RESOURCE-AWARE CONFIGURATION MANAGEMENT
USING XML FOR MITIGATING INFORMATION ASSURANCE
VULNERABILITY

Namho Yoo
Department of Computer Science, The George Washington University,
801 22nd Street, N.W., Room 730, Washington, DC 20052, USA

Keywords: XML, Configuration Management, Vulnerability, Information Assurance, System Engineering, Resource-Aware Management

Abstract: This paper suggests an XML-based configuration management for mitigating information assurance vulnerability. Once an information assurance vulnerability notice is given for a system, it is important for reducing massive system engineering efforts for configuration management. When multiple systems are updated by security patches for mitigating system vulnerability, configuration management based on system resource is trivial, in order to increase accuracy, efficiency and effectiveness of software processes. By employing XML technology, we can achieve seamless and efficient configuration management between heterogeneous system format as well as data formats in analysing and exchanging the pertinent information for information assurance vulnerability. Thus, when a system is updated to improve system vulnerability, the proposed XML-based configuration management mechanism refers to the system resource information and analyse the security model and posture of affected sustained system and minimize the propagated negative impact. Then, an executable architecture for implementation to verify the proposed scheme and algorithm and testing environment is presented to mitigate vulnerable systems for sustained system.

1 INTRODUCTION

In a sustained system, configuration management efforts toward software process management are required for decision-making(Arnold, 1993). For applying new vulnerability requirements for system security, configuration management should be considered prior to system implementation for minimizing the negative impact to another system. If security requirement has an ongoing feature to be considered, even after implementing the change, configuration management efforts for system security are still required for continued decision-making.

With a given changing requirement, a System Engineer and an Information Assurance (IA) Engineer should be involved in the configuration management process. In the case of large-scale and globally deployed systems, engineering evaluations for configuration management rely upon the test results of developmental laboratories. Configuration management on the system interfaces is dependent upon knowledge about interface details based on system resource information. If changing security requirement is not a one-time request, it is necessary to involve engineers for continued analysis with more objective evidence from the system resource and build a stronger foundation(MIL, 1997)

In this paper, as an applicable security requirement, we focus on information assurance vulnerability(DoD, 2004). This security requirement is an appropriate example of an applied to entire systems on an ongoing basis(Yoo, 2004). We present a globally deployed US health system (see Figure 1 for example) and suggest an approach to handle the above issues.

Even though System Engineers have sufficient knowledge on each system resource, it will be very difficult to trace all the detailed records on the system engineering efforts during the impact analysis for configuration management. Thus, this paper suggests a resource-aware configuration management process, which is a good vehicle for improving the efficiency of the impact analysis by managing the security information systematically during the process for configuration management.
More specifically, we propose using the extensible markup language (XML) to conduct resource-aware configuration management for mitigating information assurance vulnerability. XML is widely accepted as a standard for an information exchange on the World Wide Web (W3C, 2000). Accordingly, a resource-aware information model using XML for security in sustained systems has been developed, using offline documentation. However, this scheme is still a labor-intensive procedure.

This approach is based on XML representation, with improving the configuration management for information assurance vulnerability with applying security notice. The analysis uses a case study in the globally deployed US health systems, which were analyzed manually by System Engineers. An efficient scheme based on resource-aware configuration management scheme using XML is discussed.

2 BACKGROUND AND PROBLEM STATEMENTS

The system shown in Figure 1 is a world-wide deployed health system involving eight sub-systems, A through H. The configuration management is essential for good decision support. As an example, Table 2 shows us the system resource information for decision-making whether or not given notice is applicable.

![Figure 1: Sustained System Architecture affected by Information Assurance Vulnerability](image)

Also, as some resource information may exist without specification gathered, gathering specification and verifying it with comparison of the current status is another difficult problem to specify the Engineering Change Proposal (ECP) for Configuration Management (CM), as a common vehicle for final decision making. Figure 2 shows us the response policy and process of IA vulnerability for applicability.

