Patching Assignment Optimization for Security Vulnerabilities

Shao-Ming Tong, Chien-Cheng Huang, Feng-Yu Lin, and Yeali Sun
Department of Information Management, National Taiwan University, Taiwan

Abstract: *This research is focusing on how IT support center applies the limited resources to elaborate a vulnerability patch in face of its disclosure in a system. We propose the most optimized procedure to design the patch in question and let second-tier security engineer handle the update for vulnerabilities with patch release. While the frontline security engineer are able to provide a firewall to hold the leakage plus create and update the patch in the shortest amount of time. In face of, some system vulnerabilities, the frontline security engineer has to build up a prevention procedure before the patch is released. The strategy of this study is to focus on the transfer of patch demand to the adequate system engineer in a mathematical programming problem module. Within it the objective function is minimized to pursue the shortest amount of survival time for the vulnerability (before the patch is released), we also added some related constraints. The main contributions of this study is a non-linear non-convex mixed integer programming problem formulation for patching assignment optimization and a near optimal solution approach.*

Keywords: *Vulnerability, patch management, assignment algorithm, optimization, mathematical programming, near optimal solution*

Received September 4, 2013; accepted June 29, 2014; published online April 1, 2015

1. Introduction

The information security issue has become increasingly important for every system controller, because most of the attacks made through internet are targeting system weaknesses. Therefore, one of the most important tasks of system engineer is to provide a vulnerability patch. However, in most of the cases, the engineer in question cannot solve all the issues immediately after the disclosure of the vulnerabilities [17]. As an example: In a semiconductor manufacturing company, there are eight Fabs, all in different locations, in total more than 50,000 (fifty thousand) IT devices. Such a huge environment makes the process slower, opening a window for the attacks, when the patch is not yet operational or updated.

Due to this reason, this study would like to propose an equation providing these companies with the most optimized solution when using their limited resources, so that they are applied efficiently [19, 21].

The main reason to search for the adequate system engineer in face of each system issue is that we can first prioritize the needs of the company: By categorizing the vulnerabilities by status, device, usage frequency of the engineer etc., we can allow the second-tier problems being handled by the appropriate team in the time needed while the first priority issues will be dealt with immediately by first-tier engineer. Through this organization the survival time of the system vulnerabilities can be lessen to the minimum, therefore leading us to the optimization of system vulnerability patch.

The key of this study is the admission control policy of the system engineer and emergency backup devices.

The goal of admission control is to ensure that new issues to face won't affect the loading of the system engineer from the IT support center. Therefore, when a new patch is needed, the admission control must made an analysis based on the available resources and engineer team loading to carry out a trade-off decision whether or not to face the new issue immediately. As a result, the design of system vulnerabilities level, admission control threshold and definition of system engineer maximum loading will be the key points of this study.

The summary of this paper is organized as follows: In section 2, we discuss related work. In section 3, we develop our research methodology. We propose a patching assignment optimization for security vulnerabilities and a near optimal solution approach in sections 4 and 5. Section 6 contains computational experiments, finally, the research will be concluded in section 7.

2. Related Works

In recent years, the frequency of information security incidents increased dramatically, causing a strong impact on enterprises operations efficiency, if severe even resulting in important loss or damage. In most of the cases, information systems are attacked through misconfigurations and non-patch of vulnerabilities [1, 5]. Various tactics deployed by the hackers have a very bad impact on information security and economics [2, 14, 18]. These attacks are not only creating security

incidents or damage to the informative system, but also network paralysis. These vulnerabilities might be in the operation system, the application software or the weakness on the computer server hardware. Being the reasons to system getting: Accessed, modified and shut-down out of the security policy in organizations [17].

Arbaugh *et al.* [4] presented a vulnerability life-cycle model. In general, the vulnerability timeline is starting from the discovery of software vulnerability until the patch is deployed. When software vendors release the patch in response to the weakness of the system, the user can then apply it to solve the issue. [3, 7, 13, 20]. Consequently, Brykczynski and Small [8] reported practices of security patch management and also, emphasized an importance of economic security patch management as a part of information asset management.

