

# A Survey on Network Security Hardening Models

Tito Waluyo Purboyo<sup>1</sup>, Kuspriyanto<sup>2</sup>

<sup>1,2</sup>School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40553, Indonesia

## Abstract

*In order to secure an organization's network assets, a network administrator must determine how to harden the network. Network administrators are often faced with a more challenging problem since they have to work within a fixed budget which may be less than the minimum cost of system hardening. Their problem is how to select a subset of security hardening measures so as to be within the budget and yet minimize the residual damage to the system caused by not plugging all required security holes. To aid the decision-making process, network administrators may use attack graphs, which, through analysis, yield network hardening suggestions. Researchers have previously looked into the problem of determining if a given set of security hardening measures can effectively make a networked system secure. Many of them also addressed the problem of minimizing the total cost of implementing these hardening measures, given costs for individual measures. In this work, we survey a systematic approach to solve this problem by formulating it as a mathematical model.*

**Keywords:** Mathematical Model, Network Security, Security Hardening, Security Metrics, Attack Graph

## 1. INTRODUCTION

Information security is an important aspect of a successful business today. However, financial difficulties and budget cuts create a problem of selecting appropriate security measures and keeping networked systems up and running. Economic models suggested in literatures do not address the challenging issues of security countermeasure selection. We have surveyed some network security hardening models, which can be used to harden a system in a cost effective manner based on the methodologies used.

Network vulnerability is the leading reason of network attacks. Attack graphs have been proposed by many researchers as a tool for vulnerability analysis, but the biggest challenge is how the users can understand them quickly. So the study on network security hardening is very important and has practical significance.

It is difficult to secure large corporate networks. The network is vulnerable to physical attacks on network components, social engineering attacks where users are forced to disclose sensitive information, and cyber-attacks in which malicious attackers achieve a higher level of permissions than it should be allowed through a network connection. Analysis of cyber-attacks is difficult because of the compromised systems often provides a launching pad that can be used to launch further attacks. An attacker can often jump from system to system, eventually achieving a goal such as compromising an essential mail or file server [1].

In the analysis of network vulnerability to attack, considering the vulnerabilities in isolation is not enough. This is because attackers often combine multiple exploits against vulnerabilities to achieve their goals. While no single vulnerability may pose a significant threat to a network, the combination of vulnerabilities allows it. So even given the network could be vulnerable to attack, because the consequences of security that offers a variety of combined services. Attack graphs (and even exploit dependency graph) shows the sequence of exploitation, which may be useful to use the focus on the attack itself. But network administrators usually do not care about the order to exploit - they just want to know the best way to harden their networks. What is needed is a set of explicit and easily managed network hardening options that provide a guarantee for the safety of a given network resources [2].

The existing approach to build an attack graphs is to represent it as attack paths, i.e., the possibility sequences that an attacker could exploit this vulnerability during a multi-step intrusion. However, while the graph shows the threat of attack, they do not directly provide a solution to harden networks against them. Removing vulnerabilities usually incurs different costs, and in practice it is usually infeasible to remove all identified vulnerabilities. A critical but unanswered question in defending against multi-step intrusions is thus: which of the vulnerabilities should be removed, such that none of the attack paths leading to given critical resources can be realized, where such removal incurs the least cost? Finding answers to these questions manually is error-prone and tedious, and become unfit for larger networks and less secure that the attack graph is too complicated [3].

Network administrators fulfill the duty of preventing network attacks by identifying vulnerabilities in the network and then systematically removing the identified vulnerabilities. The removal of an identified vulnerability from a network may be referred to as a patch or a security measure. A security measure is any action performed to remove at least one vulnerability from a system. Subsequently, the set of all security measures is infinite. However, practically, a network

administrator will consider only a finite set of security measures for possible application to the network she is protecting. General security measures include, for example, modifying firewall rules, updating software on networked hosts, shutting down system services, or modifying an authentication routine [5].

As stated in [4] [6], there are so many open problems in the security metrics area including application security, network security, software security, etc.

