Open access • Proceedings Article • DOI:10.1109/ARES.2008.138

VEA-bility Security Metric: A Network Security Analysis Tool — Source link

M. Tupper, A.N. Zincir-Heywood

Institutions: Dalhousie University

Published on: 04 Mar 2008 - Availability, Reliability and Security

Topics: Network security, Network simulation, Vulnerability (computing), Data security and Intrusion detection system

VEA-bility Security Metric: A Network Security Analysis Tool

Melanie Tupper
Dalhousie University
tupper@cs.dal.ca

A. Nur Zincir-Heywood
Dalhousie University
zincir@cs.dal.ca

Abstract

In this work, we propose a novel quantitative security metric, VEA-bility, which measures the desirability of different network configurations. An administrator can then use the VEA-bility scores of different configurations to configure a secure network. Based on our findings, we conclude that the VEA-bility can be used to accurately estimate the comparative desirability of a specific network configuration. This information can then be used to explore alternate possible configurations and allows an administrator to select one among the given options. These tools are important to network administrators as they strive to provide secure, yet functional, network configurations.

1. Introduction

Many of our everyday activities rely on services provided by computer networks. For this reason, a primary objective for a network/system administrator is to maintain a stable, secure network infrastructure. This objective includes ensuring that the network is hardened enough against malicious computer users, known as attackers or intruders.

Software vulnerabilities are weaknesses in software that attackers can use to gain or escalate network privileges. An exploit, or attack, is a way for the attacker to take advantage of vulnerabilities and can take the form of a software program, sequence of commands, or a block of data. If successful, the intruder will have gained privileges equal to that of the vulnerable program, allowing the intruder to access information or escalate privileges on the target host.

To minimize such problems, one approach that administrators use is network hardening, which refers to the various methods that can be employed to secure a system. These methods include patching software vulnerabilities and adding firewalls, demilitarized zones (DMZs), intrusion detection systems (IDSs), or intrusion prevention systems (IPSs).

Even with the best security practices, it is inevitable that systems become vulnerable. A common practice for detecting vulnerabilities on a network employs a vulnerability scanner. Once vulnerabilities have been identified, an administrator can attempt to fix the hole by downloading and installing the corresponding vulnerability patch from an online database. While these tools are useful for increasing security, new software vulnerabilities are still being discovered at an alarming rate of approximately 18 vulnerabilities per day [1]. Thus, exploit prevention has become an attractive research area.

Given this big arms race between attackers and administrators, it becomes much more important to configure a network that is as secure as possible. To this end, one question to investigate is how to compare different possible configurations of a system in order to select the most secure one given the constraints.

In general, a metric is a quantifiable measurement that allows for comparison. A security metric can be either qualitative or quantitative, and measures the degree of security controls, policies and procedures. Figure 1 shows two simple configurations for a network offering similar services. A quantitative security metric would allow an administrator to select the optimal configuration by determining which of the two configurations is better able to meet their security requirements.

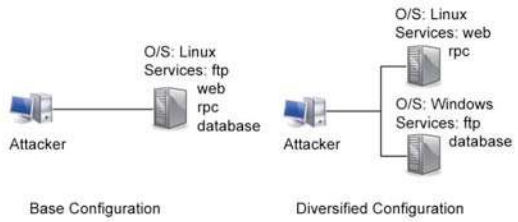

Figure 1. Two network configurations

A current limitation in the exploration of network security practices is the lack of quantitative security metrics. The purpose of this research is to propose a quantitative metric, VEA-bility, which can be used to

compare different physical and virtual network configurations. The underlying idea behind the VEA-bility metric is that the security of a network is influenced by many factors, including the severity of existing vulnerabilities, distribution of services, connectivity of hosts, and possible attack paths. These factors are modeled into three network dimensions: Vulnerability, Exploitability, and Attackability. The overall VEA-bility score, a numeric value in the range [0,10], is a function of these three dimensions.

