected Severity in Fig. 3(c). The maximum accuracy obtained in this experiment was 98.94% for the training data set size of 3 months and test data set size of 9 months.

b) Discussion Of Results: From the graph in Fig. 3(a) and Fig. 3(b), it is apparent that the accuracy of the Exponential CDF model is not much sensitive to training data set sizes but increases with T. This implies that this method is not sensitive to the volume of training data available to it. From Fig. 3(c), it is readily observable that the accuracy of the *Expected Severity* is not dependent on the test data set size. It increases quite sharply with decreasing values of training data set size for small values. This means that the expectated values calculated from the most recent data is actually the best model for the expected severity in the test data. And from Fig. 2(b) and Fig. 2(c), we can observe that the accuracy of the *Expected Risk (ER)* is slightly correlated with the training data set size but strongly correlated with the test data set size.

C. Running Time Evaluation Of Attack Propagation Metric

To assess the feasibility of calculating the AP under different conditions, we ran a MATLAB implementation of the algorithm for different network sizes, as well as different levels of network connectivity percentages (the average percentage of the network



(a) (b) (c)

Fig. 3. (a) Training Data Set Size Vs. Accuracy graph for the Exponential CDF method for probability calculation, test set size = 12 months. (b) Prediction Interval Parameter (T) Vs. Accuracy graph for the Exponential CDF method for probability calculation, test set size = 12 months. (c) Training data set size Vs. Accuracy graph for the Expected Severity calculation for different values of the test data set size.

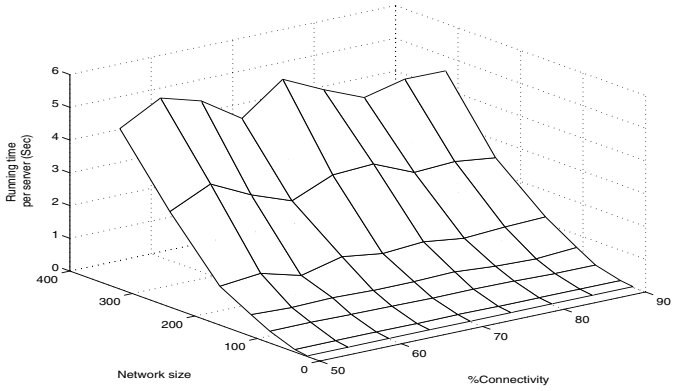


Fig. 4. The average running times for different network sizes and %connectivity

directly reachable from a host). The machine used to run the simulation was a 1.9GHz P-IV processor with 768MB RAM. The results are shown in figure 4. For each network size, we generated several random networks with the same %connectivity value. The running time was then calculated for several hosts within each network. As can be seen from the figure, the quadratic growth of the running time against network connectivity was not noticeably dependent on the network %connectivity in our implementation.

VI. Conclusions

A unified policy evaluation metric will be highly effective in assessing the protection of the current policy, and justifying consequent decisions to strengthen security of a network. In this paper, we present a novel approach to quantitatively evaluate network security by identifying, formulating and validating several important factors that greatly affect the security of a network. Our experiments validate our hypothesis that if a service has a highly vulnerability prone history, then there is higher probability that the service will become vulnerable again in the near future. These metrics are useful not only for administrators to evaluate policy/network changes and, take timely and judicious decisions, but also for enabling adaptive security systems based on vulnerability and network changes.

Our experiments provide very promising results regarding our metric. Our vulnerability prediction model proved to be up to 78% accurate, while the accuracy level of our historical vulnerability measurement was 83.33% based on real-life data from *National Vulnerability Database (NVD)*. The accuracies obtained in these experiments, vindicate our claims about the components of our risk measurement framework and proves its effectiveness.

Acknowledgments

The authors would like to thank Mohamed Mahmoud Taibah of Depaul University and Muhammad Abedin and Syeda Nessa of The University of Texas at Dallas for their help with the formalization and experiment which made this work possible.

References

[1] M. Abedin, S. Nessa, E. Al-Shaer, and L. Khan. Vulnerability analysis for evaluating quality of protection of security policies. In *2nd ACM CCS Workshop on Quality of Protection*, Alexandria, Virginia, October 2006.
[2] E. Al-Shaer and H. Hamed. Discovery of policy anomalies in distributed firewalls. In *Proceedings of IEEE INFOCOM'04*, March 2004.
[3] E. Al-Shaer, H. Hamed, R. Boutaba, and M. Hasan. Conflict classification and analysis of distributed firewall policies. *IEEE Journal on Selected Areas in Communications (JSAC)*, 23(10), October 2005.
[4] O. H. Alhazmi and Y. K. Malaiya. Prediction capabilities of vulnerability discovery models. In *Proc. Reliability and Maintainability Symposium*, pages 86–91, January 2006.
[5] P. Ammann, D. Wijesekera, and S. Kaushik. Scalable, graph-based network vulnerability analysis. In *9th ACM conference on Computer and communications security*, pages 217–224, 2002.
[6] A. Atzeni and A. Lioy. Why to adopt a security metric? a little survey. In *QoP-2005: Quality of Protection workshop*, September 2005.
[7] H. Hamed, E. Al-Shaer, and W. Marrero. Modeling and verification of ipsec and vpn security policies. In *IEEE ICNP'2005*, November 2005.
[8] S. Kamara, S. Fahmy, E. Schultz, F. Kerschbaum, and M. Frantzen. Analysis of vulnerabilities in internet firewalls. *Computers and Security*, 22(3):214232, April 2003.
[9] P. Manadhata and J. Wing. An attack surface metric. In *First Workshop on Security Metrics*, Vancouver, BC, August 2006.
[10] National institute of science and technology (nist). http://nvd.nist.gov.
[11] J. Pamula, P. Ammann, S. Jajodia, and V. Swarup. A weakest-adversary security metric for network configuration security analysis. In *ACM 2nd Workshop on Quality of Protection 2006*, Alexandria, VA, October 2006.
[12] R. Rogers, E. Fuller, G. Miles, M. Hoagberg, T. Schack, T. Dykstra, and B. Cunningham. *Network Security Evaluation Using the NSA IEM*. Syngress Publishing, Inc., first edition, August 2005.
[13] M. Sahinoglu. Security meter: A practical decision-tree model to quantify risk. In *IEEE Security and Privacy*, June 2005.
[14] M. Schiffman. A complete guide to the common vulnerability scoring system (cvss). http://www.first.org/cvss/cvss-guide.html, June 2005.