IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITB12709

This paper appears in the book, *Emerging Trends and Challenges in Information Technology Management, Volume 1 and Volume 2* edited by Mehdi Khosrow-Pour © 2006, Idea Group Inc.

Security by Integration, Correlation, and Collaboration: Integrating Verification Reusable Agents into SOAP Messages

Khalil A. Abuosba, Philadelphia University, PO Box 851978, Amman, 11185, Jordan, T. #: 962-2-637-4444 ext 465, F. #: 962-2-637-4440; kh_abuosba@philadelphia.edu.jo, abuosba@nol.com.jo

ABSTRACT

Web Services deployment success is based on crucial reusability and availability factors. Composition of Web Services depends on tasks fulfillment of the delivery technologies along with successful analysis of design requirements. The purpose of web services deployment is to offer various industries powerful machine to machine interaction capabilities. Service success or failure depends on correctness of the service design blocks deployment. This paper addresses the major components required to fulfill successful web services delivery. It focuses on a hybrid approach of s/w design; for the purpose of abstracting heterogeneity of computing environments and platforms, we introduce an integrated security agent within a SOAP envelope which incorporates functionalities such as vulnerability reduction sensors. The agent's functionality is to provide an auditing facility of vulnerabilities which may compromise a resource. These agents are proposed to be considered as a basic element of services design requirements for building SOAP based Web Services. These agents shall retrieve associated security vulnerabilities based on systems operational variables environments. The main functionalities of these agents are to define characteristics of systems environments for the purpose of filtering associated security vulnerabilities definitions in order to verify compliance of the systems security policy by alerting monitors.

INTRODUCTION

Software vulnerabilities management is considered to be one of the basic building blocks of systems security engineering. Due to the nature of the distributed information systems of being composed out of dispersed networked heterogeneous nodes; securing these nodes have always been and always will be a major challenge to all systems engineers. Example of services offered in heterogeneous environment is Web Services. The W3C defines a Web Service as "A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards".[1]. As mentioned in the previous definition; the Major Web Services technologies are XML, WSDL, SOAP, and UDDI. These loosely coupled technologies form the basic building block for any primitive web service; delivery of a web service has additional requirements. These delivery requirements are Life Cycle Fulfillment, Architecture Conformity, Security Attributes Integration, and QoS attributes integration (Fig.1). The purpose of introducing Web Services on the web is to facilitate machine-to-machine interaction. Web Services have added a public automation factor to P2P communications; it is expected that many industry participants will be moving from private EDI systems to public global interoperable inexpensive Web Services as soon as reliability and high QoS are assured. As our interaction with data available on the Net demands special processing and deduction capabilities, the World Wide Web is emerging into a more semantic orientation where data may be processed, shared, and reused across applications, organizations and communities boundaries using metadata processing technologies. A Web service is identified by a URI (Uniform Resource Identifier) whose public interfaces and bindings are defined and described using an XML component; its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols. Quality of Service attributes integration is considered to be major factors in determining service usability.

WEB SERVICES VULNERABILITIES FACTS

Many deployments of Web Services fail due to software vulnerabilities. According to the US-CERT's database (which contains 11692 different vulnerabilities); currently there are 95 different XML-Vulnerabilities, 4 SOAP-Vulnerabilities [2]; however examining vulnerabilities from another coordination center (The Common Vulnerabilities and Exposure, a coordinator with a 12614 vulnerabilities list), we retrieved 45 XML-entries, 9 SOAP-entries (entries are either CVS-Names or Candidates) [3]; refer to table-1 for a comparison between the two coordination centers.

Web services are deployed in heterogeneous environment based on P-2-P architecture; vulnerabilities which may compromise successful web services deployment are not limited to only XML and/or SOAP vulnerabilities (semantic technologies); but all infrastructural components are considered causes to threats; hence we came the conclusion that vulnerabilities must be managed successfully in order to minimize their destruction effects. Clearly, we will never be able to eliminate vulnerabilities; however we will be able to take some measures that shall reduce the threats. Key design considerations pertinent to quality of service offered are scalability, performance, reliability, availability and fault tolerance" [4]. Web Services are deployed in distributed environments; the QoS parameters are dependent on network route as well as end-points infrastructure. Measuring the response time, throughput, reliability, and availability of a web service is possible to be achieved; however neither the client nor the provider has control over the others' resources.

Figure 1. Basic Web services delivery requirements

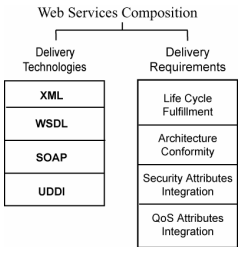
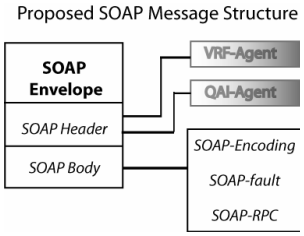


Figure 2. Proposed SOAP message composition



Vulnerabilities Disclosure and Assessment

Vulnerabilities are defined by NIAC as sets of conditions that lead or may lead to an implicit or explicit failure of the confidentiality, integrity, or availability of information system [5]. Vulnerabilities may be classified into two categories. The first category are the newly discovered vulnerabilities which we refer to as UNDOCUMENTED-Vulnerabilities (UV); the second category are the previously discovered which we refer to as the "DOCUMENTED-Vulnerabilities" (DV) category. All UVs must go through 9-steps life cycle process before resolution of vulnerability documentation based on the NIAC Vulnerability Life Cycle where any incident goes through research, verification, reporting, evaluation, acknowledgement, advisory and patch evaluation, patch release, and finally feedback and case closure.