The rest of this paper is organized as follows. Section 2 provides an overview of network security metrics in the literature. Section 3 describes attack graphs and their role in the VEA-bility score. Section 4 describes the proposed VEA-bility metric, which is applied to sample scenarios in Section 5. Section 6 presents conclusions and suggests further directions for this research.

2. Security metrics

As discussed in the previous section, comparing the desirability of different network configurations requires a security metric. A security metric measures the degree of security controls, policies and procedures. Taylor [2] recognizes a major void in the availability of quantitative security metrics with which to compare alternate configurations.

Since the definition of a secure network can be interpreted at different levels, it follows that there exists security metrics to compare different network components. The quantitative metric proposed by Pamula *et al.* [3] measures security based on the strength of the weakest adversary that can compromise the network. The algorithm they present employs an attack graph and starts with a goal state, then decomposes the requirements for the previous network state until an initial state is found. This produces the minimum set of initial attributes that an attacker would need to compromise a specified host. Annop *et al.* [4] explore the idea of a generic attack resistance metric that can be used to compare the overall security of a network. Similarly, Mayer's [5] operational metric is a quantitative metric that assigns a numeric score to each network host.

In short, the aforementioned metrics do one of two things: They either (i) combine different resources to produce a unified score for single hosts, or (ii) rely solely on the results of an attack graph to produce an overall network security score. However, our proposed metric does both: It (i) integrates three dimensions of input from various sources (including attack graphs), and (ii) defines a security score for the entire network. We will discuss how attack graphs work in the next section.

Metrics that measure the relative security of software services have also been proposed. Wysopal [6] proposes a metric for rating weaknesses found in software, while Manadhata *et al.* [7] use an attack surface metric to compare the attack surfaces of two specific ftp servers. Although these papers do not claim to compare the overall security of networks, they inspire our work to consider a network security metric with multiple dimensions.

The metric proposed by Adedin *et al.* [8] to evaluate network security policies generates a unified score that is a weighted aggregation of different factors. These factors include network vulnerabilities, vulnerability history of exposed services, exposure of services, and traffic volumes handled by services. This metric uses an exponential average to ensure that the resulting score will be at least as high as the highest vulnerability score present in the system. Additional vulnerabilities serve only to increase this score.

We also use an aggregated, unified security score as well as exponential averages in our own VEA-bility metric. However, in our proposed metric, the security of the network is evaluated with respect to the network configuration (physical components with different operating systems, services, etc.), whereas, in the systems above, the security policies are rated to enable the comparison of such policies. Naturally, this enables an administrator to decide whether changing a policy is better than maintaining the current one.

3. Attack graphs

While identifying single vulnerabilities is useful, security threats increase exponentially with multiple network vulnerabilities. One way to explore the effects of multiple vulnerabilities on a network is through an attack graph. An attack graph is a pictorial representation of the paths an attacker can take to exploit network vulnerabilities. The paths in the graph represent all the ways an intruder can penetrate the network. This information can be used to identify potential exploits and to determine what hardening measures should be taken to thwart attacks.

Traditionally, attack graphs were produced manually, which requires a substantial commitment of time and resources. Since network attack graphs are considered to be valuable tools for evaluating the security of a network, much work has been done in the area of automated attack graph generation and analysis. The main challenge of automating the generation of attack graphs is the exponential scaling with additional hosts. Therefore, most automated attack graph generators produce a pruned attack graph; the graph generated contains all the paths to a specific target host as opposed to the whole network.

A pruned attack graph is also useful to an administrator wishing to protect a critical server.

Amman *et al.* [9] present an algorithm that scales well, and is implemented by Jajodia *et al.* [10] in their Topological Vulnerability Analysis (TVA) tool. The TVA tool automatically imports results from Nessus Vulnerability scans, but also requires some manual input. Since TVA requires Nessus scan information, it can only be used on actual networks, not on network simulations directly.

