Toward Measuring Network Security Using Attack Graphs

Lingyu Wang
Concordia Institute for Information Systems Engineering
Concordia University
Montreal, QC H3G 1M8, Canada
wang@ciise.concordia.ca

Anoop Singhal
Computer Security Division
National Institute of Standards and Technology
Gaithersburg, MD 20899, USA
anoop.singhal@nist.gov

Sushil Jajodia
Center for Secure Information Systems
George Mason University
Fairfax, VA 22030-4444, USA
jajodia@gmu.edu

ABSTRACT

In measuring the overall security of a network, a crucial issue is to correctly compose the measure of individual components. Incorrect compositions may lead to misleading results. For example, a network with less vulnerabilities or a more diversified configuration is not necessarily more secure. To obtain correct compositions of individual measures, we need to first understand the interplay between network components. For example, how vulnerabilities can be combined by attackers in advancing an intrusion. Such an understanding becomes possible with recent advances in modeling network security using *attack graphs*. Based on our experiences with attack graph analysis, we propose an integrated framework for measuring various aspects of network security. We first outline our principles and methodologies. We then describe concrete examples to build intuitions. Finally, we present our formal framework. It is our belief that metrics developed based on the proposed framework will lead to novel quantitative approaches to vulnerability analysis, network hardening, and attack response.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General–Security and protection

General Terms

Security

Keywords

intrusion detection, vulnerability analysis, security metrics, attack graph

1. INTRODUCTION

Crucial to today's economy and national security, computer networks play a central role in most enterprises and critical infrastructures including power grids, financial data systems, and emergency communication systems. In protecting these networks against malicious intrusions, a standard way for measuring network security will bring together users, vendors, and labs in specifying, implementing, and evaluating network security products. Despite existing efforts in standardizing security metrics [16, 27], a widely-accepted network security metrics is largely unavailable. At the research frontier, a qualitative and imprecise view toward the evaluation of network security is still dominant. Researchers are concerned about issues with binary answers, such as whether a given critical resource is secure (vulnerability analysis) or whether an insecure network can be hardened (network hardening). The solutions to such issues only have a limited value due to their qualitative nature. For example, in practice it is usually more desired to know how secure a resource is and how much a network can be hardened.

A critical issue in measuring network security is to compose measures of individual vulnerabilities, resources, and configurations into a global measure. A naive approach to such compositions may lead to misleading results. For example, less vulnerabilities are not necessarily more secure, considering a case where these vulnerabilities must all be exploited in order to compromise a critical resource. On the other hand, less vulnerabilities can indeed mean more security when exploiting any of these vulnerabilities is sufficient for compromising that resource. This example shows that to obtain correct compositions of individual measures, we need to first understand the interplay between different network components. For example, how can an attacker combine different vulnerabilities to advance an intrusion; how can exploiting one vulnerability reduce the difficulty of exploiting another vulnerability; how can compromising one resource affect the damage or risk of compromising another resource; how can modifying one network parameter affect the cost of modifying another parameter.

The study of composing individual measures of network security becomes feasible now due to recent advances in modeling network security with *attack graphs* (a review of related work will be given in the next section). Attack graphs provide the missing information about relationships among network components and thus allow us to consider potential attacks and their consequences in a particular *context*. Such a context makes it possible to compose individual measures of vulnerabilities, resources, and configurations into a global measure of network security. The current research is based on our past experiences with attack graphs [15, 14, 12, 31, 29, 30, 33, 21] and the Topological Vulnerability Analysis (TVA) system, which can generate attack graphs for more than 37,000 vulnerabilities taken from 24 information sources including X-Force, Bugtraq, CVE, CERT, Nessus, and Snort [12]. The presence of such a powerful tool demonstrates the practicality of using attack graphs as the basis for measuring network security.

QoP'07, October 29, 2007, Alexandria, Virginia, USA.