Vulnerabilities Definition Disclosures

Vulnerabilities disclosure involves 4 stakeholders, the first is the discoverer, the second is the vendor, the third is the end user, and the fourth is the coordinator. Vulnerabilities disclosure involves a communication process among stakeholders in which encryption and digital signatures/certificates may be involved. A very crucial element in sharing information about vulnerabilities is time. The risk of miss-informing about a vulnerability; is noted and threats may be exposed due to notification process failure.

Vulnerability Assessment and Auditing Methods

"The purpose of the vulnerability assessment activity is to determine the exploitability of flaws or weaknesses in the system in the intended environment." [6]. The most common methods used for the purpose of vulnerabilities assessments are Vulnerabilities Scanning, Penetration Testing, and Integrity Checking by Hashers. These methods have several weaknesses. The most common weakness is that the latest vulnerabilities are usually missed as vulnerabilities definitions list are managed by the coordination centers. Vulnerability scanning solutions are provided as a service by a service provider remotely or as a software and/or hardware solutions implemented onsite locally. In either case vulnerabilities management requires tight coordination with the coordination center. Vulnerabilities scanners attempt to scan all possible vulnerabilities definition definitions (i.e. 10614 unique definitions from CVE).

VULNERABILITIES REDUCTION FRAMEWORK

We propose an integrated security agent to be based on the SOAP protocol where requester and provider exchange only indices (metadata) of vulnerabilities that is of relevancy to the technologies being used for the interaction fulfillment. This technique reduces the overhead needed for the vulnerabilities assessments as well as reduces remote traffic exchanges. For example suppose that the server is using an Oracle HTTP Web Server then the provider agent will be sending indices (according to vulnerabilities definitions listed in the CVE coordination center [7]); 6 definitions for the Oracle HTTP Web Server Vulnerabilities; 9 definition for the SOAP Vulnerabilities, and 45 definitions for the XML vulnerabilities, these definitions may be mapped into their OVAL equivalency; OVAL definitions are defined using the Open Vulnerability and Assessment Language, or more lately OVAL definitions are already defined for several vulnerabilities. The main goal for the deployment of

Figure 3. An example of a SOAP request for a current constrained OVAL vulnerabilities definitions

```
<env:Envelope
xmlns:env="http://www.w3.org/2003/05/soap-
envelope">
<env:Header>
<ovl:VRFassessment
xmlns:ovl="http://www.xyz.org/oval/VRF">
<ovl:OVALcharacteristicsTOKEN>
..8gh87jffd..
</ovl:OVALcharacteristicsTOKEN>
</ovl: VRFassessment>
</env:Header>
<env:Body>
<updt:GETcurrentLIST
xmlns:updt="http://www.coordinator.org/OVAL_u
pdate">
<updt:msg>
Vulnerabilities Check
</updt:msg>
</updt:GETcurrentLIST>
</env:Body>
</env:Envelope>
```

VRFs agents is to provide a more secured computing environment for parties involved in web services transaction. These agents act as a trust conveyer by performing benchmarking tests on services end-points.

VRF-AGENT DESIGN REQUIREMENTS GUIDELINES

In order for the web services to be functional and operational; design requirements call for management simplicity of the services. We obtain simplicity by securing hybrid specifications within the service. For assurances purposes, Vulnerabilities-agents must exchange a security token to a vulnerabilities coordination server and request a current list of vulnerabilities definitions. This token is considered to be a hash function value which may be computed based on pre-defined attributes of the requesters computing environment variables, based on this value; the server will pass strictly the vulnerabilities definitions that are of interest to the requesting agent. These variables may be architecture specifications, API specifications, Operating system footprint, ..etc.

This methodology will narrow down the list of vulnerabilities of interest resulting reduction of computing resources overheads as only vulnerabilities that effects the environment computing variables (Environment Profile) will be scanned and analyzed.

CONCLUSION AND CURRENT/FUTURE WORK

In order to maximize vulnerability assessment capabilities, integration of a VRF module is a must in any deployments of Web Services. This module may addresses most known possible vulnerabilities in web servers for the purpose of problems abstraction and as a compliance auditing tool of computing environments. The proposal is considered to be a new approach in software engineering where integration, collaboration, and correlation are its semantics. Future work is directed towards addressing possible architectures of implementations methods of retrieval, and environment variables computing. Most of related work is based on methods of scanning and none are directed towards an integration approach for all SOAP-based Web Services.

REFERENCES

[1] The World Wide Web Consortium, Web Service Architecture, http://www.w3.org/TR/ws-arch/
[2] US-Computer Emergency Readiness Team (US-CERT)'s database; http://search.us-cert.gov/
[3] [7] Common Vulnerabilities and Exposures (CVE) http://www.cve.mitre.org/cve/
[4] Distributed Info. Systems, Errol Simon, McGraw-Hills
[5] National Infrastructure Advisory Council, http://www.dhs.gov/
[6] Vulnerability Disclosure Framework, http://www.dhs.gov/interweb/assetlibrary/vdwgreport.pdf