The discussion in this paper is further organized as follows. The next section consists of reviews of the attack graph. In section 3 an example of which can provide motivation. Section 4 describes a mathematical model for network security hardening. Finally, in Section 5 provided a conclusion to what has been discussed in this paper.

## 2. ATTACK GRAPH

Attack graph describes the prior knowledge about the vulnerability, their interdependence, and network connectivity. An attack graph can be represented as a directed graph with two types of vertices, exploits and security conditions (or simply conditions when no confusion is possible) [3].

An attack graph is a concise representation of all the ways an attacker might exploit a known vulnerability breaking the security policy set. Each line in the graph corresponds to at least one attack scenario in the attack where the attacker reaches its destination. An attack scenario is a sequence of actions that move the network from its initial state to a compromised state. The initial state corresponds to the initial configuration of the network, and the compromised state corresponds to the state where the security policy violation(s) occurs [5].

The attack graphs model how multiple vulnerabilities may be combined for an attack. They represent the system state using a collection of security-related conditions, such as the existence of vulnerabilities in specific hosts or connectivity between different hosts. Vulnerability exploitation is modeled as a transition between system states [9].

Each network can be regarded as a collection of systems that provide various services to its clients or users. When considering security, the measurement of security metrics must be able to produce a value and expressed as real number or percentage [6].

The attack graph is a concise representation of all the ways an attacker may compromise a security policy through leveraging dependencies among known vulnerabilities. The attack graph is derived from a network model description that consists of at least the following elements: extant vulnerabilities on hosts, host connectivity, and usually at least one security policy. Extant vulnerabilities may be discovered through searching online vulnerability repositories and/or using vulnerability scanners. Note that a vulnerability scanner does not have to identify vulnerabilities that correspond to vulnerabilities found in an online repository. The network connectivity can be specified by using firewall rules and tools such as netstat. The security policies may be obtained from the Chief Security Officer (CSO) of the organization for which the attack graph is being used [11].

## 3. A MOTIVATING EXAMPLES

The attack graphs model how multiple vulnerabilities may be combined for an attack. They represent the system state using a collection of security-related conditions, such as the existence of vulnerabilities in specific hosts or connectivity between different hosts. Exploitation of the vulnerability is modeled as transitions between system states. For example, consider Figure 1. The left side shows the network configuration, and the right side shows a graph of attacks to compromise the database server by a malicious user workstation. In this network configuration, the firewall is intended to help protect the internal network. Internal file server that offers file transfer (ftp), secure shell (ssh), and remote shell (rsh) services. Server internal database offers ftp and rsh services. The firewall allows ftp, ssh, and rsh traffic from a user workstation to both servers, and blocks all other traffic [9].

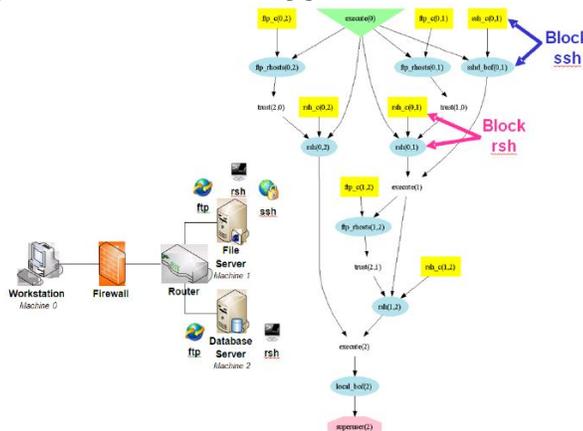
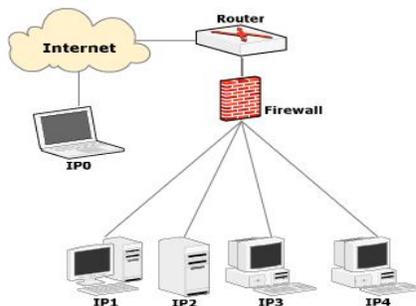


