OPTIMIZING AND ANALYSING THE EFFECTIVENESS OF SECURITY HARDENING MEASURES USING VARIOUS OPTIMIZATION TECHNIQUES AS WELL AS NETWORK MANAGEMENT MODELS GIVING SPECIAL EMPHASIS TO ATTACK TREE MODEL

Dr. Prabhat Kumar Vishwakarma[1]

[1]Department of Computer Science, University of Gondar, Ethiopia

pkv2005@indiatimes.com

ABSTRACT

To cope up the network security measures with the financial restrictions in the corporate world is still a challenge. At global scenario the tradeoff between the protection of IT infrastructure and the financial boundation for any organization using IT as valuable resource is quite essential. Every organization has different security needs and different budgets for coping with that therefore whether it has to look as single objective or as multiple objectives with fault tolerant feature is a critical issue. In the present paper an attempt has been taken to optimize and analyze the effectiveness of security hardening measures considering attack tree model as base. In short we can say that the main attention in the paper is-to rectify, to describe the notations of the attack tree model and to suggest a model which may be able to quantitatively specify the possible threats as well as cost of the security control while implementing the security hardening measures.

KEYWORDS

Security Management, Attack Tree, Objective Functions, Network Security, NSGAII.

1.0 INTRODUCTION

Network based information technology infrastructure have now become the necessity of any organization in the era of globalization for the strategic management and for achieving the competitive advantages .So it is essential for business organizations to be optimally secure and safe from both internal and external attacks. The network administration and management of the organization has the real challenge to protect and ensure the effective utilization of IT Infrastructure within the affordable budget of the organization. Thus cost effective security management should be implemented in such a way so that possible causes for the damage of the secured assets may be identified and optimal set of policy rules may be framed to defend against such losses. Even though various security based networked models have been suggested based on the idea of attack graph[1,11,15,18,20] as well as attack tree[6,13,16,17]but sorry to say that these have not been so much effective over financial restrictions. Therefore for managing the proper trade-off between security services and cost control it has been felt the need of optimal usability of the set of security hardening measures.

DOI : 10.5121/ijnsa.2011.3409

2.0 ANALYZING NETWORK VULNERABILITY AS SINGLE-OBJECTIVE OPTIMIZATION PROBLEM

The computation of minimum cost hardening measure is possible using exploit dependency graph [14] but it can be easily bypass by opting other attack path. The other method [11] can be used to find optimal collection of security attack as well as their possible security measures. This method may be useful for complete network protection but not feasible under financial and other business constraints. Therefore it is sure that these approaches are fit to treat it as single-objective optimization problem but not as multi-objective optimization problem.

3.0 FORMULATION OF MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

A multi-objective formulation of the problem [10] considers a generic set of security policies capable of covering one or more generic vulnerabilities. A security policy can also introduce possible vulnerabilities, thereby resulting in some residual vulnerability even after the application of security policies. Therefore the multi-objective problem may be thought of –Minimizing the implementation cost along with the residual weighted cost imposed by applying the security hardening measure. Thus most multi-objective algorithms use the concept of dominance (definition-1) to compare feasible solutions.

Dominance and Pareto-optimal set

In a minimization problem with M objectives, a feasible solution vector x is said to dominate another feasible solution vector y if

1. $\forall i \in \{1, 2, . . . , M\} fi(x) \leq fi(y)$ and
2. $\exists j \in \{1, 2, . . ., M\}$ $fj (x) < fj(y)$

If the above mentioned conditions do not hold, then x and y are said to be non-dominated w.r.t. each other.

Definition-1

The dominance rule is well suited for the multi-objective optimization problem because the solution obtained in this way has the characteristics that the solution reducing one objective function will increase the other objective function. This is what we need to maintain the tradeoff between security attack and implementation cost. Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [8] for the multi-objective optimization are used frequently because of its efficiency in terms of the convergence and diversity of solutions obtained.

