Volume 3, Issue 9, September 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Review of Network Security Metrics

Tito Waluyo Purboyo[1] , Kuspriyanto[2]
[1,2]*School of Electrical Engineering and Informatics,*
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40553, Indonesia

***Abstract*— A graph based on an attack can be used by a network administrator to measure an enterprise networks accurately. A network administrator can understand the most critical threats and choose the best countermeasures. Attack graph is an integral part of modeling the overview of network security. Network administrators use attack graphs to know how vulnerable their systems and to choose what security measures to maintain the system to deploy them. An attack graph is an abstraction that represents how an attacker may violate security policy by exploiting the interdependence between the discovered vulnerabilities. Analyses of attack graph that extract an information relevant to security from the attack graph are referred to as attack graph-based security metrics. Attack graphs are a valuable tool to network administrators, describing paths which can be used by an attacker to gain access to a targeted network. Network administrator can then focus their efforts on correcting the vulnerabilities and configuration errors that allow the attackers exploiting these vulnerabilities.**

***Keywords*— Network Security, Metrics of Security, Graph of Attack**

I. INTRODUCTION

Evaluating the computer network security through the analysis of the system information is very important and could protect us from an attack to the network. When the enterprise network security is analyzed, considering mulsti-stage, multi-host attacks are vey important. In particular, an attack graph that illustrates all possible multi-stage, multi-host attack paths is crucial for a system administrator to understand the nature of the threats and decide upon appropriate countermeasures [1]. An Attack Path specifies an attack scenario that results in compromising organization values. It tells us how an attacker gains access to the victim computer; how and which vulnerability attacker can take advantage of and what kind of damage may be done that can impact the organization [2]. When defending an isolated network with resources that critical, some vulnerabilities may not seem significant. Attackers can often intrude a seemingly well-guarded network using multi-step attacks by exploiting a related vulnerabilities sequences. Attack graphs can recognize such potential threats by enumerating all possible sequences attackers can exploit [3]. To protect critical resources in today's networked environments, it is desirable to quantify the probability of multi-step attacks which potential to combine multiple vulnerabilities. This fact becomes feasible due to a model of causal relationships between vulnerabilities, namely, attack graph [4]. 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 total measure of network security [5]. By measuring risk for enterprise networks precisely, attack graphs allow network defenders to understand the most critical threats and select the most effective countermeasures [6]. Even a network of moderate size can have dozens of possible attack paths, confusing a human user with the amount of information described. It is not easy for a human to determine from the information in the attack graph which configuration settings should be changed to best address the identified security problems. Without a clear observing of the existing security problems, it is hard for a human user to evaluate possible configuration changes and to verify that optimal changes are made [7]. The generation of attack graph requires knowledge of the prerequisites required for vulnerability exploitation and of the effect of exploitation on attacker privileges and the network [8].

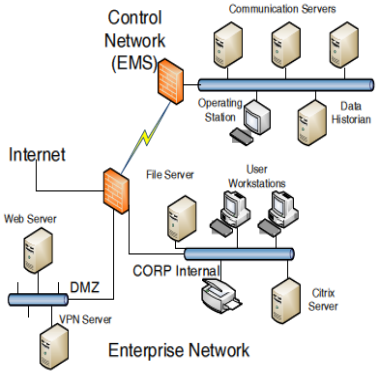

Fig. 1 An example enterprise network [7]

Figure 1 depicts an example enterprise network that is based on a real (and much bigger) system; we will return to this example later. The network includes three subnets: a DMZ (Demilitarized Zone), an internal subnet, and an EMS (Energy Management System) subnet, which is a control-system network for power grids. In this example, it is assumed that host-grouping has already been applied, based on similar configurations; the workstation node, as an example, might be an abstracted grouping of one hundred workstation machines with similar setups. The web server and the VPN server are directly accessible from the Internet. This web server can access the file server through the NFS file-sharing protocol; the VPN server is allowed access to all hosts in the internal subnet. Outside access to the EMS subnet is only allowed from the Citrix server in the internal subnet, and even can only to the data historian. In this example, we assume that the attacker's goal is to gain privileges to execute code on the commServer. From the commServer, an attacker could send commands to physical facilities such as power-generating turbines, which can cause grave damage to critical infrastructures [7].