Michael Artz's NetSPA tool [11] also requires information collected from Nessus scans, but must be manually entered into a database. On the other hand, the toolkit developed by Sheyner *et al.* [12] requires a user defined XML file describing the network for input. Therefore, this file can describe a simulated network configuration, allowing analysts to consider alternate configurations before implementation. Our research uses a toolkit based on the work of Sheyner *et al*. that has been updated by David Swasey [13]. We refer to this updated toolkit as the Sheyner/Swasey toolkit, and use it to form one dimension of the proposed VEA-bility metric.

4. VEA-bility analysis

The proposed VEA-bility security metric is defined to capture the numerous factors that influence the security of a network. To this end, we propose the VEA-bility metric to be a function of the security scores along three dimensions: Vulnerability, Exploitability, and Attackability. For simplicity, the vulnerability, exploitability, and attackability scores will be represented in equations as *V*, *E*, and *A*, respectively. Each of the three dimension scores is a numeric value in the range [0,10].

The VEA-bility metric uses data from three sources: network topology, attack graphs, and scores as assigned by the Common Vulnerability Scoring System (CVSS) [14], Figure 2.

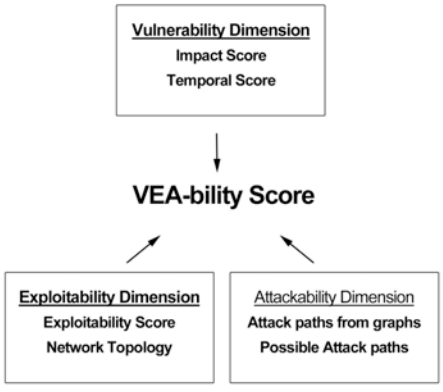

Figure 2. VEA-bility metric dimensions

Since a network is only as secure as its hosts, we define the three network dimensions as a function of the three dimensions for each network host. The network Vulnerability dimension is the exponential average of the host vulnerability scores, or a maximum of 10. On the other hand, the network Exploitability and Attackability dimensions are the summations of the Exploitability and Attackability scores of each host, respectively.

For a network, *N*, we define *V(host)*, *E(host)*, and *A(host)* as the 3 dimension scores for each network host with one or more vulnerabilities. A network void of vulnerabilities scores a 0 along each dimension; otherwise, we define the network dimensions Vulnerability (V_N), Exploitability (E_N), and Attackability (A_N) as functions of V(*host*), E*(host*), and A*(host*), respectively.

4.1 Network Vulnerability dimension

The Vulnerability dimension of the proposed VEA-bility metric is a function of two scores assigned by the CVSS [14]: the impact score and the temporal score for a known vulnerability. The impact score measures the impact that a successful exploit will have on the availability, integrity, and accessibility of information resources. The temporal score assigns a value based on the age of the vulnerability, the remediation status of a patch, and the credibility of the patch source. Thus, the vulnerability of a network is the degree to which an exploit can impact a system, a measure that is influenced by time. The vulnerability score of a network is the exponential average of the host vulnerability scores, or a maximum of 10. This captures the requirement that the vulnerability score of the network is at least as large as the largest host vulnerability score; additional vulnerable hosts serve only to increase this value, which can be a maximum of 10. Eq. (1) represents this dimension:

$$V_N = \min(10, \ln \sum e^{V(host)}) \quad (1)$$

4.2 Network Exploitability dimension

The Exploitability dimension is the sum of exploitability scores for each host on the network. This score is a function of the exploitability score assigned by the CVSS [14], which evaluates the likelihood of exploitation. Exploitability is affected by the ease of access to the host, authentication required, and whether or not a successful exploit of this type has been reported. Eq. (2) represents Exploitability as:

$$E_N = \sum E(host) \quad (2)$$

4.3 Network Attackability dimension

The Attackability dimension of the proposed VEA-bility metric is a summation of the Attackability scores of each host. Attackability is composed of information derived by generating an attack graph of the network. Eq. (3) defines Attackability as:

$$A_N = \sum A(host) \quad (3)$$

4.4 Host dimensions

We assume that a host with multiple vulnerabilities is less secure than a host with a single vulnerability, which is modeled into the vulnerability and exploitability dimensions by taking the exponential average of the values for all vulnerabilities. Again, this allows the value to be at least as large as the highest value, and additional scores serve to increase this value to a maximum value of 10.