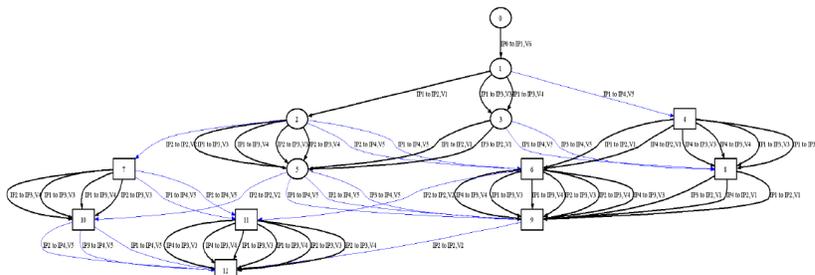
Figure 1. Example network, attack graph, and network hardening choices [9]

In the attack graph, the attacker exploits is a blue oval, with an edge to their preconditions and post conditions. The numbers in parentheses indicate the source and destination hosts. Yellow box is the initial condition of the network, and the green triangle is the ability to start attacker. Condition caused by an attacker exploits is plain text. Overall aim of the attack is a red octagon. The figure also shows the direct impact of traffic blocking ssh or rsh (the file server) through the firewall, i.e., prevent certain exploits in attack graphs.



**Figure 2.** Experimental network [12]

Zhao et. al. [12] constructed an experimental network to illustrate their algorithm. In this case, a small local area network composed by 4 computers, as shown in Figure 2, has a firewall to the outside, and the attacker is located outside the firewall. In Figure 2, the attacker has root privilege on his own computer IP0. IP1 is a computer running Mac OS using Quick time. IP2 is a server running RedHat Linux, providing SSH service. IP3 is a Windows XP host, installing Serv-U as a FTP server. IP4 is a telnet server running Windows NT. Firewall rules assure users outside the firewall have only remote access to IP1 [12].



**Figure 3.** Attack graph for experimental network [12]

Figure 3 represent an attack graph generated from experimental network in Figure 2.

#### 4. MATHEMATICAL MODEL

Noel et. al. [2] model a network security hardening with a set of exploits in terms of security conditions. They then build a directed graph of the dependencies among exploits and conditions, via exploit preconditions and post conditions. They build the dependency graph through a multi-step process, implemented through a custom Java application. They first build a dependency graph starting from an “initial conditions” exploit, i.e., an exploit with the initial network conditions as its post conditions (and null preconditions). The resulting graph represents forward dependencies from the initial conditions. More explanation about this model can be found in [2].

Wang et. al. [3] show that a network hardening solution based on exploits or intermediate conditions cannot be easily enforced. This observation motivates them to pose the following question: which of the initial conditions must be disabled, if the goal conditions are never to be satisfied? Their solution will thus be actually possible to implement, because it only includes initial conditions, which can be independently disabled. To more formally state the above problem, it is convenient to interpret an attack graph as a simple logic program as follows. Each exploit or condition in the attack graph is interpreted as a logic variable. The interdependency between exploits and conditions now becomes logic propositions involving the two connectives AND and OR, with AND between the conditions required by each exploit and OR between the exploits implying each condition.

Let there be vulnerabilities  $v_1, v_2, v_3, v_4, v_5$ , and  $v_6$  in a network net. There are the following security measures for net:  $sm_1, sm_2$ , and  $sm_3$ .  $sm_1$  removes vulnerabilities  $v_1, v_5$ , and  $v_6$ .  $sm_2$  removes vulnerability  $v_1$  and  $v_4$ , and  $sm_3$  removes  $v_1$  and  $v_3$ . If only two security measures can be chosen of the three, which two security measures should be

chosen? This issue not only exemplifies the difficulty of choosing security measures, but it underscores the importance of the security metric being used [5].

Idika et. al. [5] model the Security Measures Choosing Problem (SMCP) based on the binary Knapsack problem as follows:

The countermeasures are labeled 1...n.