As networks of hosts and security incidents continue to grow, it becomes increasing more important to automate the process of evaluating network security. Attack graphs are composed by all the attack paths which lead to intruders intentions. The attack path is formed by a chain of exploits, where each exploit is realized by taking advantage of known vulnerabilities in various of services and systems. The term vulnerabilities refers to exploitable errors in configurations and server software implemented to provide network services [13].

II. ATTACK GRAPH GENERATION

In [9], explained that an Attack graph is a directed graph representing prior knowledge about interdependencies among vulnerabilities and network connectivity. The vertices of an attack graph are divided into two categories. They are exploits and security conditions (or simply conditions when no confusion is possible). First, exploits are actions taken by attackers on one or more hosts in order to take advantage of existing vulnerabilities. We denote an exploit as a mention. For example, an exploit involving three hosts can be written as v(hs,hm,hd), which indicates an exploitation of the vulnerability v on the destination host (hd), initiated from the source host (hs), through an intermediate host (hm). Similarly, we denote v(hs,hd) or v(h), where v(hs,hd) represents an exploits involving two hosts (no intermediate host) and v(h) represents an exploits involving one (local) host. Attack graph generation requires knowledge of the prerequisites required for vulnerability exploitation and of the effect of exploitation on attacker privileges and the network. In [9] also presented a new metrics called the attack resistance metric.

In [11] explained that attack graphs are used to determine if designated goal states can be reached by attackers attempting to penetrate computer networks from initial states. For this use, they are graphs in which the starting node represents an attacker at a specified network location. Nodes and arcs represent actions the attacker takes and changes in the network state caused by these actions. Actions usually involve exploits or exploit steps that take advantage of vulnerabilities in software or protocols. The goal of these actions is for the attacker to obtain normally restricted privileges on one or more target hosts, where the target could be a computer user, a router, a firewall, or some other network component. Many actions that compromise separate hosts and use them as stepping stones may be required in large attack graphs to reach the target host. A full attack graph will describe all possible sequences of attacker actions that eventually lead to the desired level of the target privilege. Some researchers use nodes to represent network states and arcs to represent attack actions, and some researchers use other representations, including those in which both actions and network states are nodes and in which actions are nodes and network states are arcs. In addition, some attack graphs have one attacker starting location and one target host, some have multiple targets, and some have multiple attacker starting locations.

In [12] presented a new approach that uses configuration information on firewalls and vulnerability information on all network devices to build attack graphs that show how far inside and outside attackers can progress through a network by successively compromising exposed and vulnerable hosts. Moreover, attack graphs are automatically analyzed to produce a small set of prioritized recommendations to enhance network security.

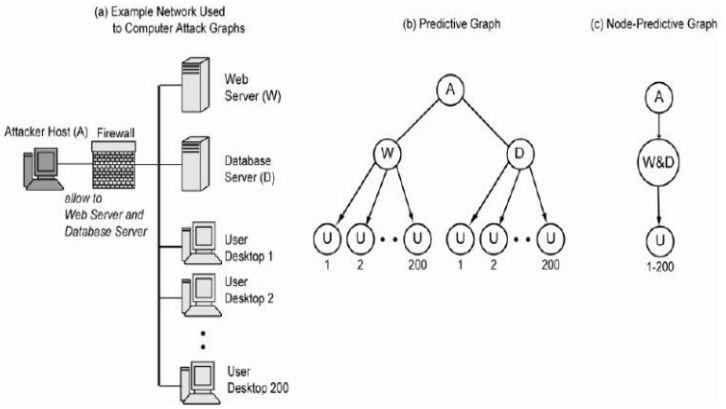

Fig. 2 Example of simple network used to compute attack graphs (a). The predictive graph (b) has repeated structure. The attack on the 200 user desktops is shown from both the web server and the database server. The nodepredictive tree (c) simplifies this topology. The web server and the database server are collapsed into a single host group labeled W&D, and all user desktops are collapsed into a single host group labeled U. [12]