Let each vulnerability, v, have an impact score, temporal score, and exploitability score as defined by the CVSS [14]. An impact and exploitability sub-scores are automatically generated for each CVE name, whereas the temporal score requires user input. We then define the severity, S, of a vulnerability to be the average of the impact and temporal scores, Eq. (4):

$$S(v) = (\text{Impact Score}(v) + \text{Temporal Score}(v)) / 2 \quad (4)$$

The host Vulnerability score is an exponential average of the severity scores of the vulnerabilities on a host, or 10, whichever is lower.

The host Exploitability score is the exponential average of the exploitability score for all host vulnerabilities multiplied by the ratio of network services on the host.

The host Attackability score is defined as the ratio of attack paths produced by attack graphs to total number of possible attack paths, and is multiplied by a factor of 10 to produce a number in the range [0,10], ensuring that all dimensions have the same range.

For a host, *host*, let v be a host vulnerability. We then define the three host dimensions as shown in equations (5), (6) and (7):

$$V(host) = \min(10, \ln \sum e^{S(v)}) \quad (5)$$

$$E(host) = (\min(10, \ln \sum e^{\text{Exploitability Score}(v)})) (\#\text{ services on host}) / (\#\text{ network services}) \quad (6)$$

$$A(host) = (10) (\#\text{attack paths}) / (\#\text{ network paths}) \quad (7)$$

The equation for network VEA-bility then becomes as in Eq. (8):

$$\textbf{VEA-bility}_N = 10 - ((V+E+A)_N / 3) \quad (8)$$

According to the NIST Security Metrics Guide for Information Technology Systems [15], a metric must yield quantifiable information, be useful for tracking system performance, measure a repeatable process, and the supporting data must be readily obtainable. Given these constraints, the metric we propose, VEA-bility, conforms to these standards in that it is quantifiable, that is, it is expressed as a numeric value. Moreover, by using the Nessus scanner and the Sheyner/Swasey toolkit, which are both freely available, the methods we employ can be easily repeated. Finally, the proposed metric serves to track the performance of a network configuration by comparing the score to other possible configurations. Thus, a system/network administrator can use the VEA-bility metric to direct or reallocate resources for network security hardening.

5. Experiments

The following discusses how the proposed VEA-bility metric can be used to compare various network configurations. To this end, we will present the data collection, feature selection, data/scenario modeling and analysis phases of our research.

5.1 Data collection

In this case, our goal is to produce simple but realistic scenarios in order to demonstrate how the score of the proposed security metric, VEA-bility, changes from one scenario to another. Thus, to accurately model a network scenario and to experiment with existing vulnerabilities, we use the Nessus Vulnerability Scanner [16] to collect network topology information on our faculty network.

The Nessus Scanner is an attractive network tool primarily because it allows “safe checks”, which do not attempt to exploit vulnerabilities. This setting enables the user to compile vulnerability information without causing harmful Denial of Service (DoS) attacks. The Nessus Scanner gathers information by sending requests to all ports on given hosts identified in the scan parameters. We use the scanner’s default settings, but limit the number of hosts scanned for each scan to 20 to avoid overwhelming hosts. The default range is 40 hosts per scan, as suggested in the Nessus 3.0 Client Guide [17].

We run the scans from a Windows XP platform. In total, we scan 250 hosts, and obtain results for 85 of these hosts. From this point onwards we will refer to this as our testbed. The testbed is comprised of hosts in diverse physical locations, and includes network servers, faculty machines, and student machines.

