Semantic Markup for Secure Survivable Enterprise Applications

Anya Kim, Amit Khashnobish, Jim Luo, Bruce Montrose, Myong Kang
Center for High Assurance Computer Systems
Naval Research Laboratory
Washington, DC 20375
{kim, amith, luo, montrose, mkang}@itd.nrl.navy.mil

Abstract

In this abstract we present an extended abstract of our work in semantic markup of security-related information.

1. Extended Abstract

While the Internet and distributed computing bring forth fresh and exciting possibilities, they also create greater security risks, threats and obstacles. Due to the security risks associated with the Internet, most of the resources are protected with some sort of security mechanisms. Such protection can be a potential roadblock to achieve the goal of interoperability in the Semantic Web or Service-oriented architectures (SOA) that are emerging as new distributed computing paradigms.[1] Therefore, it is important to be able to markup these resources with various security-related metadata in a well-understood and consistent manner so that they can be correctly and automatically discovered, compared, and invoked according to the requestor's security requirements as well as functional requirements. Such annotation will enable dynamic discovery and invocation of resources in a secure and trusted environment.

We envision a SOA as in Figure 1. At the top level exists the enterprise application that enables us to logically compose applications from distributed tasks. At the lowest level are the web services that actually correspond to the tasks in the enterprise application. In between lies the infrastructure that supports the application logic and the querying for lower-level services.

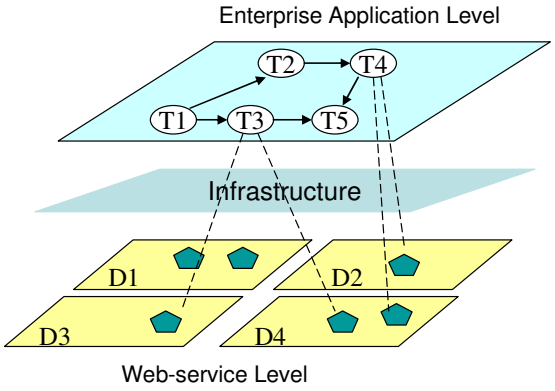

Figure 1. Service-oriented computing paradigm

Without security-related information, a web service architecture would be useless. Clients may be able to discover web services that fulfill their functionality requirements, but may not be able to access the service because they don't posses the right credentials or they use incompatible security mechanisms. Therefore matchmaking between service providers and requestors must involve not only matching of functionality, but also security requirements and capabilities. However, the current state-of-the-art does not provide any standard machine-understandable means of expressing and understanding such security-related information.

We need a means to express security requirements and capabilities of the individual services at the web services level and to express security-related information in the context of the environment at the

[1] Even though these two paradigms emphasize slightly different aspects of distributed computing, they are slowly merging. Therefore, in this paper, we use these two paradigms interchangeably.

enterprise application level, and support the query and discovery of security-related information at the infrastructure level. We provide these security features by enhancing and extending existing industry standards.

OWL (Web Ontology Language) and OWL-S (OWL for Web services) are two standards that enable specification of resources with ontologies [1, 2]. OWL provides the means to create ontologies of common domains, while OWL-S contains a set of OWL-based ontologies specifically for describing Web services. Ontology by definition is a "formal, shared conceptualization of a particular domain of interest" [3]. Ontologies are reusable, extendable, and can provide a common vernacular to understand and express complex concepts in the knowledge domain and thus are well suited to express complex security concepts.

Using security-related semantic information in the context of SOA tends to be much more complex than utilizing other semantic information. First, security descriptions are generally much more complex than functional descriptions. A security description, in general, requires concepts composed of several properties that may come from several different ontologies. For example, while a functional description of a service may be as simple as stating that it is a book buying service, a security description of the same service may state that it possesses an X.509v3 certificate for authentication purposes, who issued the certificate, its validity period, etc. Second, matchmaking of security related information is two-way, whereas functional matchmaking is one way: both the requestor and provider must have a set of security requirements and capabilities and the capabilities of one must be matched to the requirements of the other and vice versa [4].