In actual research related to optimal patching policies, Ioannidis *et al.* [15] proposed a mathematical model of trade-off process between confidentiality and availability when planning to release the patch, so that we can calculate the optimal frequencies of regular and irregular patching. On the other hand, in studies about patch management, Arora *et al.* [6] also proposed an optimal policy for software vulnerability disclosure. They analyzed the optimal timing of disclosure, proving that the lower the patch release frequency of system vendors, the better the effect on society. Beside this, Cavusoglu *et al.* [9] discussed the effect of vendor's patch release. They calculated the optimal window of exposure and decomposed the patching process into time-and event-driven incidents then analyzed the patch release and management models from both vendor and user perspectives. However, Okamura *et al.* [22] proposed an extension of Cavusoglu *et al.* [9] patch management model with non-homogeneous vulnerability-discovery processes to find the optimal security patch release times. Besides, in researches about risk assessment, Karabey and Baykal [16] proposed an attack tree based information security risk assessment method integrating enterprise objectives with vulnerabilities.

However, the enterprises usually don't have the time to install or deploy the system vulnerability patch, facing the problem that system can be attacked before the patch is released, plus the fact that update failure is also a frequent situation [17]. Therefore, we propose in this study a solution to the process optimization.

3. Research Methodology

To fulfill the maximum optimization of system vulnerability patch, in this study, we focus on how the IT support center can carry on the repartition of the work to the adequate engineer within the limited available resources.

The key is how to find the appropriate engineers in face of various system issues, this problem cannot be solved within a polynomial time, on the other hand, it belongs to the category of NP-complete problem. Even if a lot of possible approaches are present when dealing with such kind of issue, according to Fisher [10, 11], we can rely on the Lagrangian Relaxation (LR) to obtain the near optimal solution and replace the linear programming, because the lower bound calculated from LR is closer to optimal than the lower bound from linear programming. As a result, we can apply LR to solve the issue that polynomial time cannot solve.

LR is a way to calculate the near optimal solution, by relaxing the complicated constraints, it simplifies the primal problem. If we seek the minimum of the objective function, we can choose to transit the most complicated constraint and multiply it by a lagrangian multiplier and add it back into the original objective function. If the constraint is an unequal equation, the Lagrangian multiplier cannot be of minus value, after this we use the subgradient method to find out the Lagrangian multiplier and through repeated calculation retrieve the closest point toward optimization. At this moment, the difference between the most optimized point calculated from LR and from the original problem is the smallest. Due to the fact that in a NP-complete problem it is impossible to deduce an applicable best solution, following the same logic, by LR we can only calculate the closest result to the best solution.

The methodology of this research is to analyze the overall environment of system vulnerability repartition within the engineer team and its process, converting all these into a mathematical programming problem, within it, the objective function represents the minimum survival time of the system weakness, plus the related constraints, taking into consideration engineer working hours, IT support center devices delivery limitations, existing vulnerabilities quantity etc. In the following part, we will develop an optimization-based algorithm to solve this complicated issue.

4. Patching Assignment Optimization

The subject of this study being "control of allowing existing vulnerabilities transfer to the adequate engineer", it means that under the presupposition of fixed devices vulnerabilities, position of backup protocols and different levels of information security engineers, we allow that the vulnerabilities with less threat can be transferred to second-tier security engineer for system update, so that the available resources are utilized on high priority issues and therefore increase the efficiency of emergency

solution. In this study we discuss the patch update process of high level threat vulnerabilities, so that their survival time is brought to minimum size and the solution being optimized.

We plan to turn the above problem into an equation and then into an optimized mathematical model. It will be a non-linear non-convex mixed integer programming problem. Under the limit of constraints, we are looking for an objective function to ensure that each vulnerability survival time before the update patch arrival is minimized, but at the same time we need to set up some parameters through various constraints. When building them, we included into equation the loading capacity of backup devices from the IT support center, the effect of changing devices on security engineers and the impact of working hours change.