In this paper, we outline an integrated framework for measuring various aspects of network security based on attack graphs. We first state our principles and methodologies based on lessons learned from different but related areas. We then describe examples to build intuitions about general requirements on network security metrics. Finally, we present our formal framework. It is our belief that the proposed framework will spawn more interests and efforts in measuring the relative damage (instead of a binary answer of secure or insecure), the reconfiguration cost (instead of the mere feasibility of a reconfiguration), the amount of resistance to attacks (instead of the possibility of an attack), and other quantitative security properties. Many issues in vulnerability analysis will need to be revisited due to the presence of a quantitative objective. Network hardening and attack response will be guided by the pursue of an optimal solution in terms of the available metrics, rather than stopping at an arbitrary solution. Other areas of network security research may also benefit. For example, the design of a honeypot or a honeynet may want to minimize its resistance to certain attacks so as to make the network more attractive to attackers. In summary, we expect the proposed framework to lead to a series of studies in network security metrics and to have a fundamental impact on future study of vulnerability analysis, network hardening, and attack response.

The rest of the paper is organized as follows. Section 2 reviews related work on network metrics and attack graphs. Section 3 outlines our principles and methodologies. Section 4 first describes examples to build intuitions about general requirements on network security metrics, and it then presents a formal framework for composing network metrics from individual measures based on attack graphs. Finally, Section 5 concludes the paper.

2. RELATED WORK

The 2001 Workshop on Information Security System Scoring and Ranking presents an overview of various issues relevant to security metrics [2]. The NIST's efforts on standardizing security metrics are reflected in the Technology Assessment: Methods for Measuring the Level of Computer Security [16] and more recently in the Security Metrics Guide for Information Technology Systems [27]. The latter describe the current state of practice of security metrics, such as that required by the Federal Information Security Management Act (FISMA). Another overview of many aspects of network security metrics is given in [10]. Dacier et. al give intuitive properties based on common senses that should be satisfied by any security metric [7, 8, 17]. Based on an exponential distribution for an attacker's success rate over time, they use a Markov model and the MTTF (Mean Time to Failure) to measure security. They discuss simple cases of combining individual measures but do not study the general case. We borrow some of their intuitive properties but we give a more general way for combining individual measures into overall attack resistance measures.

Our approach of using additional functions for modeling the dependency between individual measures is inspired by the work by Balzarotti et. al [3]. However, their work focus on computing the minimum efforts required for executing each exploit, whereas we compute overall security with respect to given critical resources. Another difference lies in the kind of dependency modeled using additional functions. We model two kind of dependencies, the one captured by attack graphs and the other captured by additional functions. The latter affects the measure of network components but does not enables it to be reachable. Based the exploitability concept, a qualitative measure of risk is given in [4]. Another approach measures the relative risk of different configurations using the *weakest attacker* model, that is the least conditions enabling an attack [21]. Yet another series of work measures how likely a software is vulnerable to attacks using a metrics called *attack surface* [11, 18, 19, 20, 13]. These work allow a partial order to be established on different network configurations based on their relative security. However, the treatment of many aspects of security is still qualitative in nature. For example, the resources are still treated equally important (no explicit evaluation of damages) and the resistance to attacks is regarded as binary (an attack is either impossible or trivial). Our goal is to quantify all such perspectives.

Relevant work exist in other areas, such as the study of trust in distributed systems. Beth et. al proposed a metrics for measuring the trust in an identity that has been established through overlapping chains of certificates [5]. The way they combine values of trust in certificates into an overall value of trust proves to be useful in our study. Similarly, the design principles given by Reiter et. al are intended for developing metrics of trust, but we found these principles applicable to our study [24]. The formal logic language introduced for measuring risks in trust delegation in the RT framework provides us hints for developing the formal semantics of our metrics [6]. Our work on minimum-cost network hardening is one of the first efforts toward the quantitative study of network security [15, 31]. This work quantifies the cost of removing vulnerabilities in hardening a network, but it does not consider other hardening options, such as modifying the connectivity. It also has the limitation of adopting a qualitative view of damages (all the given critical resources are equally important) and of attack resistance (attacks on critical resources are either impossible or trivial). More recently, we propose an attack resistance metric based on attack graph in [32]. In this paper, we generalize that work into a framework for devising metrics for other aspects of network security, such as damages, risks, and configuration costs.