3.0 TO SET UP A SIMPLE EXAMPLE NETWORK MODEL

For understanding the vulnerabilities and attack scenarios, consider for example a network setup with four hosts consisting of two services-FTP, SSH for the external users with firewall having policy to allow external user only to communicate with SMTP and FTP Servers. Suppose an attacker is trying to attack such a network for accessing the data server located inside the firewall, diagrammatically the situation is shown as below (figure-1)-

The possible vulnerability (table-1) host wise is tabulated as below-

Host	Vulnerability
FTP SERVER-215.15.0.1	FTP .rhost attack,FTP Buffer Overflow,SSH Buffer Overflow
SMTP SERVER-215.15.0.4	FTP .rhost attack
DATA SERVER-215.15.0.2	LICQ remote-2-user suid Buffer Overflow
TERMINAL-215.15.0.3	LICQ remote-2-user "at" heap corruption

Table-1

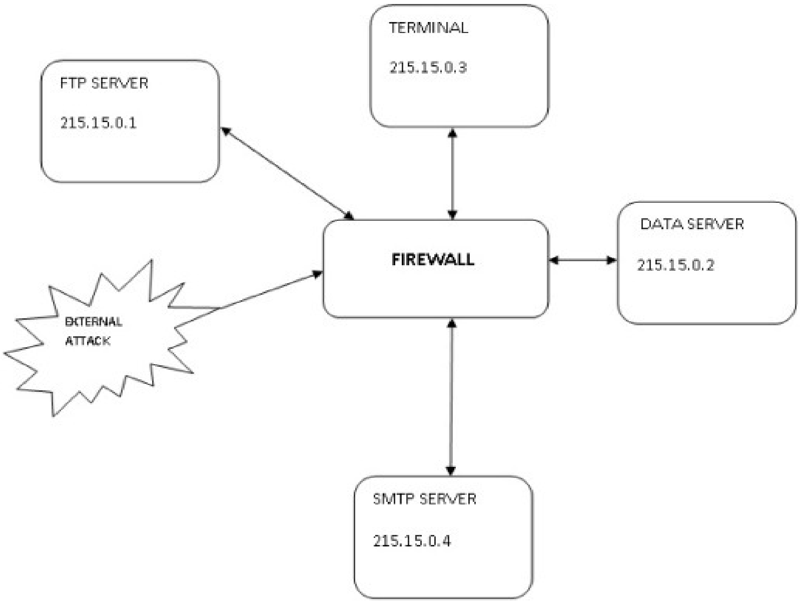

Figure-1

4.0 REPRESENTATION OF ATTACK USING ATTACK TREE MODEL

To determine the minimal set of preventive action there is a need of attack tree model. For reducing the visualization complexity the attack tree uses a technique known as conjunctive and disjunctive branch decomposition. The representation also helps to calculate the cost factors. For exactly understanding the attack tree model some more terms are required to be defined-

Attribute-Template

An attribute-template is a generic property of the hardware or software configuration of a network which includes, but not limited to, the following:

- System vulnerabilities
- Network configuration.
- System configuration • Access privilege such as user account, guest account, or root account.
- Connectivity.

Definition-2

The atomic properties incurred from the definition of attribute template may provide some clue to the attackers. Such templates specify the properties in propositional logic.

Attribute

An attribute is a propositional instance of an attribute-template. It can have the truth values either true or false

Definition-3

Since the discussion is based on the propositional logic, therefore the truth values of attributes is the deciding factor of the success or unsuccessful of the attacker's goal. It also provides the basis for analyzing the possible threats to the members involved in the security analysis.

An attack (definition-4) relates the truth values of two different attributes to embed a cause-consequence relationship between the two.Mathematically an attack tree (definition-5) can be defined as-

Attack

Let S be a set of attributes. We define Att to be a mapping
Att : $S \times S \rightarrow \{true, false\}$ and Att(sc, sp) = truth value of sp.
$a = Att(sc, sp)$ is an attack if $sc \neq sp \wedge a \equiv sc \leftrightarrow sp$.
sc and sp are then respectively called a precondition and postcondition of the attack, denoted by pre(a) and post(a) respectively.

Definition-4

Attack Tree

Let A be the set of attacks, including the φ–attacks. An attack tree is a tuple AT = (s_{root}, S, τ, ε), where

1. s_{root} is an attribute which the attacker wants to become true.
2. S = $N_{internal} \cup N_{external} \cup \{s_{root}\}$ is a multiset of attributes. $N_{external}$ denotes the multiset of attributes s_i for which $\exists a \in A | s_i \in post(a)$. $N_{interna}l$ denotes the multiset of attributes s_j for which $\exists a1, a2 \in A | [s_j \in pre(a1) \wedge s_j \in post(a2)]$.
3. $\tau \subseteq S \times S$. An ordered pair (spre, spost) $\in \tau$ if $\exists a \in A | [spre \in pre(a) \wedge spost \in post(a)]$. Further, if si $\in$ S and has multiplicity n, then $\exists s1, s2. . . sn \in S | (si, s1)$, (si,s2), . . . , (si, sn) $\in \tau$, and
4. ε is a set of decomposition tuples of the form (s_j, d_j)defined for all $s_j \in N_{internal} \cup \{s_{root}\}$ and $d_j \in$ {AND,OR}.

d_j is AND when $\bigwedge_i[si \wedge (si,sj) \in \tau] \leftrightarrow sj$ is true, and OR when $\bigvee_i[si \wedge (si,sj) \in \tau] \leftrightarrow sj$ is true.

