COST REDUCTION ESTIMATION METHOD OF A SOFTWARE VULNERABILITY MANAGEMENT TOOL

Satoshi Yashiro¹,², Pranay Verma³, Norihisa Komoda⁴ and Takenao Ohkawa⁵

¹Graduate School of System Informatics, Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe 657-8501, Japan
²Research & Development Group, Hitachi, Ltd., 292, Yoshida-cho, Totsuka-ku, Yokohama 244-0817, Japan
³R&D Centre, Hitachi India, Pvt. World Trade Center, #S 704, 7th floor, Brigade Gateway Campus, No.26/1, Dr. Rajkumar Road, Malleswaram-Rajajinagar, Bengaluru 560-055, India
⁴Code Solutions Co., Ltd., 1-2-11F, Edobori, Nishi-ku, Osaka 550-0002, Japan
⁵Graduate School of System Informatics, Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe 657-8501, Japan

ABSTRACT
Software vulnerability identification is necessary work to have IT assets secure in organizations. There can be so many vulnerabilities found as usual, and the severities of the vulnerabilities are various. Attack path analysis is known as helpful to clarify the severities, and a vulnerability management tool with attack path analysis will reduce vulnerability management cost. However, the amount of cost reduction by applying the tool is not clear in each case. Also, it is difficult to estimate the amount of the cost reduction. In this paper, we introduce an efficient approach to estimate the cost benefit of implementing a vulnerability management tool with attack path analysis into actual vulnerability management process by integrating qualitative analysis with quantitative analysis.

KEYWORDS
Vulnerability Management, Attack Path Analysis, CBA: Cost Benefit Analysis, Cost Factor Tree

1. INTRODUCTION
In these days, cyber security risks are getting severer. Attackers are trying to intrude to business systems by using malware or penetration tools. Those malware or tools typically use software vulnerability of servers or devices, so it is important to identify software vulnerabilities regularly and fix them immediately.

On the other hand, organizations tend to pay less efforts to vulnerability scanning and fixing activity because such activities typically spend much cost. Some security tools provide aids to reduce such costs, and one of the known tools is vulnerability management tool with attack path analysis. When we use the tool, we can detect vulnerabilities in enterprise network systems with less efforts and identify how attackers will intrude each server or device by attacking the vulnerabilities step by step. Even though some servers or devices have severer vulnerabilities, actual severity is not so high when the servers or devices are far from the attack path. This means that attack path analysis may contribute to prioritize vulnerabilities to be fixed and reduce costs by focusing on really severe vulnerabilities. This tool typically automates vulnerability scanning procedure, so cost reduction by implementing the tool into security monitoring process will also affect vulnerability scanning procedure.

Qualitatively, we can understand the tool will bring a certain amount of cost reduction in security monitoring process, but it is not easy to identify how much the cost will be reduced quantitatively.

In this paper, we introduce a method on cost reduction analysis of software vulnerability management process with a tool which supports attack path analysis. We also show the method needs less efforts to estimate the reduction. The method based on a CBA (Cost Benefit Analysis) that is originally invented for cost reduction analysis of production process in plants is expanded to software intensive management process.
2. VULNERABILITY SCANNING AND UPDATE PROCESS

2.1 Typical Procedure of Vulnerability Scanning and Update

In an enterprise network system, there are many instances of software running on servers or devices. Those instances unfortunately have vulnerabilities in some time. Security organizations such as NIST (National Institute of Standards and Technology) announce existence of such vulnerabilities regularly, and vendors of the software or OSS (Open-Source Software) community disclose patches to the vulnerable version of software. IT asset management teams in organizations scan the assets and survey the vulnerability disclosure. Once they find vulnerabilities which may affect their IT assets, they try to fix the vulnerabilities to avoid cyber security attacks.

Software vulnerability scanning and update procedure typically follows the manner described in figure 1.