To generate attack graphs, topological vulnerability analysis enumerates potential multi-step intrusions based on prior knowledge about vulnerabilities and their relationships [7, 9, 17, 22, 34, 28]. Based on whether a search starts from the initial state or the final state, such analyses can be forward [22, 28] or backward [25, 26]. Model checking is first used to analyze whether a given goal state is reachable from the initial state [23, 25] and later is modified to enumerate all possible sequences of attacks between the two states [26]. To avoid the exponential explosion in the number of such explicit attack sequences, a more compact representation of attack graphs was proposed based on the *monotonicity assumption* saying an attacker never needs to relinquish any obtained capability [1]. On the attack response front, attack graphs have been used for the correlation of attacks, the hypotheses of alerts missed by IDSs, and the prediction of possible future attacks [29, 30].

3. PRINCIPLES AND METHODOLOGIES

We borrow design principles proposed for developing metrics of trust in distributed systems [24]. The original principles apply to situations where the amount of trust needs to be evaluated for an identity established via overlapping chains of certificates. The difference in players (identities and certificates versus machines and vulnerabilities) and purposes (to measure the amount of trust versus to measure the damage, attack resistance, and cost) prevents a direct application of original principles. We first list the adapted principles and then interpret them one by one.

1. Value assignment should be based on specific, unambiguous interpretations rather than abstract and meaningless rules.
2. The metrics should take into consideration all the information that may be relevant to its potential applications.
3. The metrics should at the same time leave to users the decisions that cannot be automated with unambiguous rules.

4. Measuring hosts as collections of vulnerabilities, instead of as the combination of hardware/software configurations.
5. The outcome of the metrics should enable its application to make an immediate decision.

The first principle indicates that efforts should be made to assign individual measure values on the basis of concrete and intuitive facts, which can be measured in practice without ambiguity. For example, the damage is ideally measured in dollars or in the time for recovery; the difficulty of an attack may be measured by the average time or computational complexity required for completing the attack. On the other hand, measuring these with an abstract and meaningless probability value may render the metrics less repeatable and hence lose its practical value. The second and third principles jointly require a clear definition of the scope of the metrics. That is, a metrics should not measure what involves human decisions, whereas it should not leave out anything else. For example, the minimum-cost hardening approach described in [15, 31] does not adhere to the third principle because it only measures one of the reconfiguration possibilities (that is, removing vulnerabilities) while leaving out other important factors, such as the cost of changing the connectivity, the attack resistance of a configuration, and the relative importance of each resource.

The fourth principle limits the scope of a metrics from another perspective, that is a metrics should be based on the correctness of a underlying qualitative understanding of the network, rather than to replace the latter. For example, if a metrics measures a host as a specific combination of hardware/software configurations, then the metrics is making an assumption about the association between vulnerabilities and configurations. Such an association is established by a qualitative vulnerability analysis, and it should remain so; expanding the scope of metrics to cover the job of such an analysis will only introduce potential ambiguity. The fifth principle says that the outcome of a measure should not be far from the requirement of an application (or a human user) in making its decision. Using exploitability or the minimal set of conditions required for compromising a resource [21] falls short on this principle. Although in some situations such a metrics may tell us whether a configuration is better than the other (if the former requires a proper subset of conditions of what required by the latter), in other cases the two configurations will have incomparable results according to the metrics and thus a decision cannot be made.

We develop methodologies based on the above principles. First of all, according to the fourth principle, we take as inputs the attack graph of a network, instead of the network configuration information. This approach relieves security metrics from unnecessary burdens of ensuring the accuracy in vulnerability information and in the relationships among vulnerabilities. Second, we divide the development of a security metrics into two phases, that is the measures of individual security components and the composition.

1. According to the first principle, the focus of the first stage is to devise methods for specific applications to assign values based on a concrete and meaningful ground. Case studies can help to establish typical requirements in applications and to define metrics as different weighted combinations of such requirements. For example, more weight will be given to recovery time for damages in availability-centric applications (for example, ISPs), whereas more weight is given to dollar amounts for damages in electronic commerce applications. As another example, the difficulty of exploiting a vulnerability can be measured on a trivial or non-trivial basis in a coarse-grained application, or using the order of average (or theoretical) running time in finer-grained applications. Finally, the cost in removing a vulnerability or changing the connectivity both involve administrative costs, and may be measured as a weighted amount in time and dollars.