Predictive graphs occasionally produce large amounts of seemingly redundant structure. As an example, consider the simple network on the left of Figure 2, consisting of an Attacker Host (A), a firewall, a Web Server (W), a Database Server (D), and 200 User Desktop hosts (U). Every host behind firewall has a single remote-to-admin vulnerability. The firewall allows the attacker to directly compromise the web server and the database server, but not the user desktops. This scenario's predictive attack graph is shown in the middle of Figure 2. In [14] presented a novel multi-faceted approach to separately combine CVSS base metrics. More spesific, instead of taking the base score as a black box input, the approach breaks it down to the underlying base metrics. At the base metric level, dependency relationships between vulnerabilities have well-defined semantics and can thus be handled. The approach also interprets CVSS scores in three different aspects, namely, probability, effort, and skill. The scores need to be combined in different ways for different aspects.

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, 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 [5].

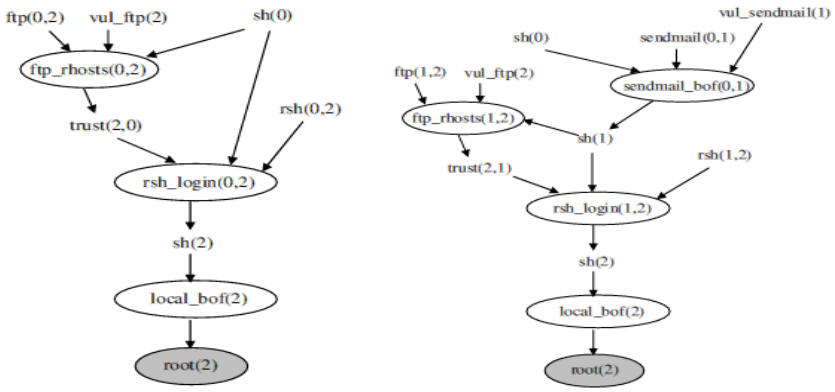

Fig. 3 Two Network Configurations [5]

A block diagram of NetSPA is shown in Fig. 4. The left side of this figure shows that NetSPA automatically imports the following into an internal database:

1) Vulnerability Scans, such as those generated by Nessus, that list where vulnerabilities are in the network and provide information on individual hosts and open ports.
2) Vulnerability Databases, such as NVD, that describe the prerequisites for and the effects of exploiting vulnerabilities.
3) Rules of Firewall, such as Sidewinder rulesets, that describe how traffic may or may not flow through a filtering device.
4) Information of Topology that specifies how firewalls and hosts from vulnerability scans are connected together.

The first three items in the list can be obtained automatically and imported, but topology information needs to be provided by hand. This information is limited, doesn't change often, and is relatively easy to enter. The right side of Figure 4 shows that NetSPA uses imported data to perform three main tasks. These are (1) Compute reachability, (2) Create an attack graph, and (3) Generate recommendations. Algorithms to perform these tasks that are efficient in time and space are described in [8].

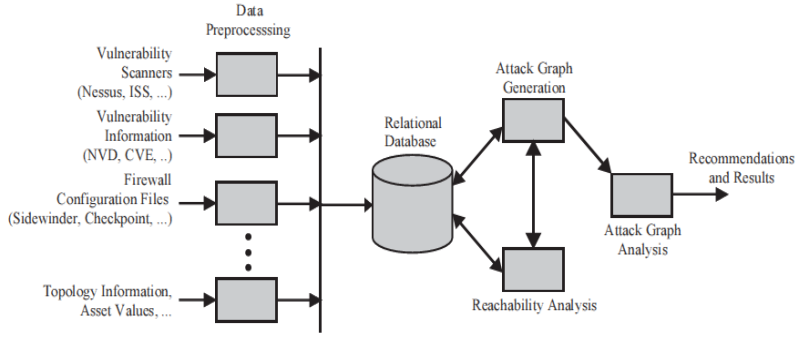

Fig. 4 NetSPA System Block Diagram [8]