In this study the main three decision variables are the accessibility of vulnerable devices of the IT support center, the accessibility of informative engineers and the accessibility of their devices. These combined to the procedure of vulnerability patch update must be taken into consideration at the same time so that we ensure that the new total will not exceed the capacity of the system or the engineer. Moreover, when considering the transfer of less important issues to second-tier engineer, will it have a severe impact on their devices or personnel? As a result, the main task of this study is to develop a control protocol which will minimize the survival time of system vulnerabilities when designing the update patch for system issues.

Within the three main factors taken into consideration for this study, the change of working hours from system engineer is the core of the equation. When taking first the vulnerabilities quantity into account, the factors influencing the working hours of engineer include not only the actual existing vulnerabilities but also the new incoming ones, the ones transferable to second-tier engineer plus the ones being transferred from other engineers. Beside the pure amount of the issues, we must also add the status of threatened devices, the capacity of the IT support center to face such problems, etc. All these factors assembled together will also lead us to the fact that during the transfer of cases, the accessibility of different engineers added to their working hours may also have an impact on the patch update efficiency.

After deciding the admission control protocol of the security engineers, we must then considerate the admission control of the IT support center, to be brief: The center will allocate different cases according to precise job repartition by devices status and engineer capacity. According to different issues, the center in question will accord the mission to the engineer capable of providing the minimum vulnerability survival time and IT support center waiting time (e.g., Erlang-C formula), so that the maximum efficiency is attained.

Table 1 is a verbal description of the patch assignment system design problem we considered. Tables 2 and 3 show the notation of given parameters and notation of decision variables we use in this study.

Table 1. Problem description.

Given
• The risk level of the vulnerabilities from the National Vulnerability Database (NVD).
• The disclosure time of the vulnerabilities.
• The patch release time of the vulnerabilities.
• The shortest update time of the released patch.
• The total quantity of backup devices.
• The actual vulnerabilities on-task of the engineers from different levels.
• The repair range and backup devices quantity of the IT support center.
• The appearance rate of vulnerabilities.
• The transfer rate of vulnerabilities to different engineers.
Objective
• Minimize the total update process time of vulnerabilities patch.
Subject to
• The working hours limit of informative engineers.
• The limit of the backup devices provided by the IT support center.
• The limit of repair range and update process time of different patches.
Assumption
• One individual vulnerability is handled by one individual IT support center.
• One individual vulnerability is handled by one individual security engineer.
• One individual vulnerability is transferred to one individual security engineer.
• The appearance rate of vulnerability for each device.
• The transfer rate of vulnerability for each device.
• The different levels of system engineers and IT support center waiting time.

Table 2. Notation of given parameters.

Notation	Description
A	All the available security engineers within the repair range.
B	All the backup devices within the repair range.
P	The total of devices.
δ_a	The total quantity of vulnerabilities actually in the hand of the engineers.
S_p	The average survival time of the device vulnerability p.
S_a	The average survival time of the vulnerability when transferring from engineer a.
M_a	The maximum handled vulnerabilities of engineer a.
V_b	The maximum quantity of backup devise from IT support center b.
$C(M_a, g_a)$	The average waiting time of security engineer a.
$D(V_b, h_b)$	The average waiting time of IT support center b.
R_b	The repair range of IT support center b.
γ_{ab}	If the security engineer a is within the responsibility range of IT support center b then it is 1, otherwise it is 0.
β_{pb}	If the device vulnerability p is within the responsibility range of IT support center b then it is 1, otherwise it is 0.
$E_{aa'}$	The shortest time of patch update when the device vulnerability is transferred from security engineer a to engineer a'.
F_{pa}	The shortest time of patch update by security engineer a on device vulnerability p
λ_p	The vulnerability appearance rate of device p.
α_a	The appearance rate of device vulnerability transfer from security engineer a to engineer a'.

Table 3. Notation of decision variables.

Notation	Description
z_{pa}	If the accessible device vulnerability p is patched by the engineer a, then it is 1, otherwise it is 0.
x_{pb}	If the IT support center b can handle the patch of the vulnerability of device p than it is 1, otherwise it is 0.
$y_{aa'}$	If the device vulnerability can be transferred from engineer a to a' for patch update, than it is 1, otherwise it is 0.
h_b	The average rate of patch update toward device vulnerabilities of IT support center b (e.g., Erlang).
g_a	The average of patch update toward device vulnerabilities of security engineer a (e.g., Erlang).