Figure 1. Typical procedure of software vulnerability scanning and update

<table>
<thead>
<tr>
<th>Scanning installed software</th>
<th>Searching vulnerability DB</th>
<th>Identify vulnerabilities in installed software</th>
<th>Identify severity of the vulnerabilities</th>
<th>Making update plan and fixing the vulnerabilities</th>
</tr>
</thead>
</table>

Figure 2 shows a concept of attack path analysis and vulnerability prioritizing procedure that is covered by the vulnerability management tool with attack path analysis. The tool initially collects IT assets information into the asset repository, then executes vulnerability scan regularly. Once vulnerabilities are found, the tool conducts an attack path analysis. Based on the result of the analysis, the tool calculates priorities of the vulnerabilities to be fixed. Sometimes the priorities are different from severities of the vulnerability itself. For example, a vulnerability with higher severity in a server placed in a secluded area has less priority to a vulnerability with middle severity in a web server opened to the Internet. In the case of figure 2, there are five vulnerabilities found in the network, where v1 has the highest severity and v5 has the lowest severity. However, once the attack path is found as v1, v2, and v5, then the priority of v5 becomes higher than v3. In this case, not to fix v3 and v4 immediately can be an option. When the option is chosen, the team can save the costs on vulnerability fixing process. From the other viewpoint, to minimize vulnerabilities to be fixed immediately could contribute to avoid opportunity losses that are brought by server down during the patching.

Figure 2. Concept of vulnerability scanning and prioritizing process with attack path analysis
2.2 Difficulty of Cost Reduction Estimation

Vulnerability scanning and update process is a process that is needed to execute often and regularly, and workload of the process would be large when an organization has large numbers of servers or devices. The process typically becomes costly work. Also, downtime of servers or devices during patching will bring loss cost such as sales loss or production loss. Qualitatively, it is easier to understand the vulnerability management tool will contribute to reduce such costs, but quantitatively, it is not easy to understand the investment of the tool will pay for the amount of cost reduction or not. If the investment will not pay for the amount of the cost reduction, not using the tool might be a reasonable option.

The cost reduction estimation is typically difficult work, because there can be a lot of cost factors to be considered. Some factors will contribute to cost reduction directly, but some other factors will affect indirectly. Also, to quantify each cost factors or contribution ratio is difficult work. That is why quantitative effect estimation has not usually been conducted before we make a decision that the software vulnerability management tool should be introduced to the actual software vulnerability management process or not.

3. PROCEDURE OF COST REDUCTION ANALYSES

3.1 Basic Idea of the Proposed Method

In this paper, we propose a CBA based effect estimation method. In the CBA approach, secondary effect or thirdly effect can be considered, but we focused on only primary effect to avoid over estimation. In vulnerability management case, primary effect means reduction of work time, rework time, or unit human cost of skilled persons to conduct vulnerability management process. When the vulnerability management tool is implemented into the management process, work process would be changed, so we can estimate the changes of work cost by focusing on only changed work items. To identify work process and affected work items, we introduce “stakeholder table” which makes identifying process easier. Stakeholder table is a table which lists stakeholders of vulnerability management process, their concerns, and their happy and unhappy situation about the concerns. This table contributes to list all work items related to the stakeholders and identify which work items will be affected by the vulnerability management tool.

Once all stakeholders are listed, work activities of the stakeholders would be depicted. The vulnerability management tool will affect some activities, and the activities may change as automated, less manual work, or less professional skills. Such changes of activities are drilled down to the cost drivers, and we can finally estimate the cost difference by giving assumed parameters to the cost drivers.

To identify the stakeholders, their concerns, their activity changes, cost drivers and cost parameters, series of workshops are conducted. Basically, the workshops are held at three or four times as illustrated in figure 3. Software vulnerability management process is common process in any organizations even when they have IT assets, so the structures of the cost factor trees are expected as almost the same. Only cost parameters are expected as different in the organizations. Therefore, once a cost factor tree is produced through the workshops, it is expected to reuse the tree to estimate the amount of cost reduction in other organization case by just arranging cost parameters.

<table>
<thead>
<tr>
<th>Step</th>
<th>To identify stakeholders’ issues</th>
<th>To clarify business flow</th>
<th>To estimate value of the tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>List up stakeholders</td>
<td>Identify business flow before and after the tool is used</td>
<td>Depict cost factor tree</td>
</tr>
<tr>
<td></td>
<td>List up their concerns</td>
<td>Clarify quantitative KPIs</td>
<td>Estimate cost parameters</td>
</tr>
<tr>
<td>Output</td>
<td>Stakeholder table</td>
<td>Business workflow diagram</td>
<td>Cost factor tree</td>
</tr>
</tbody>
</table>

Figure 3. Outline of cost reduction estimation workshops
Table 1. Identified stakeholder’s concern, happy and unhappy situation by applying Stakeholder table

