Security Issues for the Semantic Web

Dr. Bhavani Thuraisingham

Program Director
Data and Applications Security
The National Science Foundation
Arlington, VA
On leave from The MITRE Corporation
Bedford, MA

Abstract

This paper first describes the developments in semantic web and then provides an overview of secure semantic web. In particular XML security, RDF security, and secure information integration and trust on the semantic web are discussed. Finally directions for research on secure semantic web are provided.

1. Introduction

Recent developments in information systems technologies have resulted in computerizing many applications in various business areas. Data has become a critical resource in many organizations, and therefore, efficient access to data, sharing the data, extracting information from the data, and making use of the information has become an urgent need. As a result, there have been many efforts on not only integrating the various data sources scattered across several sites, but extracting information from these databases in the form of patterns and trends has also become important. These data sources may be databases managed by database management systems, or they could be data warehoused in a repository from multiple data sources.

The advent of the World Wide Web (WWW) in the mid 1990s has resulted in even greater demand for managing data, information and knowledge effectively. There is now so much ~~of~~ data on the web that managing it with conventional tools is becoming almost ~~impossibility~~impossible. New tools and techniques are needed to effectively manage this data. Therefore, to provide ~~the~~ interoperability as well as warehousing between the multiple data sources and systems, and to extract information from the databases and warehouses on the web, various tools are being developed.

In [THUR02a] we provided an overview of some directions in data and applications security research. In this paper we focus on one of the topics and that is securing the semantic web. While the current web technologies facilitate the integration of information from a syntactic point of view, there is still a lot to be done to integrate the semantics of various systems and applications. That is, current web technologies depend a lot on the human-in-the-loop for information integration. Tim Berners Lee, the father of WWW, realized the inadequacies of current web technologies and subsequently strived to make the web more intelligent. His goal was to have a web that will essentially alleviate humans from the burden of having to integrate disparate information sources as well as to carry out extensive searches. He then came to the conclusion that one needs machine understandable web pages and the use of ontologies for information integration. This resulted in the notion of the semantic web [LEE01].

A semantic web can be thought of as a web that is highly intelligent and sophisticated and one needs little or no human intervention to carry out tasks such as scheduling appointments, coordinating activities, searching for complex documents as well as integrating disparate databases and information systems. While much progress has been made toward developing such an intelligent web there is still a lot to be done. For

Proceedings of the 27th Annual International Computer Software and Applications Conference (COMPSAC'03)
0730-3157/03 $ 17.00 © 2003 IEEE

example, technologies such as ontology matching, intelligent agents, and markup languages are contributing a lot toward developing the semantic web. Nevertheless one still needs the human to make decisions and take actions.

Recently there have been many developments on the semantic web (see for example, [THUR02b], [THUR03]). However it is also very important that the semantic web be secure. That is, the components that constitute the semantic web have to be secure. In addition, the components have to be integrated securely. The components include XML, RDF and Ontologies. In addition, we need secure information integration. We also need to examine trust issues for the semantic web.

This paper first provides an overview of the semantic web. In particular, Tim Berners Lee's description of the various layers that would comprise the semantic web. It then focuses on security issues for the semantic web. In particular, XML security, RDF security, and secure information integration will be discussed. The paper is concluded with a discussion of the directions.

2. Overview of the Semantic Web

Tim Berners Lee has specified various layers for the semantic web. At the lowest level one has the protocols for communication including TCP/IP (Transmission Control Protocol/Internet protocol), HTTP (Hypertext Transfer Protocol) and SSL (Secure Socket Layer). The next level is the XML (eXtensible Markup Language) layer that also includes XML schemas. The next level is the RDF (Resource Description Framework) layer. Next come the Ontologies and Interoperability layer. Finally at the highest-level one has the Trust Management layer. Each of the layers is discussed below.

TCP/IP, SSL and HTTP are the protocols for data transmission. They are built on top of more basic communication layers. With these protocols one can transmit the web pages over the Internet. At this level one does not deal with syntax or the semantics of the documents. Then comes the XML and XML Schemas layer. XML is the standard representation language for document exchange. For example, if a document is not marked-up, then each machine may display the document in its own way. This makes document exchange extremely difficult. XML is a markup language that follows certain rules and if all documents are marked up using XML then there is uniform representation and presentation of documents. This is one of the significant developments of the World Wide Web. Without some form of common representation of documents, it is impossible to have any sort of communication on the web. XML schemas essentially describe the structure of the XML documents. Both XML and XML schemas are the invention of Tim Berners Lee and his consortium called the World Wide Web Consortium (W3C), which was formed in the mid 1990s (see also [LAUR00] and [THUR02]).