The main purpose of this study is to pursue the minimum survival time of vulnerabilities before their adequate patch is updated. Therefore, after analyzing all the influencing factors we realize that the main targets are: The waiting time of informative engineer $C(M_a, g_a)$、the waiting time of IT support center $D(V_b, h_b)$ and the patch update time of the engineer F_{pa}. We

use them as the additions of the average survival time of device vulnerabilities S_p to calculate the survival time of vulnerabilities before its patch is deployed.

Moreover, we must also take into consideration the fact that the transfer time of cases between engineers may have an effect on the vulnerabilities survival time. So, we use the minimal time of transfer between engineer and the IT support center waiting time $D(V_b, h_b)$ as an addition to the average survival time of vulnerability S_a, to calculate the survival time of vulnerability after adding the time needed when transferring the case to the adequate engineer. In the end, we add the survival time durations to calculate the objective function (IP1) of the minimum survival time of the vulnerabilities before their patch is updated.

Regarding the part of the constraints, In Constraint 1: The engineer will not exceed its working capacity. Constraint 2 points out that the situation in which the engineer a can accept to handle the vulnerability p, must fulfill the condition that the engineer a is within the responsibility range of the IT support center b and that the IT support center b can handle the patch update of the device vulnerability p. Constraint 3 highlights that the maximum quantity of device vulnerabilities p that IT support center b can take in charge cannot exceed the average device vulnerability patch update rate. Constraint 4 shows that each vulnerabilities can only be solved by an individual engineer. Constraint 5 is used to control the relation between the two variables x_{pb} and β_{pb} within constraint 3, if device vulnerability p is within the responsibility range of IT support center b, then β_{pb}=1 and x_{pb}=1. Constraint 6 is to decide if the device vulnerability needs to be transferred to another security engineer for patch update. Constraint 7 states that device vulnerability p can only be handled by and individual IT support center b. Constraints 8, 9, 10) values are either 1 or 0.

After the entire process of converting the data into a mathematical model, we retrieve the below objective function and constraint.

- Objective function (IP1):

$$Z_{IP1}=min \sum_{p\in P}(S_p+\sum_{b\in B} x_{pb}D(V_b,h_b)+\sum_{a\in A} z_{pa}(F_{pa}+C(M_a,g_a))) + \sum_{a\in A}(S_a+\sum_{a'\in A} y_{aa'}(E_{aa'}+C(M_{a'},g_{a'}))) \quad (1)$$

Subject to:

$$\delta_a+\sum_{p\in P}\lambda_p z_{pa}-\sum_{a'\in A-\{a\}}\alpha_a y_{aa'}+\sum_{a'\in A-\{a\}}\alpha_{a'} y_{a'a}\le g_a \quad (2)$$

Where $\forall a\in A$

$$z_{pa}\le \gamma_{ab} x_{pb} \quad (3)$$

Where $\forall p\in P, b\in B, a\in A$

$$\sum_{p\in P}\lambda_p x_{pb}\le h_b \quad (4)$$

Where $\forall b\in B$

$$\sum_{a\in A} z_{pa}=1 \quad (5)$$

Where $\forall p\in P$

$$x_{pb}\le \beta_{pb} \quad (6)$$

Where $\forall p\in P, b\in B$

$$\sum_{a'\in A} y_{aa'}\le 1 \quad (7)$$

Where $\forall a\in A$

$$\sum_{b\in B} x_{pb}=1 \quad (8)$$

Where $\forall p\in P$

$$z_{pa}=0 \text{ or } 1 \quad (9)$$

$$y_{aa'}=0 \text{ or } 1 \quad (10)$$

Where $\forall a, a'\in A$

$$x_{pb}=0 \text{ or } 1 \quad (11)$$

Where $\forall p\in P, b\in B$

5. Solution Approach

Our problem within this study is of minimization nature, suitable for the LR approach, but we can observe from the objective function Z_{IP1} that: S_p and S_a are all with definite value, we only need to select the plus function of S_p and S_a within Z_{IP1} and calculate the total, which will also allow us to reach the minimal objective function, therefore we can note the from the reformulation Z_{IP2}.