5.2 Feature selection

The purpose of the Nessus scans is to model realistic network scenarios. This requires a wealth of host information. From the scan results, we extract the following information:

- IP addresses
- Operating Systems
- Number of open ports
- Number of notes
- Number of warnings
- Number of holes
- Services running on the open port(s)
- CVE identification numbers and risk factors associated with vulnerabilities

The magnitude of information requires that we construct a more compact representation of data from which to choose a set of operating systems to use in our experiments. We reorganize the host information into categories by operating systems, further decomposing the operating systems by version or distribution. For each category we record:

- Number of hosts
- Number of warnings
- Number of holes
- Number of hosts with at least one vulnerability in each of the three highest risk factors

Based on our Nessus Scan results, shown in Table 1, and the vulnerabilities observed by our scan results, we develop an extensive set of scenarios to test the performance of the proposed VEA-bility metric. In section 5.3 we present a small subset of these scenarios due to page limitations. However, an interested reader can view the entire set of experiments in [18].

Table 1. Nessus Scan Results of the Testbed

	Mac OS X	Windows	Solaris	FreeBSD	Linux
#Hosts	16	10	7	2	23
#Warnings	8	11	45	0	26
#Holes	10	12	37	0	20
#Hosts w/h +1 medium CVE	0	3	4	0	4
#Hosts w/h +1 high CVE	0	0	4	0	3
#Hosts w/h +1 critical CVE	7	3	4	0	3

5.3 Sample network configurations

We start with a simple sample configuration, then step-by-step, add more devices and compare the configurations using the proposed VEA-bility metric.

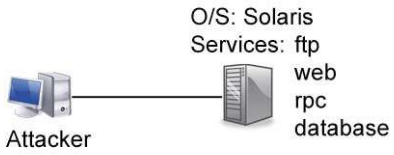

Figure 3. Network Configuration-1

Step-1: The simplest scenario to start with is the case of one host running four services and one malicious user, Figure 3. In this case, there is one host with the Solaris operating system. This system has a heap buffer overflow vulnerability that allows an attacker to possibly execute arbitrary code on the target host or cause a DoS attack. The CVE corresponding to this vulnerability is CVE-2004-0492. It has an impact score of 10, a temporal score of 8.7, and exploitability score of 10. Since in this scenario there is only one host, network dimensions are, in fact, host dimensions. It should be noted here that there is only one attack path on the attack graph of this scenario. Thus:

$$V_N = (10+8.7) / 2 = 9.35$$
$$E_N = 10\ (4/4) = 10$$
$$A_N = (1/1)\ 10 = 10$$
$$\textbf{VEA-bility} = 10 - ((9.35+10+10)/3) = \mathbf{0.22}$$

The above score (0.22) implies that this network configuration is very poor and has a very low security metric, i.e. can be compromised very easily by an attacker.

In order to show how an administrator might use the VEA-bility metric to compare different network configurations, we will now analyze other scenarios in the following steps.

Step-2: Configuration-2 results from isolating the database on the Solaris host and dispersing the other services onto an additional host running Windows operating system, Figure 4. One would expect that since this configuration results in a secure database server, the VEA-bility score would be higher than that of the previous configuration. Hence, this time the score becomes 3.6.

$$V_N = (10+8.7) / 2 = 9.35$$
$$E_N = 10\ (3/4) = 7.5$$
$$A_N = (1/4)10 = 2.5$$
$$\textbf{VEA-bility} = 10 - ((9.35+7.5+2.5)/3) = \mathbf{3.6}$$

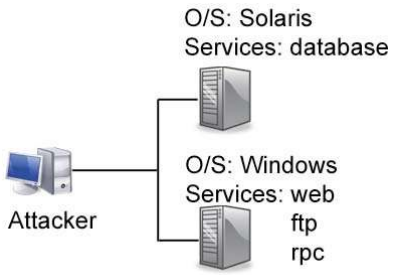

Figure 4. Network Configuration-2

Step-3: Figure 5 shows configuration-3. This is the result of isolating the database on the Solaris host, dispersing the remaining services onto an additional host running a Windows operating system, and adding a firewall between the attacker and internal network.