To semantically describe security-related concepts, we developed a set of security ontologies using OWL and OWL-S that we call the NRL Security Ontology [4]. While a set of DAML Security ontologies are available [5], ours are more intuitive and capable of describing a wider range of security concepts. The NRL Security Ontology can be used to express security concepts ranging from high-level security objectives to protocols and mechanisms, credentials, security assurance levels and security algorithms in various levels of detail. For example, one can describe a security requirement at an abstract level such as 'Confidentiality' or in detail such as 'Symmetric encryption algorithm DES in CBC mode'.

At the web services level, this set of ontologies can be used to describe the security requirements and capabilities of web services so that services can specify what security requirements must be met to access the service and what security precautions it can take for the service requestor.

At the enterprise application level, BPEL4WS (Business Process Execution Language for Web Services) is the de facto industry standard to capture application logic [6]. BPEL4WS is an XML-based standard that facilitates designing, defining, implementing, and deploying enterprise applications from several distributed and autonomous software components. However, the specification does not provide a clear picture of the topological relationship of the underlying web service components, and does not address security at all. We augmented BPEL4WS with security-related markup while maintaining compliance with the specification. We extract pertinent information from the BPEL into a format that better captures the interaction between participating tasks within the application. We provide a GUI renders this format into a convenient network diagram. From the GUI, users can access the NRL Security Ontology to add security information in the context of an enterprise application.

At the infrastructure level, UDDI (Universal Description, Discovery and Integration) is an industry-standard registry that stores taxonomical descriptions of Web services [7]. As it stands, UDDI is not capable of storing or processing semantic descriptions due to its flat syntax-based classification system. We added semantic markup and query capabilities to the existing registry implementation of UDDI v3 through client-side modules that do not require any modification to the existing UDDI infrastructure [8]. These modules enable UDDI to store complex semantic markup of Web services in its data model and use this information to perform matchmaking for security-related searches. For this purpose, we provided a matching algorithm that can match a requestor's security capabilities to a service's security requirements and a requestor's security requirements to a service's security capabilities. The GUI developed for the enterprise application level can also extract ontology information from the UDDI.

The creation of a security ontology to describe security concepts and the development of tools that make use of this semantic annotation of security are only the first steps. The ontologies may need to be expanded to include additional concepts such as trust and QoS. The infrastructure components themselves may need additional security measures and must be survivable against failures or attacks. There is still much work to be done in creating a secure and survivable SOA infrastructure.

In conclusion, annotating resources with metadata enables them to be machine-understandable and facilitates automatic discovery and invocation. Since most resources on the network are protected by some sort of security mechanism, finding resources that meet one's functional requirement alone may not guarantee access to desired resources or produce desired results. Thus, annotation of resources in terms of security is just as important as annotation in terms of functionality. We are using semantic markup to dynamically integrate and compose services into secure and survivable enterprise applications in the SOA framework. This work affects every layer in the SOA architecture from security markup of an application composed of multiple services to the security annotation of individual services themselves, and the development of infrastructures to support semantic querying and matchmaking.

2. References

[1] W3C (2004). OWL Web Ontology Language Overview, available online at http://www.w3.org/TR/owl features/.

[2] DAML. OWL-S 1.1 Documentation, available online at http://www.daml.org/services/owl-s/1.1.

[3] Gill, T. Metadata and the World Wide Web. in Baca, M. ed. *Introduction to Metadata: Pathways to Digital Information*, Getty Information Institute, Los Angeles, 2000.

[4] Kim, A., Luo, J., and Kang, M. (2005). Security Ontology for Annotating Resources, in 4th *International Conference on Ontologies, Databases, and Applications of Semantics (ODBASE 2005)*, Agia Napa, Cyprus.

[5] Denker, G., Kagal, L., Finin, T., Paolucci, M. and Sycara, K. (2003). Security for DAML Web Services: Annotation and Matchmaking, in *Proc. of the 2nd International Semantic Web Conference (ISWC2003)*, Sanibel Island, Florida.

[6] Andrews, T., Curbera, F., et al. (2003). Business Process Execution Language for Web Services, version 1.1, available online at http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/

[7] UDDI Spec Technical Committee. UDDI Version 3.0.2., OASIS 2004, available online at http://uddi.org/pubs/uddi_v3.htm.

[8] Luo, J., Montrose, B, and Kang, M (2005). An Approach for Semantic Query Processing with UDDI, in *1st International Workshop on Agents, Web Services, and Ontologies Merging (AWeSOMe'05)*, Agia Napa, Cyprus.