- Objective function (IP2):

$$Z_{IP2}=\min\sum_{p\in P}\left(\begin{array}{l}\sum_{b\in B} x_{pb}D(V_b,h_b)\\ +\sum_{a\in A} z_{pa}(F_{pa}+C(M_a,g_a))\end{array}\right) + \sum_{a\in A}\sum_{a'\in A} y_{aa'}(E_{aa'}+C(M_{a'},g_{a'})) \quad (12)$$

The LR approach can be used to calculate a near optimal solution by "relaxing" complicated constraints it simplifies the primal problem, so according to the solving process of LR, we first turn Z_{IP2} into a LR problem model. They are then multiplied by Lagrangian multipliers and added to the objective function. This process is referred to as "dualizing" the complicating constraints. We choose to dualize Constraints (1, 2, 3, 4) and construct the following LR problem.

- Objective function (LR):

$$
\begin{aligned}
Z_{LR}(\mu_a^1,\mu_{pba}^2,\mu_{pb}^3,\mu_p^4) = min &+ \sum_{p\in P}\sum_{b\in B} x_{pb} D(V_b,h_b) \\
&+ \sum_{p\in P}\sum_{a\in A} z_{pa} F_{pa} + \sum_{p\in P}\sum_{a\in A} z_{pa} C(M_a,g_a) \\
&+ \sum_{a\in A}\sum_{a'\in A} y_{aa'} E_{aa'} + \sum_{a\in A}\sum_{a'\in A} y_{aa'} C(M_{a'},g_{a'}) \\
&+ \sum_{a\in A} \mu_a^1 (\delta_a + \sum_{p\in P} \lambda_p z_{pa} - \sum_{a'\in A-\{a\}} \alpha_{a'} y_{aa'} \\
&+ \sum_{a'\in A-\{a\}} \alpha_{a'} y_{a'a} - g_a) \\
&+ \sum_{p\in P}\sum_{b\in B}\sum_{a\in A} \mu_{pba}^2 (z_{pa} - \gamma_{ab} x_{pb}) \\
&+ \sum_{p\in P}\sum_{b\in B} \mu_{pb}^3 (\sum_{p'\in P} \lambda_{p'} x_{p'b} - h_b) \\
&+ \sum_{p\in P} \mu_p^4 (\sum_{a\in A} z_{pa} - 1)
\end{aligned}
$$

The $\mu_a^1, \mu_{pba}^2, \mu_{pb}^3, \mu_p^4$ within it are lagrangian multipliers and if the chosen constraints to relax are all unequal equations, these lagrangian multipliers cannot be of minus value: $(\mu_a^1, \mu_{pba}^2, \mu_{pb}^3, \mu_p^4) \geq 0$. Therefore, Z_{LR} can be decomposed into two independent and easily solvable sub-problems. Adding to this another constant part Z_{Const}.

- *Sub-Problem* 1: Related to decision variable x_{pb}, h_b and constraints 5, 7 and 10.

  Objective function (Sub1):

$$
\begin{aligned}
Z_{Sub1} = min &+ \sum_{p\in P}\sum_{b\in B} x_{pb} D(V_b,h_b) - \sum_{p\in P}\sum_{b\in B}\sum_{a\in A} \mu_{pba}^2 \gamma_{ab} x_{pb} \\
&+ \sum_{p\in P}\sum_{b\in B}\sum_{p'\in P} \mu_{pb}^3 \lambda_{p'} x_{p'b} - \sum_{p\in P}\sum_{b\in B} \mu_{pb}^3 h_b
\end{aligned}
$$