Definition-5

For the considered simple network scenario the attack tree formed will look like as (figure-2)-

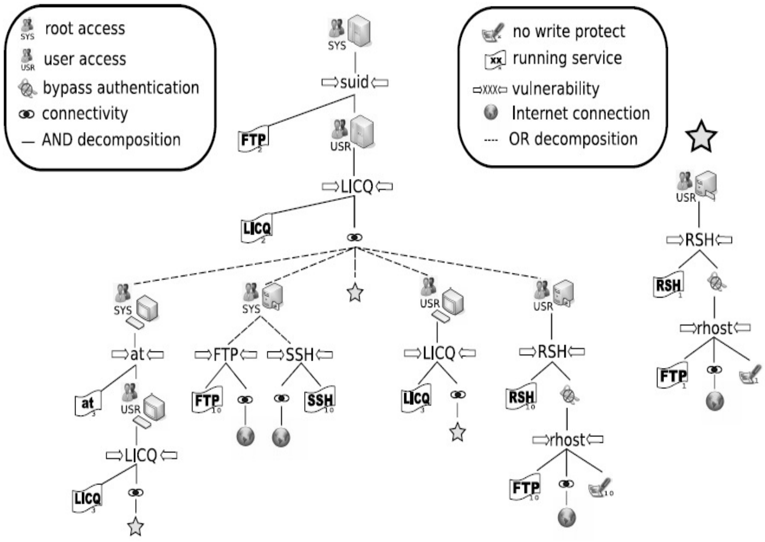

Figure-2

5.0 SECURITY PLANNING AND COST MODELING

Security planning begins with risk assessment which determines threats, loss expectancy, potential safeguards and installation costs. The hardening cost and magnitude of loss can be useful for evaluating the risk[2,12,19].In case of resource-constraints, for any organization the relative cost approach is not very much useful for security measures. Butler's multi-attribute risk assessment framework [3, 4] to develop quantitative risk assessments for security optimization enables an aggregated representation of the various factors dominating the business model of an organization. The security control (definition-6) can be mathematically defined as-

Security Control
Given an attack tree (s_{root}, S, τ, ε), the mapping SC: Nexternal → {true, false} is a security control if $\exists si \in Nexternal | SC(si) = false$.

Definition-6

Here the security control is of preventive nature and it sets some of the truth values of the attributes in such a way that the attackers cannot be successful in their goal.Further, in the presence of multiple security controls *SCk*, the truth value of an attribute $si \in N_{external}$ is taken as $\bigwedge_k SCk(si)$. Given a security control *SC*, the set of all $s_i \in N_{external} | SC(s_i) = false$ is called the *coverage* of *SC*. Hence, for a given set of security controls it can be defined the *coverage matrix* specifying the coverage of each control. For a given set of *m* security controls, one can use the boolean vector $\overrightarrow{T} = (T1, T2, \ldots, Tm)$ It can be useful to check whether the security control has been selected by the security controller or not, it is also the indirect indication about the attributes behavior in the attack tree.

6.1 SPECIFYING THE POSSIBLE LOSSES

According to Butler's multi-attribute risk assessment framework [3, 4] the evaluation of possible damage can be specified by following the steps depicted as below-

STEP1: To identify the outcomes due to the truth values imposed by the attacker into the associated attributes.In the case considered the outcomes are-low penalty, revenue losses, unused downtime, damage recovery and public embarrassment. Let us consider it as-x_{1j}, x_{2j}, x_{3j}, x_{4j} and x_{5j}.

STEP2: Estimate the expected number of attack occurrence Freqj, resulting in the consequences.

STEP3: for every outcome, to compute the function Vij(Xij) as-

$$V_{ij}(x_{ij}) = \frac{x_{ij}}{\underset{j}{Max}\, x_{ij}} \times 100 \qquad ,1 \leq i \leq 5$$

STEP4: For every outcome assign a weight factor say Wi.

The potential damage for the attribute can then be calculated from the following equation-

$$P_j = Freq_j \times \sum_{i=1}^{5} W_i V_{ij}(x_{ij})$$

In attack tree modeling, the cost can be quantitatively represented using the residual damage after the implementation of the security policy. Therefore augmentation (definition-7) for every attribute in the attack tree is required for specifying the possible damage in the tree.