$m_j$  = benefits gained by implementing countermeasures  $j$ ,

$q_j$  = costs resulting from the implementation of countermeasures  $j$ , and

$B$  = budget that is owned by the network administrator,

select a subset of countermeasures for hardening a network so as to

$$\text{maximize } \varphi = \sum_{j=1}^n m_j x_j$$

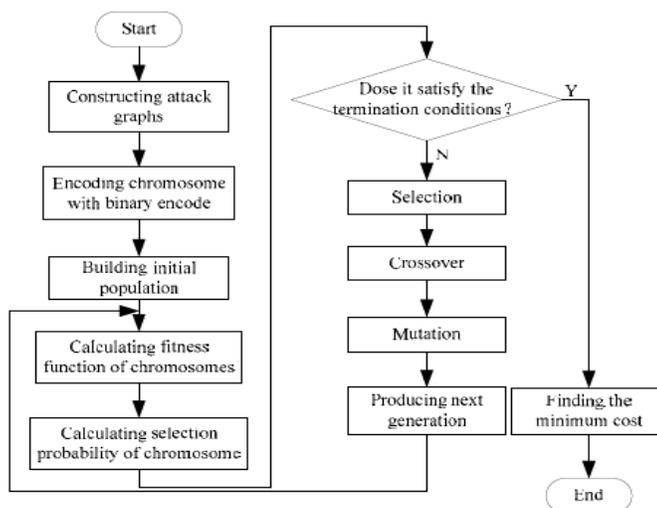
$$\text{subject to } \sum_{j=1}^n m_j x_j \leq B,$$

$$x_j = 0 \text{ or } 1,$$

$$x_j = \begin{cases} 1 & \text{if measure } j \text{ is selected;} \\ 0 & \text{otherwise.} \end{cases}$$

There are requirements associated with their model of the SMCP. First, the model implicitly assumes that there exists a security metric to derive  $m_i$ . A countermeasure  $m_i$  is calculated with respect to the security measures that are already being used. The rationale for this calculation constraint stems from the possible overlapping in the security measure's ability. For example, assume there are two security measures being considered. When either security measure is used in isolation, protection is provided to half the network. However, if both security measures are used at the same time, then it is not necessarily the case that the network is completely protected. The two security measures may overlap in what network assets they protect. Hence,  $m_i$  may take on different values depending on what security measures are already in place within the network. The model also assumes that the network administrator is able to assign costs to the hardening measures in terms of money or time [5].

Ma Junchun et. al. [7] proposed a Minimum Cost of Network Hardening Model (MCNHM) as can be seen in Figure 4.



**Figure 4.** The flow of MCNHM's algorithm [7]

The minimum cost of network hardening is a NP complete problem, so it is impossible when using exact search method, but it can use a heuristic search method. On the other hand, compared with other optimization algorithms, the genetic algorithm has the following advantages:

- a. large coverage benefiting the global merit,
- b. less dependence on the problem and stronger robustness in solution,
- c. the characteristics of parallel computing can improve the speed of calculation,
- d. especially for solving the large complex system optimization problem.

Therefore, they present a MCNHM which combines the genetic algorithm and attack graphs [7].

## 5. CONCLUSION

Network security hardening by using attack graph can be done. Some network security hardening model described in this paper is based on exploit dependency graphs, dynamic programming and genetic algorithm. The work of our future is to improve network security hardening model and make simulation studies. Optimal network hardening is NP complete problem, which is not possible when using the exact search method and the best way to prevent the exploitation is to remove the initial attribute that causes exploitation. Because attackers often achieve their goals through multiple exploits, network vulnerability analysis must consider the combined impact of the vulnerability. Exploiting sequence analysis is a good first start. But what is really needed are explicit and managed network hardening options that provide safety guarantees, as we describe here.