Since, all the sub-categories of Z_{Sub1} are an additive total based on p, we split it into the quantity P of connected *equation*, the total being:

$$Z_{Sub1} = \sum_{p\in P} Z_{Sub1p}(p)$$

h_b within Z_{Sub1p} is a continuous value, therefore we add the constraint 11 to limit the range of h_b. Golden section search method may be a good way for inquiring the maximum value of line segment:

$$\underline{H_b} \leq h_b \leq \overline{H_b}$$

Sub 1:

$$
\begin{aligned}
Z_{Sub1p}(p) = min &+ \sum_{b\in B} x_{pb} D(V_b,h_b) - \sum_{b\in B} \mu_{pb}^3 h_b \\
&+ \sum_{b\in B} x_{pb} (\lambda_p \sum_{p'\in P} \mu_{p'b}^3 - \sum_{a\in A} \mu_{pba}^2 \gamma_{ab})
\end{aligned} \qquad (13)
$$

$$\underline{H_b} \leq h_b \leq \overline{H_b}$$

- *Sub*-Problem 2: Related to decision variable $y_{aa'}$, z_{pa}, g_a and constraints 6, 8 and 9.

  Sub 2:

$$Z_{Sub2} = \min + \sum_{a\in A}\sum_{a'\in A} y_{aa'} (E_{aa'} + C(M_{a'}, g_{a'})) + \sum_{a\in A} \mu_a^1 g_a \qquad (14)$$

Since, all the sub-categories within Z_{Sub2} are additive values based on a, then we split it into connected equations with *quantity* of A, their total being:

$$Z_{Sub2} = \sum_{a\in A} Z_{Sub2a}(a)$$

g_a is a continuous value within Z_{Sub2a}, we therefore add the Constraint 12 to limit the range of g_a. Golden section search method may be a good one for inquiring the most optimized value of line segment $\underline{G_a} \leq g_a \leq \overline{G_a}$.

Sub2a:

$$
\begin{aligned}
Z_{Sub2a}(a) = min &+ C(M_a,g_a)(\sum_{p\in P} z_{pa} + \sum_{a'\in A} y_{aa'}) \\
&+ \sum_{p\in P} z_{pa} (F_{pa} + \mu_a^1 \lambda_p + \sum_{b\in B} \mu_{pba}^2 + \mu_p^4) \\
&+ \sum_{a'\in A} y_{aa'} (E_{aa'} - \alpha_a (\mu_a^1 + \mu_{a'}^1)) - \mu_a^1 g_a \\
&\underline{G_a} \leq g_a \leq \overline{G_a}
\end{aligned} \qquad (15)
$$

Constant Part (Const):

$$Z_{Const} = + \sum_{a\in A} \mu_a^1 \delta_a$$

According to the weak lagrangean duality theorem [12], for any $(\mu_a^1, \mu_{pba}^2, \mu_{pb}^3, \mu_p^4) \geq 0$, Z_{LR} is a lower bound on Z_{IP2}. The dual problem $Z_D = max\ Z_{LR}(\mu_a^1, \mu_{pba}^2, \mu_{pb}^3, \mu_p^4)$ is then constructed to calculate the tightest lower bound. To solve problem Z_D, the sub-gradient method is applied.

To let the primal problem being solved in an easier way, we relaxed a few constraints from it, so the decision variables we obtained from the dual problem could be not applicable for a solution, therefore, we must develop another heuristic approach, which is the reason why we proposed a LR-based algorithm (Algorithm 1), allowing these decision variables to build up a primal feasible solution within the primal problem.

Algorithm 1: LR-based.

Step 1: *Calculate the coefficients z_{pa} and x_{pb} of each system vulnerabilities, the results are classified in ascending order.*

Step 2: *According to the classification obtained in step 1, we first choose the security engineer, then the IT support center. After this we also, check if it fulfills all the constraints, if not then go on to the next in the raw, until all the parameters are fulfilled.*

Step 3: *Calculate the coefficient $y_{aa'}$ of each security engineer, the result being classified in ascending order.*

Step 4: *According to the list obtained in step 3, search for the $y_{aa'}$ who is in accordance with the constraints and can minimize the objective function.*

Step 5: *Use z_{pa}, x_{pb} and $y_{aa'}$ decisions result to calculate h_b and g_a.*

Step 6: *Insert all the decision variables into the objective function to obtain the result*

Step 7: *Note down the fixed decision in step 2 and 4 as our most optimized strategies and finish the math.*

6. Computational Experiment

The detailed parameters used in the experiments are shown in Table 4.