$V_N = (10+8.7) / 2 = 9.35$
$E_N = 10\ (3/4) = 7.5$
$A_N = (0/4)10 = 0$
VEA-bility = 10- ((9.35+7.5)/3) = **4.4**

There are no attack paths to hosts inside the network because the firewall restricts connectivity to the vulnerable rpc service. However, the fact that rpc service is vulnerable on Windows and 75% of the network services are on this host gives a resulting VEA-bility score of 4.4. This vulnerability could be exploited if the attacker discovers another way to get inside the network.

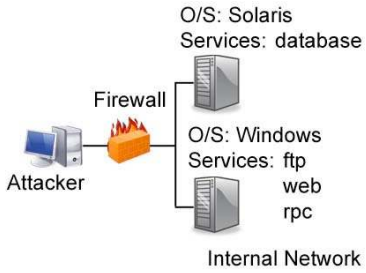

Figure 5. Network Configuration-3

Step-4: As shown in Figure 6, by isolating the database on the Solaris host inside the network, adding a Windows host for the remaining services, and shielding the network with a DMZ, we have configuration- 4.

$V_N = (10+8.7) / 2 = 9.35$
$E_N = 10\ (2/4) = 5.0$
$A_N = (0/5)10 = 0$
VEA-bility = 10- ((9.35+5.0)/3) = **5.2**

Again, there are no attack paths inside the network due to connectivity restrictions. However vulnerabilities still exist, even though the security metric is higher this time than the previous cases. This is the reason for the VEA-bility of 5.2 and reiterates our sentiment that the security of the network cannot be determined from attack graphs alone.

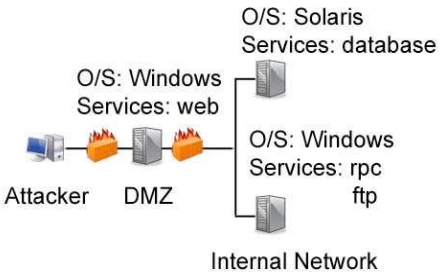

Figure 6. Network Configuration-4

Step-5: Finally, by adding another host, Linux, to the backend, we separate the services running on the Windows machine to get configuration-5, Figure 7.

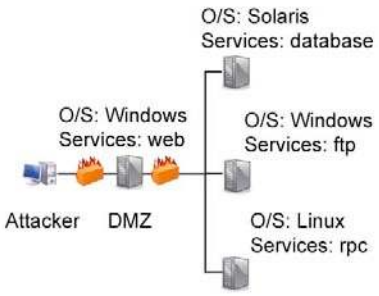

Figure 7. Network Configuration-5

$V_N = 0$
$E_N = 0$
$A_N = 0$
VEA-bility = 10- (0/3) = **10**

In this case, the VEA-bility score of 10 is achieved. This implies that this network configuration is secure (very VEA-ble) since there are no known vulnerabilities to exploit.

These examples also illustrate the value of our VEA-bility metric when considering the security of critical servers. For example, in the previous cases, consider the implications if the machine hosting the database is considered to be the critical server. When comparing dimension scores for the corresponding hosts, keep in mind that a lower dimension score is more favorable. In short, it is wiser to consider the configuration in step-5 to be the safest alternative among the configurations analyzed.

5.4 Summary of results

Our results for the Vulnerability dimension indicate that by introducing different operating systems to the network, it is possible to reduce the vulnerability dimension score. This occurs since running a vulnerable service on a different operating system removes the vulnerability, provided the software is not vulnerable on the alternate operating system.

Moreover, results for the Exploitability dimension show that exploitability is best controlled by the addition of a DMZ, but can be improved through introducing different hosts for different services. Since the exploitability dimension is related to the number of services on hosts with vulnerabilities, it seems reasonable that isolating the services to different hosts will best affect this dimension.