   The second and the third principles require us to include all and only the measurable factors in the defined security metrics. Currently, we consider three aspects, that is the *significance* of a resource in terms of damages caused by compromising the resource, the *cost* of reconfiguration, and the *resistance* of a vulnerability. More aspects may be added to our framework as the research evolves. Nevertheless, the current collection of the three aspects already yields several advantages over many existing approaches:

   - Measuring the potential damage caused by each attack allows us to explicitly identify the relative significance of different resources. In most existing work (including ours [15, 31]), a common assumption is that each resource either needs to be protected or can be left unattended, and all the protected resources are of equal importance. This is apparently an over-simplified assumption that does not always hold in practice. Our approach makes no such assumptions, and instead the relative significance of each resource is more precisely measured by the amount of damage it may cause.
   - Measuring the cost of reconfiguring a network allows us to judge the relative overhead in adopting each hardening solution. This reflects our consideration of the last principle, that is to present users or applications with the most direct solution. We consider not just the cost in removing vulnerabilities [15, 31] but also costs associated with other possible hardening operations, such as modifying connectivity (which may remove existing vulnerabilities while at the same time introducing new vulnerabilities). We also consider both the off-line cost (in hardening a network) and the on-line cost (in real-time attack response). By measuring the costs, we (partially) order all solutions so an application or user can easily pick the most desirable one.
   - Attack resistance [32] removes the limitation of existing qualitative methods in assuming each attack to be either impossible or trivial. Under such an assumption, if no network hardening solution exists to make all potential attacks impossible, then the network cannot be hardened. However, analogous to the fact that we still need IDSs even though we already have vulnerability scanners, a less-than-ideal situation is usually the case where a hardening solution cannot eliminate all potential attacks. Nevertheless, a solution is more desirable if it leaves out attacks that are more difficult to achieve.

2. The second stage in developing a metrics focuses on the composition of individual measures. In contrast to the concrete approach we adopt for the first stage, we assume abstract domains for each factor of the metrics. This abstract approach ensures the extensibility of the proposed framework. The framework only depends on assumptions about individual measures. This allows us to compose unforeseen measure types. For example, the two abstract composition operators used to compose individual attack resistances can easily accommodate different domains of attack resistance, such as non-negative integers and sets of security conditions [32]. Next section describes the framework in more details.

4. A GENERIC FRAMEWORK

In this section, we first build intuitions through examples and then present a framework for composing individual measures.

4.1 Examples

We shall use the notion of attack graph for a directed graph with two types of vertices that correspond to exploits, and the pre- and post-conditions of exploits, respectively. Directed edges point from each pre-condition to an exploit and from the exploit to each post-condition. In Figure 1 and Figure 2, exploits are represented as predicates of the form $vulnerability(source, target)$ inside ovals, and conditions as predicates of the form $condition(host)$ (conditions involving a single host) or $condition(source, target)$ (conditions involving a pair of hosts) in plaintext. The left-hand side attack graph in Figure 1 depicts a well-known attack scenario where an attacker may establish trust relationship between host 0 and host 2 by misusing the .rhost file and then obtain user privilege and root privilege on host 2 via other two vulnerabilities. The right-hand side of Figure 1 depicts another more complicated scenario where a firewall blocks attack attempts from the external host 0 to the internal host 2, but the attacker can get around the restriction by using another internal host 1 as a stepping stone. Figure 2 is simply the composition of the previous two scenario, that is the attacker can attack host 2 via either host 0 or host 1.