Table 4. Experimental parameter settings of lagrangean relaxation.

Parameters	Value
Iteration Counter Limit	2000
Improvement Counter Limit	100
Initial Upper Bound	The 1st Result of Getting a Primal Feasible Solution
Initial Multiplier Value	0
Initial Scalar of Step Size	1

The proposed model of this paper is for non-linear integer programming problems. The near optimal solution can be obtained. The LR value means the primal feasible solution derived by the Lagrangean relaxation process and the LB represents the lower bound obtained from the Lagrangean relaxation process. Moreover, the gap is calculated by:

$$\frac{LB - LR}{LR} \times 100\% \quad (16)$$

The model experimental results are shown in Table 5.

Table 5. Experimental results.

Total Quantity of Device Vulnerabilities	LB	LR	Gap (%)
50	1.7431	0.9925	75.6272
100	2.9382	1.5752	86.5286
150	2.3071	1.2012	92.0649
200	3.1348	1.5683	99.8861

7. Conclusions

The vulnerability patch update policy has the particularity of being multifunctional and requires the combined utilization of actual available resources, therefore the topic of this study is the "transfer of device vulnerabilities to the adequate system engineer", the main purpose of this protocol being that by transferring minor threat to second-tier engineers for patch update, the prior issues are handled with more important resources, so that in the end the device vulnerability survival time is minimized and the patch update efficiency optimized.

In this paper, we propose a near optimal patching assignment algorithm for security vulnerabilities. We build up the problem into a mathematical model, which is a non-linear non-convex mixed integer programming problem. We are looking for an objective function to minimize the survival time of the device vulnerabilities before the patch is updated, but we take into consideration the backup devices of the IT support center and its loading change, plus the working hours' limit of its personnel.

Moreover, to construct the mathematical model and facilitate the equation, we set up the presupposition that "each device vulnerability is handled by an individual IT support center", "each device vulnerability is handled by an individual security engineer", "each device vulnerability can be transferred only to an individual security engineer for patch update". These parameters simplify the complexity of the problem for this first approach, how to lessen these limits will be the main focus of future studies, adding to this the fact that other resources will also be taken into consideration.

Acknowledgments

The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] Alhazmi O., Malaiya Y., and Ray I., "Measuring, Analyzing and Predicting Security Vulnerabilities in Software Systems," *Computers and Security*, vol. 26, no. 3, pp. 219-168, 2007.

[2] Anderson R. and Moore T., "The Economics of Information Security," *Science*, vol. 314, no. 5799, pp. 610-613, 2006.

[3] Andrew C., "The Five Ps of Patch Management: Is there a Simple Way for Businesses to Develop and Deploy an Advanced Security Patch Management Strategy?," *Computers and Security*, vol. 24, no. 5, pp. 362-363, 2005.

[4] Arbaugh W., Fithen W., and McHugh J., "Windows of Vulnerability: A Case Study Analysis," *Computer*, vol. 33, no. 12, pp. 52-59, 2000.

[5] Arora A., Krishnan R., Telang R., and Yang Y., "An Empirical Analysis of Software Vendors Patch Release Behavior: Impact of Vulnerability Disclosure," *Information System Research*, vol. 21, no. 1, pp. 115-132, 2010.

[6] Arora A., Telang R., and Xu H., "Optimal Policy for Software Vulnerability Disclosure," *Management Science*, vol. 54, no. 4, pp. 642-656, 2008.

[7] August T. and Tunca T., "Let the Pirates Patch? an Economic Analysis of Software Security Patch Restrictions," *Information Systems Research*, vol. 19, no. 1, pp. 48-70, 2008.

[8] Brykczynski B. and Small R., "Reducing Internet-based Intrusions: Effective Security Patch Management," *IEEE Software*, vol. 20, no. 1, pp. 50-57, 2003.