Furthermore, Attackability is a function of the ratio of attack paths to total paths through the network. This ratio is multiplied by 10 to generate a number compatible with the other two dimensions. Our results for the Attackability dimension highlight both the benefits of isolating services on different hosts and adding a DMZ on the network security as represented in the attack graphs. While many administrators use attack graphs alone to evaluate the security of a network [12], we propose that they are more useful when aggregated with other network factors. For example, consider a vulnerable host on a network that cannot be exploited due to connectivity restrictions. We consider this network less secure than a network with no software vulnerabilities, but more secure than a network with no connectivity restrictions. This is reflected in our VEA-bility metric score.

The overall average VEA-bility scores observed in our experiments are given in Figures 9 and 10. A higher score indicates a more secure configuration, which we call more "VEA-ble". Although these scores are averages, it is evident that it is possible to increase the security rating of a network configuration through isolation of services. Figure 8 is a pictorial representation of this data, which highlights the benefits of isolating services as well as introducing a firewall or DMZ. The significant advantage of isolation of services (as the operating systems increase the number of hosts increase) is apparent when comparing the average scores for the base configurations with three operating systems to the DMZ configurations with three operating systems. As indicated, the DMZ adds only 0.4 to the final VEA-bility score.

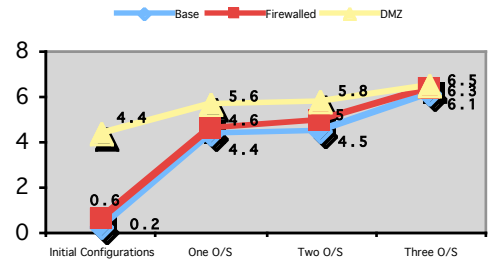

Figure 8. Average VEA-bility scores for different scenarios

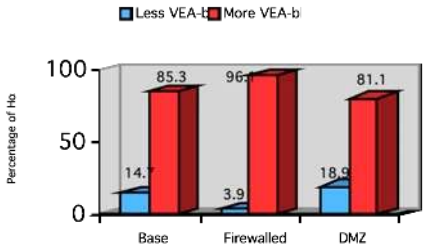

Figure 9. VEA-bility score distribution

On the other hand, Figure 9 shows the distribution of network VEA-bility scores. This figure shows the distribution of configurations that resulted in lower and higher VEA-bility scores than the average initial scores of 0.22, 0.6, and 4.4. To reduce influencing the results by including firewalled and DMZ configurations, we divide the results into three categories: base, firewalled, and DMZ.

6. Conclusions and future work

In this work, the objective was to develop a quantitative security metric with which to compare the attractiveness of different potential network configurations. To this end, we propose the VEA-bility metric. The proposed metric assigns a numeric value in the range [0,10] to each network configuration where zero indicates a poorly configured network and ten indicates the most secure network configuration possible. Using our VEA-bility metric applied to a set of sample scenarios, we find that the VEA-bility metric accurately rates the comparative desirability of different configurations.

There are a number of directions for extending this research including improving the network model used to generate the Attackability dimension, improving the metric itself, and using the VEA-bility metric to investigate specific aspects of network security.

In this research, we did not have access to information provided by an IDS or IPS on the testbed. Obtaining this information and including it in the network model would allow an administrator to make more confident decisions regarding secure network topologies. Also, continuing from the work of Sheyner *et al.* [12], we did not model trust relations within the network, but rather modeled the resulting authentications as connectivity relations. Since the Sheyner/Swasey toolkit is designed to recognize host trust relations, representing these relations can provide more accurate representations of a network configuration.

One way to improve the VEA-bility metric includes adding more information provided by the CVSS [14]. One such example is the environmental score, which assigns a numeric value based on software implementation and network environment. The environmental score is calculated based on user defined input such as the potential for damage.