## References

- [1] R.P. Lippmann, K.W. Ingols et al., "Evaluating and Strengthening Enterprise Network Security Using Attack Graphs," Project Report, ESC-TR-2005-064, Lincoln Laboratory, MIT, 2005.
- [2] S. Noel, S. Jajodia, B. O'Berry, M. Jacobs, "Efficient minimum-cost network hardening via exploit dependency graphs," In Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC 2003), 2003.
- [3] L. Wang, S. Noel, S. Jajodia, "Minimum-Cost Network Hardening Using Attack Graphs," Computer Communications, 29(18), 3812-3824, 2006.
- [4] T.W. Purboyo, B. Rahardjo, Kuspriyanto, "Security Metrics: A Brief Survey," International Conference on Instrumentation, Communication, Information Technology and Biomedical Engineering (ICICI-BME) 2011, ITB Bandung, 8-9 November 2011.
- [5] N.C. Idika, B. Marshall, B. Bhargava, "Maximizing Security given a Limited Budget," Proc. TAPIA '09, Richard Tapia Celebration of Diversity in Computing, Apr. 2009.
- [6] T.W. Purboyo, B. Rahardjo, Kuspriyanto, I. Muchtadi, "A New Metrics for Predicting Network Security Level," Journal of Global Research in Computer Science (JGRCS), Volume 3, No. 3, March 2012.
- [7] M. Jun-chun, W. Yong-jun, S. Ji-yin, C. Shan, "A Minimum Cost of Network Hardening Model Based on Attack Graphs," Procedia Engineering 15, Advanced in Control Engineering and Information Science, 3227 – 3233, 2011.
- [8] T.W. Purboyo, Kuspriyanto, "New Non Path Metrics for Evaluating Network Security Based on Vulnerability," International Journal of Computer Science Issues, Volume 9, Issue 4, July 2012.
- [9] S. Noel, S. Jajodia, "Measuring Security Risk of Networks Using Attack Graphs," International Journal of Next-Generation Computing, Vol. 1, No. 1, July 2010.
- [10] T.W. Purboyo, B. Rahardjo, Kuspriyanto, I. Muchtadi, "Survey of Attack Graph-Based Security Metrics," Submitted.
- [11] Idika, N., Bhargava, B., Extending Attack Graph-Based Security Metrics and Aggregating Their Application, IEEE Transaction on Dependable and Secure Computing, Vol. 9, No. 1, January/February 2012.
- [12] Y. Zhao, Z. Wang, X. Zhang, J. Zheng, "An Improved Algorithm for Generation of Attack Graph Based on Virtual Performance Node," 2009 International Conference on Multimedia Information Networking and Security, 2009.
- [13] Irawati, T.W. Purboyo, "Developing Computer Program for Computing Eigen pairs of 2x2 Matrices and 3x3 Upper Triangular Matrices Using the Simple Algorithm," Far East Journal of Mathematical Sciences (FJMS), Volume 56, Issue 2, p. 185-200, September 2011.
- [14] T.W. Purboyo, Kuspriyanto, "Some Algorithms for Generating Attack Graph," International Journal of Advanced Research in Computer Science and Software Engineering, Volume 9, Issue 4, August 2012.
- [15] T.W. Purboyo, Kuspriyanto, "Attack Graph Based Security Metrics: State of The Art," International Journal of Science and Engineering Investigations, Volume 1, Issue 7, August 2012.
- [16] T.W. Purboyo, B. Rahardjo, Kuspriyanto, I. Muchtadi, "Optimal Network Hardening Using Attack Graph: A Survey," Submitted.

## AUTHOR



**Tito Waluyo Purboyo** is currently a Ph.D. student at Institut Teknologi Bandung since August 2010. He received his Master's degree in mathematics from Institut Teknologi Bandung in 2009. He is currently a research assistant at Department of Computer Engineering, School of Electrical Engineering and Informatics, Institut Teknologi Bandung. His research interest includes security, cryptography, physics and mathematics.



**Kuspriyanto** is Professor of Computer Engineering at Institut Teknologi Bandung. He received his D.E.A. in Automatic System (1979) from USTL France and Ph.D. in Automatic System (1981) from the same university. He is working as a lecturer in Computer Engineering Department, School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Indonesia. His field of interest includes network security, neural network, genetic algorithm, robotics, real time system etc.