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>Concerns</th>
<th>Happy</th>
<th>Unhappy</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSO: Chief Security Officer</td>
<td>• Decision making</td>
<td>• Speedy</td>
<td>• Slow</td>
</tr>
<tr>
<td></td>
<td>• Correctness of decision</td>
<td>• Right</td>
<td>• Wrong</td>
</tr>
<tr>
<td></td>
<td>• Performance of security operations</td>
<td>• Easy to monitor</td>
<td>• Difficult to monitor</td>
</tr>
<tr>
<td>Security Analysis and Planning Team</td>
<td>• Audit process</td>
<td>• Less manual work</td>
<td>• Much manual work</td>
</tr>
<tr>
<td></td>
<td>• Compliance level</td>
<td>• Meets standards</td>
<td>• Doesn’t meet standards</td>
</tr>
<tr>
<td></td>
<td>• Mitigation plan</td>
<td>• Speedy and accurate</td>
<td>• Slow and less accurate</td>
</tr>
<tr>
<td></td>
<td>• Remaining high vulnerabilities</td>
<td>• Least</td>
<td>• Many</td>
</tr>
<tr>
<td></td>
<td>• Vulnerability detection</td>
<td>• Full coverage</td>
<td>• Low coverage</td>
</tr>
<tr>
<td></td>
<td>• Report to CSO</td>
<td>• Easy loss estimation</td>
<td>• Difficulty to access loss</td>
</tr>
<tr>
<td></td>
<td>• Patch effectiveness</td>
<td>• Less side effects</td>
<td>• Much side effects</td>
</tr>
</tbody>
</table>

3.2 Identification of Stakeholders and their Concerns

The first step of the analysis is to identify stakeholders of software vulnerability management process and their concerns. In this step, all stakeholders should be captured, and their concerns, happy and unhappy situation about the concerns should be identified. Table 1 is an example of the stakeholder table.

3.3 Identification of Procedure Changes by Applying the Tool

The second step is to clarify stakeholders’ work items using “Ex-table (Experience table)” diagram. Ex-table is a table formed diagram which lists stakeholders in the row direction and work phases in the column direction and illustrates each stakeholders’ work items as a swim-lane. Figure 4 is an example of Ex-table. It is supposed that the vulnerability management tool will contribute to automated vulnerability detection. Automation of vulnerability detection will bring fewer remaining vulnerabilities than those in manual work. Such state change is also captured using a comment box in the Ex-table.

3.4 Identification of Cost Drivers

The final step is to estimate cost before and after vulnerability management tool is implemented. To conduct this analysis, cost factor tree is used. Figure 5 is an example of cost factor tree. First level is total cost, second level is total cost of a work phase that is same as a column in the Ex-table, and third level is cost of an activity that is same as a work item in the Ex-table. Fourth and deeper levels are cost drivers or elements to calculate cost drivers.

Stakeholder Risk Management Planning Asset Management Vulnerability Detection
CSO: Chief Security Officer
Security Analysis and Planning Team
IT Operation Team
Compliance Team

B: More remaining vulnerabilities
   A: Less remaining vulnerabilities

Figure 4. Process swim lane by applying Ex-table
4. RESULTS AND EVALUATION OF COST REDUCTION BY APPLYING VULNERABILITY MANAGEMENT TOOL

4.1 Conditions of Cost Reduction Estimation

We conducted an estimation of cost reduction on vulnerability management process by applying the method explained in chapter 3. To conduct the estimation, we assumed some cost parameters which are typical case in enterprise organization in India. One of them is unit human cost per day shown in table 2(a), where INR is the currency unit in India. There are five persons in different roles listed, and they have different unit costs. Another parameter is average quantities of assumed enterprise network shown in Table 3(b). There are four cost parameters listed, number of servers, number of security reports per year, number of vulnerabilities per server, and number of audits per year. These parameters are estimated by security researchers who joined the workshops, based on their experiences on working with security management team in India.