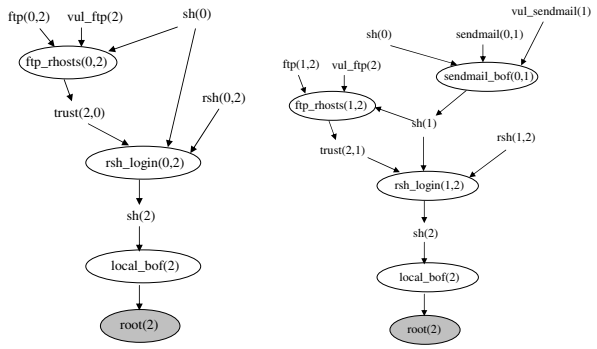

Figure 1: Two Network Configurations

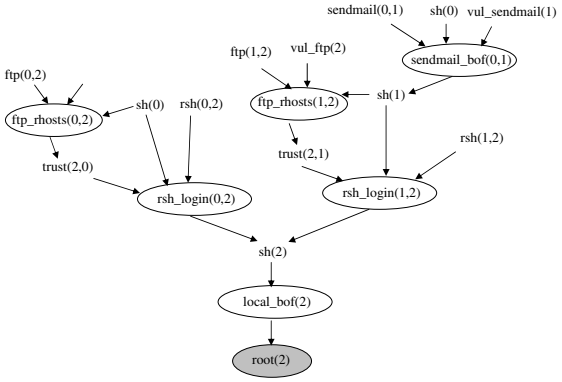

Figure 2: Another Network Configuration

The dependency relationship between exploits shown in these figures clearly indicates that separately measuring each exploit or condition would be insufficient for understanding the overall security of the network. For example, suppose the reconfiguration cost metrics is simply the number of services to be disabled. Even in these simple cases, it may not be straightforward to find a solution that minimizes the reconfiguration cost and at the same time prevents the attacker from reaching the goal. On the other hand, by exploring the dependency relationships among vulnerabilities, the minimum-cost network hardening approach in [15, 31] can compute optimal solutions based on the given attack graphs. As another example, suppose we know exactly how difficult it is to exploit each vulnerability in the attack graphs. Again, it may not be straightforward to see how difficult reaching the attack goal is, or equivalently, which of the attack graphs provides the most resistance to attacks.

We now show through examples that incorrect compositions of individual measures can lead to misleading results. These examples also help to build intuitions about requirements that a security metrics should meet. First, a common myth is that *less vulnerabilities mean better security*. As a counterexample, the network in the right-hand side of Figure 1 apparently has more vulnerabilities than the left-hand side does. Nonetheless, the fact that the right-hand side has a firewall blocking traffic from external hosts implies better security in the sense that an attacker needs to make more efforts to reach his/her goal (by first obtaining user privilege on host 1 through the sendmail buffer overflow vulnerability and then using host 1 as a stepping stone to attack host 2). This example shows that more vulnerabilities may actually mean better security, if these vulnerabilities are all needed for reaching the goal.

Another common myth is that *the security of a network is equal to the least effort required for reaching the goal*. As a counterexample, suppose we apply this rule to Figure 2. There are two possible ways for reaching the attack goal, which correspond to the left-hand side and the right-hand side of Figure 1, respectively. From above discussions, we know that the second option requires more effort than the first option. Hence, one may be inclined to measure the overall security in Figure 2 as the same as that in the left-hand side of Figure 1. However, the fact that an attacker now has two different choices in reaching the goal (that is, either to attack host 2 directly from host 0, or using host 1 as a stepping stone) clearly indicates less security in Figure 2 [1]. This example also shows that more vulnerabilities may mean less security since the network in Figure 2 has more vulnerabilities than both networks in Figure 2. That is, the number of vulnerabilities and the least effort required for reaching the attack goal are both insufficient for characterizing the security of a network.

A third myth is that *more diversity in configurations means better security*. As a counterexample, consider another network configuration obtained by replacing the vulnerability $ftp\_rhost(1,2)$ in Figure 2 with a different type of vulnerability that has the same pre- and post-conditions and the same difficulty. This new configuration thus has more diversity. However, this configuration is actually less secure because now an attacker has more choices in advancing the intrusion. That is, in Figure 2 the attacker must succeed in exploiting the vulnerability ftp_rhost at least once, whereas in the new configuration he/she can choose to exploit two different vulnerabilities. On the other hand, if we replace the exploit $sendmail\_bof(0,1)$ with $ftp\_rhost(0,1)$ (notice that the pre- and post-conditions would not match, which we shall ignore) then the new configuration has less diversity. This new configuration is indeed less secure, considering that the attacker may accumulate experiences after exploiting the vulnerability ftp_rhost for the first time. These examples show that diversity in configuration may either increase or decrease the overall security.

[1]Notice that there may be other reasons for using host 1 as a stepping stone, such as to avoid detection.