Augmented attack tree

Let AT = (sroot, S, τ, ε) be an attack tree. An augmentedattack tree ATaug = AT|_I,V _ is obtained by associating a tuple (Ii, Vi) to each si ∈S, where

1. Ii is an indicator variable for the attribute si, where
Ii =(0 , if si is false 1 , if si is true
2. Vi is a value associated with the attribute si.

Definition-7

In this proposed work, all attributes $s_i \in N_{external}$ are given a zero value. The value associated with $s_j \in N_{internal} \cup \{sroot\}$ is then computed recursively as follows-

$$V_j = \begin{cases} \sum\limits_{k|(s_k,s_j)\in\tau} V_k + I_j P_j & , if\ d_j\ is\ AND \\ \underset{k|(s_k,s_j)\in\tau}{Max}\, V_k + I_j P_j & , if\ d_j\ is\ OR \end{cases}$$

Therefore using the following definition (definition-8) the residual damage can be evaluated.

Residual damage

Given an augmented-attack tree (s_{root}, S, τ, ε)|(I,V) and a vector T = (T_i), T_i ∈{0, 1}; 1 ≤ i ≤ m, the residual damage is defined as the value associated with s_{root}, i.e.,

RD(T) = Vroot

Definition-8

6.2 SPECIFYING THE SECURITY CONTROL COST

Similar to the potential damage, the security costs are enlisted for the possible implementation of a security control, assigns the weight factor on them, and computes the normalized value. The only difference is that there is no expected number of occurrences needed in the evaluation of security cost. In this study, the identified different costs of implementing a security control are installation cost (monetary), operation cost (monetary), system downtime (time), incompatibility cost (scale), and training cost (monetary). The overall cost C_j, for the security control SC_j, is then computed in a similar manner as for potential damage, with an expected frequency of 1. The total security cost (definition-9) for a set of security controls implemented is then defined as follows-

Total Security Control Cost

Given a set of m security controls, each having a cost Ci; $1 \leq i \leq$ m, and a vector T = (Ti), Ti $\in \{0, 1\}$; $1 \leq i \leq m$, the total security control cost is defined as

$$SCC(\vec{T}) = \sum_{i=1}^{m}(T_iC_i)$$

Definition-9

7.0 FORMULATING AND ANALYZING THE NETWORK SECURITY PROBLEM USING NSGA II

Mathematically the above discussed concepts can be represented in the form of multiple objective problems as follows-

For a given augmented-attack tree (s_{root},S, τ, ε)|(I,V) with m number of security controls, our objective is to find a vector T= (Ti), Ti$\in$ {0, 1}; with condition1 $\leq$i $\leq$ m, which will be required to minimize the total security control cost as well as the residual damage, which satisfies the constraint Max RD(T_r) – RD(T) $\leq$ D where, D is the maximum perturbation permissible in the residual damage. Let us consider T = (Ti) as Boolean vector then for the purpose of perturbed assignment of radius r, Tr, can be obtained just by inverting the value of at most r elements of the vector T. NSGA-II is initialized with a population *P*0 out of *N* randomly generated security control vectors *T*. then after for finding each trial based solution,we need to calculate the total security control cost .Just by assigning false to the set of security control,the attributes can be decided which is covered by security control vector into the attack tree and this can be helpful in computing the residual damage. The remaining attributes in *Nexternal* are initialized with truth value *true*. For specyfing the truth values of the internal nodes we need to use DFS traversal into the attack tree which will be required to compute the residual damage ie Vroot.For recording the number of iterations in NSGGA II we need generation index as *t* = 0, 1, . . . , *GenMAX*. Every generation of NSGA II will be executed as follow-

The genetic operations viz. mutation, crossover as well as selection will be required for creating the offspring population say Qt from the parent population Pt. For the solution of every offspring population the residual damage and total control cost are also computed. By rank X solution it means that there exists X different solutions of different rank which dominates it. For finding it we need to apply non-dominated sorting on the population say Rt, which can be obtained by joining the parent as well as offspring populationThe infeasible solutions are ranked higher than the highest feasible solution. The NSGA II uses a special mechanism known as diversity-preservation mechanism. This mechanism is based on a metric known as crowding distance metric. This mechanism is helpful for finding the preferable solution having lesser density of solution around it.