[9] Cavusoglu H., Cavusoglu H., and Zhang J., "Security Patch Management: Share the Burden or Share the Damage," *Management Science*, vol. 54, no. 4, pp. 657-670, 2008.

[10] Fisher M., "An Applications Oriented Guide to Lagrangian Relaxation," *Interfaces*, vol. 15, no. 2, pp. 10-21, 1985.

[11] Fisher M., "The Lagrangian Relaxation Method for Solving Integer Programming Problems," *Management Science*, vol. 27, no. 1, pp. 1-18, 1981.

[12] Geoffrion A., "Lagrangean Relaxation and its Use in Integer Programming," *Mathematical Programming Study*, vol. 2, pp. 82-114, 1974.

[13] Gerace T. and Cavusoglu H., "The Critical Elements of the Patch Management Process,"

Communications of the ACM, vol. 52, no. 8, pp. 117-121, 2009.
[14] Gordon L. and Loeb M., "The Economics of Information Security Investment," *ACM Transactions on Information and System Security*, vol. 5, no. 4, pp. 438-457, 2002.
[15] Ioannidis C., Pym D., and Williams J., "Information Security Trade-offs and Optimal Patching Policies," *European Journal of Operational Research*, vol. 216, no. 2, pp. 434-444, 2012.
[16] Karabey B. and Baykal N., "Attack Tree based Information Security Risk Assessment Method Integrating Enterprise Objectives with Vulnerabilities," *the International Arab Journal of Information Technology*, vol. 10, no. 3, pp. 297-304, 2013.
[17] Lai Y. and Hsia P., "Using the Vulnerability Information of Computer Systems to Improve the Network Security," *Computer Communications*, vol. 30, no. 9, pp. 2032-2047, 2007.
[18] Lesk M., "Cybersecurity and Economics," *IEEE Security and Privacy*, vol. 9, no. 6, pp. 76-79, 2011.
[19] Lin F., "Quasi-static Channel Assignment Algorithms for Wireless Communications Networks," *in Proceedings of the 12th International Conference on Information Networking*, Tokyo, pp. 434-437, 1998.
[20] Rescorla E., "Is Finding Security Holes a Good Idea?," *IEEE Security and Privacy*, vol. 3, no. 1, pp. 14-19, 2005.
[21] Shih H., "Safety Transfer Medical Assignment Algorithm for Emergency Medical Services," *Master Thesis*, Chung Yuan Christian University, 2004.
[22] Okamura H., Tokuzane M., and Dohi T., "Optimal Security Patch Release Timing under Non-homogeneous Vulnerability-Discovery Processes," *in Proceedings of the 20th International Symposium on Software Reliability Engineering*, Mysuru, Karnataka, pp. 120-128, 2009.

Shao-Ming Tong received his MS degree in electrical engineering from the National Central University in 1997. He is currently a PhD student in information management at the National Taiwan University. His current research interests include computer networks, information security and communication/ network forensics.

Chien-Cheng Huang received his MS degree in information management from the National Chiao Tung University in 2008 and his PhD degree in information management from the National Taiwan University in 2014. He is an adjunct assistant professor with the National Taipei University of Nursing and Health Sciences. His current research interests include data mining, business intelligence, information security and cyber/network forensics.

Feng-Yu Lin received his PhD degree from the National Chiao Tung University in 2004 and his second PhD degree in Information Management from the National Taiwan University in 2014. He is an adjunct assistant professor with the Department of Criminal Investigation, Central Police University in Taiwan. His research interests include communication/network forensics, data mining, and information security.

Yeali Sun received her BS from the Computer Science and Information Engineering department of National Taiwan University in 1982 and MS and PhD degrees in Computer Science from the University of California, USA in 1984 and 1988, respectively. From 1988 to 1993, she was with Bell Communications Research Inc. (Bellcore; now Telcordia). She jointed National Taiwan University in 1993. Currently, she is a professor of the Department of Information Management. Her research interests are in the area of wireless networks, quality of service and pricing, internet security and forensics, scalable resource management and business model in cloud services and performance modeling and evaluation.