4.2 The Framework

Above examples also reveal general requirements that should be satisfied by a security metric in composing individual measures. First, the difference between the two attack graphs in Figure 1 shows that *a longer path leading to the attack goal means better security*. Here the path means a sequence of interdependent vulnerabilities leading to an attack goal. Notice that this is only an intuitive statement, and a longer path not only refers to the number of vulnerabilities along the path. Second, the key observation in Figure 2 is that there are two possible attack paths reaching the attack goal. Although one of the paths is more difficult than the other, it nonetheless provides attackers more chances in successfully advancing an intrusion. Hence, *multiple attack paths are less secure than any of the paths alone*. Finally, we have argued that the vulnerability ftp_rhosts may be easier to exploit for the second time than a different vulnerability due to accumulated experiences or tools. However, notice that this fact cannot be modeled as an additional edge in attack graph, because the two vulnerabilities may not be directly adjacent. In another word, *executing an exploit may change the difficulty of executing another exploit, even if the two do not directly depend on each other in the attack graph*.

We have been assuming a single attack goal $root(2)$ in above discussions. However, any condition in an attack graph may mean certain amount of risk and thus should be guarded against potential attacks. For example, having the user privilege $sh(1)$ in Figure 2 may already allow an attacker to cause considerable damage to host 1 even though this risk is not as significant as that of $root(2)$. That is, *the security of a network should be measured against all relevant resources with different weights*. Finally, we mention that the right-hand side of Figure 1 is obtained by introducing a firewall that blocks direct accesses from any external host to host 2. This restriction apparently incurs two type of costs, that is the administrative cost of setting up the firewall and the functional cost in terms of reduced functionality on host 2. That is, *each reconfiguration may incur a certain cost that must be considered together with security*.

Given an initial configuration, we measure the *resource significance*, *reconfiguration cost*, and *attack resistance* of a network configuration based on the composition of individual measures. We assume individual measures are given as a partial order, such as using real numbers (which is a total order). This requirement allows us to estimate an approximate ordering on the domain of resource significance, reconfiguration cost, and attack resistance even when there is not enough information to precisely measure them. For example, we may simply regard the resistance of individual exploits as the set of initial conditions (that is, conditions not implied by other exploits) required by that exploit [32]. Different applications may define these properties in significantly different ways. To make our framework broadly applicable, we define it in a generic form while leaving the individual measures uninterpreted.

Central to the framework are two type of composition operators, denoted as $\oplus$ and $\otimes$. The two operators correspond to the disjunctive and conjunctive dependency relationship between exploits in an attack graph, respectively. Based on the intuitive properties mentioned in previous sections, the two operators should satisfy that $r_1 \oplus r_2$ is no greater than r_1 or r_2, whereas $r_1 \otimes r_2$ is no less than r_1 and r_2, with respect to a given ordering on the domain of attack resistance. In addition to the two composition operators, we use a function $F()$ to map a set of exploits to another exploit and its resistance value. The function is intended to capture the dependency relationship between the resistance value of exploits. That is, executing some exploits may affect the resistance value of another exploit, even though the latter cannot be executed yet. In most cases, this effect will be to assign a lower resistance value to the affected exploit. For example, exploits involving the same type of vulnerability should be related together using this function such that successfully exploiting one instance of the vulnerability reduces the resistance of others due to the attacker's accumulated experiences and tools. We summarize the model in Definition 1.

DEFINITION 1. *Given an attack graph with the set of exploits $\mathcal{E}$ and the set of conditions $\mathcal{C}$, we define*

- *A total function $S() : \mathcal{C} \rightarrow \mathcal{S}$,*
- *a total function $T() : \mathcal{C} \rightarrow \mathcal{T}$,*
- *a total function $r() : \mathcal{E} \rightarrow \mathcal{D}$,*
- *a total function $R() : \mathcal{E} \rightarrow \mathcal{D}$,*
- *an operator $\oplus : \mathcal{D} \times \mathcal{D} \rightarrow \mathcal{D}$,*
- *an operator $\otimes : \mathcal{D} \times \mathcal{D} \rightarrow \mathcal{D}$, and*
- *a function $F() : \mathcal{E} \rightarrow \mathcal{E} \times \mathcal{D}$.*