III. ATTACK GRAPH-BASED SECURITY METRICS

As the hosts networks grow continuously, it becomes increasingly more important to automate the process of evaluating their vulnerability to attack. When evaluating the network security, it is usually not enough to consider the presence or absence of isolated vulnerabilities. Big networks typically contain multiple platforms and software packages and employ several modes of connectivity. So that, such networks have security holes that escape notice of even the most diligent system administrator [15]. Attack graphs can serve as a useful tool in several areas of network security, including intrusion detection, defense, and forensic analysis. A further benefit of attack graphs is that they can help analyze potential effectiveness of many different security measures offline. In [15] explained how both the security policy and the intrusion detection system can be incorporated explicitly in the attack model. The system administrator can then perform several kinds of analysis on such configuration-specific attack graphs to assess the security needs of the network. The K-step Condition Accumulation (KCA) metric is proposed in [10]. The K-step Condition Accumulation (KCA) metric specifies the "power" the attacker can obtain on a network in K steps. Practically, "power" represents the capability an attacker attains. Capabilities are controlled by access controls. Therefore, power may be represented by the privilege(s) the attacker attains on a machine. Therefore, if an attacker can obtain more capabilities on Sys1 in K steps, than the attacker can on Sys2 in K steps, then Sys2 is more secure than Sys1. More generally, if an attacker can obtain strictly more power in Sys1 than in Sys2 in K or less steps, then Sys2 is more secure than Sys1.

Experience has shown that organizations are reluctant to change their security policy or patch their machines even in the face of evidence that their network is vulnerable to attack. Sometimes it is impractical to keep all of the machines up to date—some networks contain thousands of hosts, and even with the help of automation applying frequent security patches is a challenge. Organizations sometimes keep a relatively lax security policy because tightening it prevents legitimate users from using the network to do their work. For example, an organization may choose to keep an FTP server open, or give its employees remote access to the network.

Compiling a list of vulnerabilities with a security scanning tool may not be enough to persuade organization that action is needed. In [15], they show that existing vulnerability scanners, such as Nessus, can be connected with automated attack graph generation software. Such a combination automatically scans a network and generates customized attack graphs. An attack graph showing multiple concrete break-in scenarios that could be executed on the organization's network could be used as persuasive evidence that the organization's security policies are too lax.

IV. CONTRIBUTION

This paper discussed some of network security metrics based on attack graph for analyzing the vulnerability of computer networks. A new results from our research include the metrics and methods for measuring network security. The originality from this work include the proposed metrics [25] and the methods as can be seen in Figure 5. In the studies conducted until now, we proposed a new method that can be seen in the Figure 5. The proposed method will be implemented using the software that will be developed by the researchers. A simulation study which implementing our framework will be done in the next paper.