Our VEA-bility metric could also be used to study broader network security concerns such as investigating whether network diversity has an impact on network security or which of our three defined dimensions has a greater impact on the overall security. The results of these types of studies would better allow administrators to focus their efforts on events that would have the most impact on the security of their networks.

Acknowledgements

The authors would like to thank *The Computer Research Association's Committee on the Status of Women in Computing Research* (CRA-W) and *The Natural Sciences and Engineering Research Council of Canada* (NSERC) for supporting this research. We would also like to thank the entire Tech Support team at Dalhousie University for their cooperation and assistance for the duration of this project. This work is conducted as part of the NIMS project at http://www.cs.dal.ca/projectx/.

References

[1] National Vulnerability Database: http://nvd.nist.gov.

[2] Taylor C., Alves-Foss J., Diversity As a Computer Defense Mechanism, Proceedings of the ACM Workshop on New Security Paradigms, pp. 11-14, 2005.

[3] Pamula, J., Jajodia, S., Ammann, P., Swarup, V., A Weakest Adversary Security Metric for Network Configuration Security Analysis, Proceedings of the ACM Workshop on Quality of Protection, pp. 31-38, 2006.

[4] Wang, L., Anoop, S., Jajodia, S., Toward Measuring Network Security Using Attack Graphs, Proceedings of the ACM Workshop on Quality of Protection, pp. 49-55, 2007.

[5] Mayer, A., Seal R., Operational Security Risk Metrics: Definitions, Calculations, Visualizations, Metricon 2.0, http://www.securitymetrics.org/content/attach/Metricon2.0/Mayer_Metricon-Final.ppt, 2007.

[6] Wysopal C., Software Security Weakness Scoring, Metricon 2.0, http://www.securitymetrics.org/content/attach/Metricon2.0/Wysopal-metricon2.0-software-weakness-scoring.ppt, 2007.

[7] Manadhata, P., Wing, J., Flynn, M., McQueen, M., Measuring the Attack Surfaces of Two FTP Daemons, Proceedings of the ACM Workshop on Quality of Protection, pp. 3-10, 2006.

[8] Abedin, M., Nessa, S., Al-Shaer, E., Khan, L., Vulnerability Analysis For Evaluating Quality of Protection Security Policies, Proceedings of the ACM Workshop on Quality of Protection, pp. 49-52, 2006.

[9] Ammann, P., Wijesekera, D., Kaushik, S., Scalable, Graph-Based Network Vulnerability Analysis, Proceedings of the ACM Conference on Computer and Communications Security, pp. 217-224, 2002.

[10] Jajodia, S., Noel, S., O'Berry, B., Topological Analysis of Network Attack Vulnerability, Managing Cyber Threats: Issues, Approaches and Challenges, Chapter-9, pp. 248-266, V. Kumar, J. Srivastava, and A. Lazarevic (Eds.), Springer-Verlag, 2005.

[11] Artz, M., NetSPA, A Network Security Planning Architecture, M.S. Thesis, Massachusetts Institute of Technology, 2002.

[12] Sheyner, O., Wing, J. M., Tools for Generating and Analyzing Attack Graphs, Proceedings of the Workshop on Formal Methods for Components and Objects, pp. 344-371, 2004.

[13] Scenario and Attack Graphs: http://www.cs.cmu.edu/~scenariograph.

[14] A Complete Guide to the Common Vulnerability Scoring System (CVSS): http://www.first.org/cvss/v1/guide.html.

[15] Security Metrics Guide for Information Technology Systems: http://csrc.nist.gov/publications/nistpubs/800-55/sp800-55.pdf.

[16] Nessus Vulnerability Scanner: http://www.nessus.org.

[17] Nessus Client Guide: http://www.nessus.org/documentation/nessus_3.0_client_guide.pdf.

[18] Melanie Tupper, CDMP Technical Report, 2007: http://www.cra.org/Activities/craw/cdmp/awards/2007/Tupper/tuppercdmp/CDMP_Report.pdf.