We call the set $\mathcal{S}$, $\mathcal{T}$, and $\mathcal{D}$ the domain of resource significance, the domain of reconfiguration cost, and the domain of attack resistance. We call $S(c)$ the resource significance *of a condition c and $T(c)$ the* reconfiguration cost *of c. We call $r(e)$ the* individual resistance *of an exploit e and $R(e)$ the* cumulative resistance *of e.*

The main tasks in instantiating the framework to be a security metric for a specific application include populating the individual measures by defining the function $S()$, $T()$, and $r()$, determining suitable operators $\oplus$ and $\otimes$, and capturing additional dependency relationship between individual resistances using the function $F()$. The cumulative resistance function $R()$ can then be computed by composing individual resistances against each resource as weighted by $S()$, while considering the dependency captured in $F()$.

5. CONCLUSION

When optimizing networks for security, a qualitative and imprecise argument can mislead the decision making and as a result cause the reconfigured network to be in fact less secure. We have shown examples of such misleading results, which demonstrated the significance of correctly composing individual measures into a global metrics. We have outlined a general framework for devising network security metrics based on attack graphs. The framework had two stages. The first stage adopted a concrete approach to the assignment of values to individual security components in order to avoid introducing ambiguous or meaningless value assignments into a metrics. The second stage, in contrast, adopted an abstract approach to the composition of individually assigned values. This reflected the effort in accommodating different types of value assignment. Our future work will instantiate the framework into concrete metrics for different applications. We will also revisit the study of vulnerability analysis, network hardening, and attack response based on the proposed metrics.

Acknowledgements.

This material is based upon work supported by National Institute of Standards and Technology Computer Security Division; by Homeland Security Advanced Research Projects Agency under the contract FA8750-05-C-0212 administered by the Air Force Research Laboratory/Rome; by Air Force Research Laboratory/Rome under the contract FA8750-06-C-0246; by Army Research Office under grant W911NF-05-1-0374; by Federal Aviation Administration under the contract DTFAWA-04-P-00278/0001; by National Science Foundation under grants CT-0627493, IIS-0242237, and IIS-0430402; and by Natural Sciences and Engineering Research

Council of Canada under Discovery Grant N01035. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsoring organizations. The authors are grateful to the anonymous reviewers for their valuable comments.

6. REFERENCES

[1] P. Ammann, D. Wijesekera, and S. Kaushik. Scalable, graph-based network vulnerability analysis. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS'02)*, pages 217–224, 2002.

[2] Applied Computer Security Associates. Workshop on. In *Information Security System Scoring and Ranking*, 2001.

[3] D. Balzarotti, M. Monga, and S. Sicari. Assessing the risk of using vulnerable components. In *Proceedings of the 1st Workshop on Quality of Protection*, 2005.

[4] P. Balzarotti, M. Monga, and S. Sicari. Assessing the risk of using vulnerable components. In *Proceedings of the 2nd ACM workshop on Quality of protection*, 2005.

[5] T. Beth, M. Borcherding, and B. Klein. Valuation of trust in open networks. In *Proceedings of the Third European Symposium on Research in Computer Security (ESORICS'94)*, pages 3–18, 1994.

[6] P. Chapin, C. Skalka, and X.S. Wang. Risk assessment in distributed authorization. In *3rd ACM Workshop on Formal Methods in Security Engineering: From Specifications to Code*, 2005.

[7] M. Dacier. Towards quantitative evaluation of computer security. Ph.D. Thesis, Institut National Polytechnique de Toulouse, 1994.

[8] M. Dacier, Y. Deswarte, and M. Kaaniche. Quantitative assessment of operational security: Models and tools. Technical Report 96493, 1996.

[9] D. Farmer and E.H. Spafford. The COPS security checker system. In *USENIX Summer*, pages 165–170, 1990.

[10] K.S. Hoo. Metrics of network security. White Paper, 2004.

[11] M. Howard, J. Pincus, and J. Wing. Measuring relative attack surfaces. In *Workshop on Advanced Developments in Software and Systems Security*, 2003.