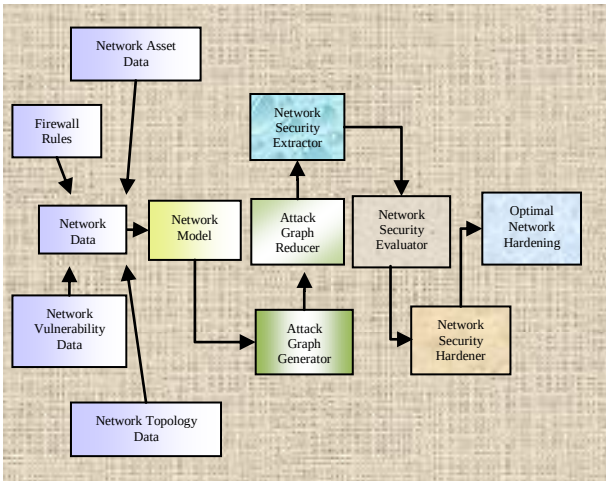

Fig. 5 A New Methods for Measuring Network Security

V. FUTURE RESEARCH TOPICS ON ATTACK GRAPH-BASED SECURITY METRICS

Future work consists of developing a formalized model for applying the attack graph to individual hosts. There is nothing inherent in attack graphs that make them exclusively useful for networks. Providing such a model has the important benefit of making evaluation of attack graph-based security metrics on real systems easier to accomplish. By assessing a single host, researchers would have the ability to assess how well different attack graph-based security metrics correspond to security incidents without putting in the effort to build an entire network [10]. Liu in [20] explained that it would be necessary in future work to develop explicit attackability prediction models. The current approach to attackability prediction is an indirect one, in the sense that it relies on the empirical security design hypotheses outlined in [20]. We base ourselves on these hypotheses to infer some relationships between attackability and

internal software attributes, and use these relationships to guide security analysis process. It would be interesting to go beyond this implicit model by formalizing these relationships under the form of, for instance, some mathematical equations or formulas explicitly linking attackability metrics to internal software metrics. Using regression techniques, it might be possible to develop such models. But once again for such models to be viable, we would need to validate those empirically using field data, which as mentioned above has yet to be publicly available.

Another important issue that should be addressed in future work is the need to develop aggregate attackability prediction models. The current approach as explained in [20] is to develop attack specific models. Although this approach is convenient from research perspective it can be cumbersome in practical environments. In practice, there are many different kinds of software attack patterns, and new ones are being invented regularly. Developing a separate attackability model for each of those attack patterns is impossible. At this stage of the work, a recommendation for practitioners is to conduct proper risk analysis at the beginning of the design process, and prioritize the identified risks. Attackability analysis can then focus only on the most important risks in terms of assets and feasibility. Liu explained [20] that the goal, in future work, is to go beyond the current paradigm by developing aggregate attackability models which can be used to predict any kind of security attacks, known or unknown. This might necessitate using machine learning techniques to predict general attacking behaviors and conditions.

VI. CONCLUSION

This paper discussed some of network security metrics based on attack graph for analyzing the vulnerability of computer networks. A new results from our research include the metrics and methods for measuring network security.

Attack graph provides a powerful way to understand the context and the relative importance of vulnerabilities in systems and networks. Attack graph analysis depends on complete and accurate model of the network. Such models are usually built using data from network (remote) vulnerability scanners such as Nessus. However, the scanning range has a fundamental limitation on the information available about the target host. Our future work is to improve the method and developing a model for vulnerability analysis including metrics in [25]. A simulation study also will be improved in our next paper.

REFERENCES

[1] Z. Lufeng, T. Hong, C. YiMing, Z. JianBo, "Network Security Evaluation through Attack Graph Generation", World Academy of Science, Engineering and Technology 54, 2009.

[2] S. Khaitan, S. Raheja, "Finding Optimal Attack Path Using Attack Graphs: A Survey", International Journal of Soft Computing and Engineering (IJSCE), Volume-1, Issue-3, July 2011.

[3] F. Chen, A. Liu, Y. Zhang, J. Su, "A Scalable Approach to Analyzing Network Security using Compact Attack Graph", JOURNAL OF NETWORKS, VOL. 5 NO. 5, 2010, pp. 543-555.

[4] L. Wang, T. Islam, T. Long, A. Singhal, S. Jajodia, "An attack graph-based probabilistic security metric", DAS 2008, LNCS 5094, 2008, pp. 283–296.

[5] L. Wang, A. Singhal, S. Jajodia, "Toward Measuring Network Security Using Attack Graphs", QoP'07, Alexandria, Virginia, USA, October 29, 2007.