Now XML focuses only on the syntax of the document. A document could have different interpretations at different sites. This is a major issue for integrating information seamlessly across the web. In order to overcome this significant limitation, W3C started discussions on a language called RDF in the late 1990s. RDF essentially uses XML syntax but has support to express semantics. One needs to use RDF for integrating and exchanging information in a meaningful way on the web. While XML has received widespread acceptance, RDF is only now beginning to get acceptance. So while XML documents are exchanged over protocols such as TCP/IP, HTTP and SSL, RDF documents are built using XML.

Next layer is the Ontologies and Interoperability layer. Now RDF is only a specification language for expressing syntax and semantics. The question is what entities do we need to specify? How can the community accept common definitions? To solve this issue, various communities such as the medical community, financial community, defense community, and even entertainment community have come up with what are called ontologies. One could use ontologies to describe the various wines of the world or the different types of aircraft used by the United States Air Force. Ontologies can also be used to specify various diseases or financial entities. Once a community has developed ontologies, the community has to publish these ontologies on the web. The idea is that everyone interested in ontologies of a community use the ontologies defined by the community. Now, within a

community there could be different factions and each faction could come up with its own ontologies. For example the American Medical Association could come up with its ontologies for diseases while the British Medical Association could come up with its own ontologies. This poses a challenge as the system and in this case the semantic web has to examine the ontologies and decide how to develop some common ontologies. While the goal is for the British and American communities to agree and come up with common ontologies, in the real-world differences do exist. The next question is what do ontologies do for the web. Now, using these ontologies different groups can communicate information. That is, ontologies facilitate information exchange and integration. Ontologies are used by web services so that the web can provide semantic web services to the humans. Ontologies may be specified using RDF syntax.

The final layer is logic, proof and trust. The idea here is how do you trust the information on the web? Obviously it depends on whom it comes from. How do you carry out trust negotiation? That is interested parties have to communicate with each other and determine how to trust each other and how to trust the information obtained on the web. Closely related to trust issues is security and will be discussed later on. Logic-based approaches and proof theories are being examined for enforcing trust on the semantic web.

3. Security Issues for the Semantic Web

3.1 Overview of Security issues

We first provide an overview of the security issues and then discuss some details on XML security, RDF security and secure information integration, which are components of the secure semantic web.

As stated earlier, logic, proof and trust are at the highest layers of the semantic web. That is, how can we trust the information that the web gives us? Closely related to trust is security. However security cannot be considered in isolation. That is, there is no one layer that should focus on security. Security cuts across all layers and this is a challenge.

For example, consider the lowest layer. One needs secure TCP/IP, secure sockets, and secure HTTP. There are now security protocols for these various lower layer protocols. One needs end-to-end security. That is, one cannot just have secure TCP/IP built on untrusted communication layers. That is, we need network security. Next layer is XML and XML schemas. One needs secure XML. That is, access must be controlled to various portions of the document for reading, browsing and modifications. There is research on securing XML and XML schemas. The next step is securing RDF. Now with RDF not only do we need secure XML, we also need security for the interpretations and semantics. For example under certain context, portions of the document may be Unclassified while under certain other context the document may be Classified. As an example one could declassify an RDF document, once the war is over. Lot of work has been carried out security constraints processing for relational databases. One needs to determine whether these results could be applied for the semantic web (see [THUR95]).

Once XML and RDF have been secured the next step is to examine security for ontologies and interoperation. That is, ontologies may have security levels attached to them. Certain parts of the ontologies could be Secret while certain other parts may be Unclassified. The challenge is how does one use these ontologies for secure information integration? Researchers have done some work on the secure interoperability of databases. We need to revisit this research and then determine what else needs to be done so that the information on the web can be managed, integrated and exchanged securely.

Closely related to security is privacy. That is, certain portions of the document may be private while certain other portions may be public or semi-private. Privacy has received a lot of attention recently partly due to national security concerns. Privacy for the semantic web may be a critical issue, That is, how does one take advantage of the semantic web and still maintain privacy and sometimes anonymity.

We also need to examine the inference problem for the semantic web. Inference is the process of posing queries and deducing new information. It becomes a problem when the deduced information is something the user is unauthorized to know.

With the semantic web, and especially with data mining tools, one can make all kinds of inferences. That is the semantic web exacerbates the inference problem (see [THUR98]). Recently there has been some research on controlling unauthorized inferences on the semantic web. We need to continue with such research (see for example, [FARK03]).

Security should not be an afterthought. We have often heard that one needs to insert security into the system right from the beginning. Similarly security cannot be an afterthought for the semantic web. However, we cannot also make the system inefficient if we must guarantee one hundred percent security at all times. What is needed is a flexible security policy. During some situations we may need one hundred percent security while during some other situations say thirty percent security (whatever that means) may be sufficient.

3.2 XML Security

Various research efforts have been reported on XML security (see for example, [BERT02]). We briefly discuss some of the key points.