[12] S. Jajodia, S. Noel, and B. O'Berry. Topological analysis of network attack vulnerability. In V. Kumar, J. Srivastava, and A. Lazarevic, editors, *Managing Cyber Threats: Issues, Approaches and Challenges*. Kluwer Academic Publisher, 2003.

[13] K. Manadhata, J.M. Wing, M.A. Flynn, and M.A. McQueen. Measuring the attack surfaces of two ftp daemons. In *Quality of Protection Workshop*, 2006.

[14] S. Noel and S. Jajodia. Correlating intrusion events and building attack scenarios through attack graph distance. In *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC'04)*, 2004.

[15] S. Noel, S. Jajodia, B. O'Berry, and M. Jacobs. Efficient minimum-cost network hardening via exploit dependency grpahs. In *Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC'03)*, 2003.

[16] National Institute of Standards and Technology. Technology assessment: Methods for measuring the level of computer security. NIST Special Publication 500-133, 1985.

[17] R. Ortalo, Y. Deswarte, and M. Kaaniche. Experimenting with quantitative evaluation tools for monitoring operational security. *IEEE Trans. Software Eng.*, 25(5):633–650, 1999.

[18] J. Wing P. Manadhata. Measuring a system's attack surface. Technical Report CMU-CS-04-102, 2004.

[19] J. Wing P. Manadhata. An attack surface metric. Technical Report CMU-CS-05-155, 2005.

[20] J. Wing P. Manadhata. An attack surface metric. In *First Workshop on Security Metrics (MetriCon)*, 2006.

[21] J. Pamula, S. Jajodia, P. Ammann, and V. Swarup. A weakest-adversary security metric for network configuration security analysis. In *Proceedings of the 2nd ACM workshop on Quality of protection*, pages 31–38, New York, NY, USA, 2006. ACM Press.

[22] C. Phillips and L. Swiler. A graph-based system for network-vulnerability analysis. In *Proceedings of the New Security Paradigms Workshop (NSPW'98)*, 1998.

[23] C.R. Ramakrishnan and R. Sekar. Model-based analysis of configuration vulnerabilities. *Journal of Computer Security*, 10(1/2):189–209, 2002.

[24] M.K. Reiter and S.G. Stubblebine. Authentication metric analysis and design. *ACM Transactions on Information and System Security*, 2(2):138–158, 5 1999.

[25] R. Ritchey and P. Ammann. Using model checking to analyze network vulnerabilities. In *Proceedings of the 2000 IEEE Symposium on Research on Security and Privacy (S&P'00)*, pages 156–165, 2000.

[26] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J.M. Wing. Automated generation and analysis of attack graphs. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy (S&P'02)*, pages 273–284, 2002.

[27] M. Swanson, N. Bartol, J. Sabato, J. Hash, and L. Graffo. Security metrics guide for information technology systems. NIST Special Publication 800-55, 2003.

[28] L. Swiler, C. Phillips, D. Ellis, and S. Chakerian. Computer attack graph generation tool. In *Proceedings of the DARPA Information Survivability Conference & Exposition II (DISCEX'01)*, 2001.

[29] L. Wang, A. Liu, and S. Jajodia. An efficient and unified approach to correlating, hypothesizing, and predicting intrusion alerts. In *Proceedings of the 10th European Symposium on Research in Computer Security (ESORICS 2005)*, pages 247–266, 2005.

[30] L. Wang, A. Liu, and S. Jajodia. Using attack graphs for correlating, hypothesizing, and predicting intrusion alerts. *Computer Communications*, 29(15):2917–2933, 2006.

[31] L. Wang, S. Noel, and S. Jajodia. Minimum-cost network hardening using attack graphs. *Computer Communications*, 29(18):3812–3824, 11 2006.

[32] L. Wang, A. Singhal, and S. Jajodia. Measuring the overall security of network configurations using attack graphs. In *Proceedings of 21th IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec 2007)*, 2007.

[33] L. Wang, C. Yao, A. Singhal, and S. Jajodia. Interactive analysis of attack graphs using relational queries. In *Proceedings of 20th IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec 2006)*, pages 119–132, 2006.

[34] D. Zerkle and K. Levitt. Netkuang - a multi-host configuration vulnerability checker. In *Proceedings of the 6th USENIX Unix Security Symposium (USENIX'96)*, 1996.