[6] K. Ingols, M. Chu, R. Lippmann, S. Webster, S. Boyer, "Modeling Modern Network Attacks and Countermeasures Using Attack Graphs", Annual Computer Security Applications Conference (ACSAC) 25th. 2009.

[7] J. Homer, A. Varikuti, X. Ou, M.A. McQueen, "Improving Attack Graph Visualization Through Data Reduction and Attack Grouping", Workshop on Visualization for Computer Security (VizSEC), 2008.

[8] R. Lippmann, K. Ingols, C. Scott, K. Piwowarski, K. Kratkiewicz, M. Artz, R. Cunningham, "Validating and restoring defense in depth using attack graphs", Military Communications Conference, October 2006.

[9] L. Wang, A. Singhal, S. Jajodia, "Measuring overall security of network configurations using attack graphs", Data and Applications Security XXI, vol. 4602, 2007, pp. 98–112.

[10] N.C. Idika, "Characterizing and Aggregating Attack Graph-Based Security Metrics", Ph.D. Thesis, Purdue University, West Lafayette, Indiana, 2010.

[11] R.P. Lippmann, K.W. Ingols, "An Annotated Review of Past Papers on Attack Graphs", Project Report, MIT, 2005.

[12] R.P. Lippmann, K.W. Ingols, C. Scott, K. Piwowarski, K.J. Kratkiewicz, M. Artz, R.K. Cunningham, "Evaluating and Strengthening Enterprise Network Security Using Attack Graphs", Project Report, MIT, 2005.

[13] F. Chen, C. Wang, Z. Tian, S. Jin, T. Zhang, "An Atomic-Domains-Based Approach for Attack Graph Generation", World Academy of Science, Engineering and Technology 56, 2009.

[14] P. Cheng, "A Multi-Faceted Approach to Network Security Metric Through Combining CVSS Base Scores", MSc. Thesis, Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Quebec, Canada, 2011.

[15] O. Sheyner, "Scenario Graphs and Attack Graphs", Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, 2004.

[16] O. Sheyner, J. Haines, S. Jha, R. Lippmann, J. Wing, "Automated Generation and Analisys of Attack Graph", In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, May 2002.

[17] L. Wang, C. Yao, A. Singhal, S. Jajodia, "Implementing interactive analysis of attack graphs using relational databases", Journal of Computer Security 16, 2008, pp. 419–437.

[18] R. Sawilla, X. Ou. Identifying Critical Attack Assets in Dependency Attack Graphs. ESORICS 2008, LNCS 5283, pp. 18–34, 2008.
[19] X. Ou, W. Boyer, M. McQueen, "A Scalable Approach to Attack Graph Generation", In 13th ACM Conference on Computer and Communications Security (CCS), 2006, pp. 336–345.
[20] O. Sheynar, J. Wing, "Tools for Generating and Analyzing Attack Graphs" FMCO 2003, LNCS 3188, 2004, pp. 344–371.
[21] V. Mehta, C. Bartzis, H. Zhu, E. Clarke, J. Wing, "Ranking Attack Graphs", In Proceedings of Recent Advances in Intrusion Detection (RAID) 2006, LNCS 4219, 2006, pp. 127–144.
[22] M.S. Ahmed, E. Al-Shaer, E. Khan, "A novel quantitative approach for measuring network security", Proceedings of IEEE INFO COM, 2008.
[23] K. Ingols, R. Lippmann, K. Piwowarski, "Practical Attack Graph Generation for Network Defense", In 22nd Annual Computer Security Applications Conference (ACSAC), Miami Beach, Florida, December 2006).
[24] L. Williams, R. Lippmann, K. Ingols, "An Interactive Attack Graph Cascade and Reachability Display", Workshop on Visualization for Computer Security (VizSEC), 2007.
[25] T.W. Purboyo, B. Rahardjo, Kuspriyanto, I.M. Alamsyah, "A New Metrics for Predicting Network Security Level", Journal of Global Research in Computer Science, Volume 3 No. 3, March 2012.