![Figure 2: IA vulnerability Response Policy and Process](image)

Table 1: System Resource Information

<table>
<thead>
<tr>
<th>OS</th>
<th>DB Applicat.</th>
<th>Web Applicat.</th>
<th>Developing Lang.</th>
<th>Tool</th>
<th>JVM &amp; JRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>OS-1</td>
<td>DB-4</td>
<td>IE-1</td>
<td>T-1</td>
<td>1.3.1</td>
</tr>
<tr>
<td>B</td>
<td>OS-2</td>
<td>DB-2</td>
<td>IE-2</td>
<td>T-2</td>
<td>1.3.2</td>
</tr>
<tr>
<td>C</td>
<td>OS-1</td>
<td>DB-3</td>
<td>IE-3</td>
<td>T-3</td>
<td>1.3.2</td>
</tr>
<tr>
<td>D</td>
<td>OS-4</td>
<td>DB-3</td>
<td>IE-1</td>
<td>T-4</td>
<td>1.3.3</td>
</tr>
<tr>
<td>E</td>
<td>OS-2</td>
<td>DB-2</td>
<td>IE-4</td>
<td>T-5</td>
<td>1.3.3</td>
</tr>
<tr>
<td>F</td>
<td>OS-5</td>
<td>DB-3</td>
<td>IE-3</td>
<td>T-6</td>
<td>1.3.3</td>
</tr>
<tr>
<td>G</td>
<td>OS-3</td>
<td>DB-4</td>
<td>IE-3</td>
<td>T-7</td>
<td>1.3.3</td>
</tr>
<tr>
<td>H</td>
<td>OS-2</td>
<td>DB-2</td>
<td>IE-2</td>
<td>T-8</td>
<td>1.3.3</td>
</tr>
</tbody>
</table>

Table 1: System Resource Information
management decision? 5) Is there any simple and powerful way to follow for configuration management?

3 XML-BASED RESOURCE INFORMATION DESCRIPTION

In this paper, we propose an XML-based representation of gathered specification.

![Figure 3: Vulnerability Notice XML and ECP XML for CM](image)

Figure 3 is an example of demonstrating a specification described with XML format. In the column, an example of IA vulnerability information is given, and the ECP submittal form based on XML representation is given in the right column. Using proposed lightweight XML representation; we generate a simple, powerful, and customized model for enhancing the model for configuration management for mitigating IA vulnerability.

4 RESOURCE-AWARE CONFIGURATION MANAGEMENT STEPS

We can observe each step smoothly processed based on XML DOM tree(W3C, 2000). Strengthening the security model and security posture is possible using a proposed model. Furthermore, we upgrade and customize system resource information as the resource ontology. The full version of this research had detailed information about resource information. If we use updating resource information, it is possible for us to describe the security accreditation boundary more clearly and realistically by applying the workstation level information.

![Figure 4: XML-Based CM Steps](image)

Figure 4 is a graph based on architecture design similar to that shown previously. The root node is described as a monitor and the hierarchical information as a tree. A leaf is a user level or changing status. In other words, using DOM tree representation, an information entity holding vulnerability information and changing information on configuration management is represented as the same model.

Through comparing the previous DOM tree and current version, we recognize which elements of the security profile information are changed. As an input, given user security user information and security log files are used. While comparing the
XML DOM tree, we need to check the changing status like the steps (9-10) in Figure 4.

5 IMPLEMENTING PLAN

We describe the implementation plan to verify our proposed model and scheme. The Windows system is considered as the underlying hardware environment due to its pervasiveness and we also consider various commercial tools and reliable shareware utilities are planned for installation as the software environment.

![Diagram](image)

**Figure 5: Executable Architecture-based Implementation**

For example, we are considering the FirstACT for a virtual user generation and a script programming using Python for an interface between each software components. In Figure 5, the input artifacts are extracted and are converted to XML. Once the proposed software component in the IA vulnerability system converts XML to DOM, the main configuration management process is preceded. The Graphical User Interface is shown as well.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we consider the new issues rose by the configuration management based on resource aware information for IA vulnerability in a large scaled sustained system safety. We proposed a customized process by monitoring IAV using XML for enhancing impact analysis and presented a scheme for mitigating potential security vulnerability. Through an example of a health system, we address processes to apply information assurance vulnerability notice for system safety.

The ideas presented in this paper were being developed in the context of the XML desktop system and setup implementation for verifying the data. We plan to experiment with our architecture for applying performance evaluations as well in the near future. It is very desirable to implement a simulation system integrated with various change perspectives.

REFERENCES


MIL-STD-498, 1997 Software Development and Documentation, Department of Defense, December


W3C, 2000, Extensible Markup Language (XML) 1.0 , W3C Recommendation, October