<table>
<thead>
<tr>
<th>(a) Assumed assigned person</th>
<th>Cost (INR/day)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSO (Chief Security Officer)</td>
<td>10,000</td>
</tr>
<tr>
<td>Senior security team member</td>
<td>5,000</td>
</tr>
<tr>
<td>Junior security team member</td>
<td>2,500</td>
</tr>
<tr>
<td>IT operator</td>
<td>1,000</td>
</tr>
<tr>
<td>Compliance team member</td>
<td>3,500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(b) Cost parameter</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average number of servers per network</td>
<td>500</td>
</tr>
<tr>
<td>Average number of security reports per year</td>
<td>52</td>
</tr>
<tr>
<td>Average number of vulnerabilities per server</td>
<td>20</td>
</tr>
<tr>
<td>Average number of audits per year</td>
<td>2</td>
</tr>
</tbody>
</table>

4.2 Results of Each Step

Through the first step, we derived five stakeholders such as CSO (Chief Security Officer) or IT operation team, and nineteen concerns of them. Security planning and analysis team has the most concerns, and it suggests that the team might be the main target of the vulnerability management tool.

Through the second step, we derived ten work phases and thirty-one activities. Four out of the thirty-one activities are suggested to change by applying the vulnerability management tool.

Finally, we estimated cost change before and after the tool applied by using cost factor tree. In the third step, it is found that only four activities are changed by applying the tool, but the estimation was required to almost all activities. Because once quantity change of high severity vulnerability happens, the change affects later activities.
4.3 Total Amount of Cost Reduction Estimation

Total amount of cost reduction is estimated as in table 3. In table 3, where only amount changed activities are listed. Two activities, vulnerability identification and report vulnerability are totally automated, and its work-related costs become zero. Activity of identify severity becomes much cost down thanks to the tool’s support. Other activities become cost down because some of high severity vulnerabilities become lower priority based on attack path analysis and need of immediate patching is lower.

As the result of applying the method to cost reduction analysis of software vulnerability management tool, we successfully identified the estimated quantity of the cost reduction. Necessary and sufficient cost drivers are captured by depicting Ex-table diagram, and the cost drivers are divided into cost parameters through cost factor analysis.

The analysis was conducted by six or eight members through three days workshop. Three out of the eight members have knowledge on the vulnerability management process and the vulnerability management tool with attack path analysis, and other two out of the eight members have experiences to conduct this kind of workshop. During the workshops, the members are required to do some pre-assignments, but only six hours are spent in actual workshop discussion. This result shows the proposed method gives supports to conduct efficient quantitative effect analysis of software vulnerability management tool. Also, we can reproduce another estimation in assuming different organizations or different countries by altering assumed cost parameters described in table 2.

Table 3. Result of cost reduction analysis

<table>
<thead>
<tr>
<th>Phase</th>
<th>Activity</th>
<th>Cost before the tool (M-INR/year)</th>
<th>Cost after the tool (M-INR/year)</th>
<th>Cost difference (M-INR/year)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vulnerability detection</td>
<td>Vulnerability Identification</td>
<td>5.0</td>
<td>0</td>
<td>-5.0</td>
</tr>
<tr>
<td>Vulnerability analysis</td>
<td>Report vulnerabilities</td>
<td>3.86</td>
<td>0</td>
<td>-3.86</td>
</tr>
<tr>
<td></td>
<td>Identify severity</td>
<td>4.24</td>
<td>0.618</td>
<td>-3.622</td>
</tr>
<tr>
<td></td>
<td>Decide action to high-risk vulnerabilities</td>
<td>0.13</td>
<td>0.0325</td>
<td>-0.0975</td>
</tr>
<tr>
<td>Patch testing</td>
<td>Patch approval</td>
<td>0.13</td>
<td>0.065</td>
<td>-0.065</td>
</tr>
<tr>
<td>Patch execution</td>
<td>Patch execution</td>
<td>0.22</td>
<td>0.20</td>
<td>-0.02</td>
</tr>
<tr>
<td>Compliance check</td>
<td>Check compliance</td>
<td>6.26</td>
<td>5.69</td>
<td>-0.57</td>
</tr>
<tr>
<td>Security monitoring</td>
<td>Risk analysis and report</td>
<td>125</td>
<td>25</td>
<td>-100</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>144.84</td>
<td>31.61</td>
<td>-113.23</td>
</tr>
</tbody>
</table>

5. CONCLUSION

In this paper, we introduced a method to estimate cost reduction of software vulnerability management process with a software vulnerability management tool which supports attack path analysis. Through a six hours workshop with eight persons based on the introduced method, we found that the tool will qualitatively contribute to some cost reduction especially in security monitoring related activities. Also, it is clarified that the amount of expected cost reduction was 113.23 M-INR/year in typical situation. The introduced method can efficiently estimate the reduction cost and can reproduce another estimation in different organizations or different countries by altering some assumed cost parameters.

REFERENCES