8.0 CONCLUSION

This paper discusses the various aspects of security hardening measures keeping in view the several organizational constraints, mainly the implementation cost and to devise optimal set of security policy to make the IT infrastructure robust. If it is required to protect the damage of IT resources from both internal and external attacks then the network vulnerability can be viewed as single-objective optimization problem. But if both protection of IT infrastructure as well as financial constraints of the organization are in consideration then it is mandatory to formulate the security problem as multiple objective optimization problems. .By representing the network model into attack tree model the visualization complexity of all the possible set of security controls can be reduced and subsequently it may be helpful to optimize the residual damage along with optimizing the possible set of security policies in a cost effective way.

9.0 REFERENCES

[1] Ammann, P., Wijesekera, D., and Kaushik, S. Scalable, Graph-Based Network Vulnerability Analysis. In *Proceedings of the Ninth Conference on Computer and Communications Security* (Washington, DC, USA, 2002), pp. 217–224.

[2] Berger, B. Data-centric Quantitative Computer Security Risk Assessment. *Information Security Reading Room, SANS* (2003).

[3] Butler, S. Security Attribute Evaluation Method: A Cost-benefit Approach. In *ICSE 2002: Proceedings of the 24rd International Conference on Software Engineering* (Orlando, FL, USA, 2002), pp. 232–240.

[4] Butler, S., and Fischbeck, P. Multi-attribute Risk Assessment. In *Proceedings of SREIS02 in conjunction of 10th IEEE International Requirements Engineering Conference* (Raleigh, NC, USA, 2002).

[5] Coello, C. A. C. An Updated Survey of GA-based Multiobjective Optimization Techniques. *ACM Computing Surveys 32*, 2 (2000), 109–143.

[6] Dawkins, J., Campbell, C., and Hale, J. Modeling Network Attacks: Extending the Attack Tree Paradigm. In *Proceedings of the Workshop on Statistical Machine Learning Techniques in Computer Intrusion Detection* (Baltimore, MD, USA, 2002), Johns Hopkins University.

[7] Deb, K. *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons Inc., 2001.

[8] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation 6*, 2 (2002), 182–197.

[9] Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[10] Gupta, M., Rees, J., Chaturvedi, A., and Chi, J. Matching Information Security Vulnerabilities to Organizational Security Policies: A Genetic Algorithm Approach. *Decision Support Systems 41*, 3 (2006), 592–603.

[11] Jha, S., Sheyner, O., and Wing, J. M. Two Formal Analysis of Attack Graphs. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop* (Cape Breton, Nova Scotia, Canada, 2002), pp. 49–63.

[12] Lee, W. Toward Cost-sensitive Modeling for Intrusion Detection and Response. *Journal of Computer Security 10*, 1 (2002), 5–22.

[13] Moore, A., Ellison, R., and Linger, R. Attack Modeling for Information Survivability. Technical Note CMU/SEI-2001-TN-001, Carnegie Melon University / Software Engineering Institute, March 2001.

[14] Noel, S., Jajodia, S., O'Berry, B., and Jacobs, M. Efficient Minimum-cost Network Hardening via Exploit Dependency Graphs. In *Proceedings of the 19th Annual Computer Security Applications Conference* (Las Vegas, NV, USA, 2003), pp. 86–95.

[15] Phillips, C., and Swiler, L. A Graph-Based System for Network-Vulnerability Analysis. In *Proceedings of the 1998 New Security Paradigms Workshop* (Chicago, IL, USA, 1998), pp. 71–79.

[16] Ray, I., and Poolsappasit, N. Using Attack Trees to Identify Malicious Attacks from Authorized Insiders. In *ESORICS 2005* (Milan, Italy, 2005), pp. 231–246.

[17] Schneier, B. Attack Trees. Dr. Dobb's Journal (1999).

[18] Sheyner, O., Haines, J., Jha, S., Lippmann, R., and Wing, J. M. Automated Generation and Analysis of Attack Graphs. In SP 2002: Proceedings of the IEEE Symposium on Security and Privacy (Oakland, CA, USA, 2002), pp. 273–284.

[19] Stoneburner, G., Goguen, A., and Feringa, A. Risk Management Guide for Information Technology Systems. NIST Special Publication 800–30 (2002).

[20] Swiler, L., Phillips, C., Ellis, D., and Chakerian, S. Computer-Attack Graph Generation Tool. In Proceedings of the DARPA Information Survivability Conference and Exposition II (Anaheim,CA, USA, 2001), pp. 307–321.

Author

Dr.Prabhat Kumar Vishwakarma is currently working as Assistant Professor, in Computer Science Department, University of Gondar, Ethiopia. He completed his Doctorate in Computer Science discipline in 2008 from the State University MGKVP, Varanasi; India. He has approximately 12 years of academic exposure both at national as well as international level.