XML documents have graph structures. The main challenge is whether to give access to entire XML documents or parts of the documents. Bertino et al have developed authorization models for XML. They have focused on access control policies as well as dissemination policies. They also considered push and pull architectures. They specified the policies in XML. The policy specification contains information on which users can access which portions of the documents. Users may have privileges associated with them depending on their roles. Furthermore, privileges may cascade down the documents. Privileges include read, write, append, distribute, and browse. For example, if a user has access to the root of a document then should his access, say read, propagate to all the descendants or the immediate children? In [BERT02] algorithms for access control as well as computing views of the results are also presented. In addition, architectures for securing XML documents are also discussed.

In [BERT03] the authors go further and describe how XML documents may be published on the web. The idea is for owners to publish documents, subjects to request access to the documents and publishers to give the subjects the views of the documents they are authorized to see. The idea is for the publishers to be untrusted. Essentially the owner specifies the access control policies. The publisher will then enforce the access control policies. Encryption algorithms as well as Merkel hash are used to ensure that the publisher is untrusted. Bertino et al provide the authenticity and completeness of the results computed by the publisher and sent to the subject.

W3C (World Wide Web Consortium) is also specifying standards for XML security. The XML security project (see [XML1]) is focusing on providing the implementation of security standards for XML. The focus is on XML-Signature Syntax and processing, XML-Encryption Syntax and Processing and XML Key Management. W3C also has a number of working groups including XML Signature working group (see [XML2]) and XML encryption working group (see [XML3]). While the standards are focusing on what can be implemented in the near-term lot of research is needed on securing XML documents. The work reported in [BERT02] is a good start.

3.3. RDF Security

RDF is the foundations of the semantic web. While XML is limited in providing machine understandable documents, RDF handles this limitation. As a result, RDF provides better support for interoperability as well as searching and cataloging. It also describes contents of documents as well as relationships between various entities in the document. While XML provides syntax and notations, RDF supplements this by providing semantic information in a standardized way.

The basic RDF model has three types: they are resources, properties and statements. Resource is anything described by RDF expressions. It could be a web page or a collection of pages. Property is a specific attribute used to describe a resource. RDF statements are resources together with a named property plus the value of the property. Statement components are subject, predicate and object. So for example, if we have a sentence of the form "John is the creator of xxx", then xxx is the subject or resource, Property or predicate is "Creator" and object or literal is "John". There are RDF diagrams very much like say ER diagrams or object diagrams to represent statements.

There are various aspects specific to RDF syntax. Iis very important that the intended interpretation be used for RDF sentences. This is accomplished by RDF schemas. Schema is sort of a dictionary and has interpretations of various terms used in sentences. RDF and XML namespaces to resolve conflicts in semantics.

More advanced concepts in RDF include the container model and statements about statements. The container model has three types of container objects and they are Bag, Sequence, and Alternative. A bag is an unordered list of resources or literals. It is used to mean that a property has multiple values but the order is not important. A sequence is a list of ordered resources. Here the order is important. Alternative is a list of resources that represent alternatives for the value of a property. Various tutorials in RDF describe the syntax of containers in more detail.

RDF also provides support for making statements about other statements. For example, with this facility one can make statements of the form "The statement A is false" where A is the statement "John is the creator of XXX". Again one can use object-like diagrams to represent containers and statements about statements (see [RDF])

Now to make the semantic web secure, we need to ensure that RDF documents are secure. With RDF we need to ensure that security is preserved at the semantic level. The issues include what the security implications of the concepts resource, properties and statements. That is, how is access control ensured? How can statements, properties and statements be protected? How can one provide access control at a finer grain of granularity? What are the security properties of the container model? How can bags, lists and alternatives be protected? Can we specify security policies in RDF? How can we resolve semantic inconsistencies for the policies? How can we express security constraints in RDF? What are the security implications of statements about statements? How can we protect RDF schemas? These are difficult questions and we need to start research to provide answers. XML security is just the beginning. Securing RDF is much more challenging.

3.4 Secure Information Interoperability

Information is everywhere on the web. Information is essentially data that makes sense. The database community has been working on database integration for some decades. They encountered many challenges including interoperability of heterogeneous data sources. They used schemas to integrate the various databases (see [SHET90]).

Now with the web, one needs to integrate the diverse and disparate data sources. The data may not be in databases. It could be in files both structured and unstructured. Data could be in the form of tables or in the form of text, images, audio and video. Essentially one needs the semantic web services to integrate the information on the web.

The challenge for security researchers is how does one integrate the information securely? For example, in [THUR94] and [THUR97] the schema integration work of Sheth and Larson was extended for security policies. That is, different sites have security policies and these policies have to be integrated to provide a policy for the federated databases system. One needs to examine these issues for the semantic web. Each node on the web may have its own policy. Is it feasible to have a common policy for a community on the web? Do we need a tight integration of the policies or do we focus on dynamic policy integration?

Ontologies are playing a major role in information integration on the web. How can ontologies play a role in secure information integration? How do we provide access control for ontologies? Should ontologies incorporate security policies? Do we have ontologies for specifying the security policies? How can we use some of the ideas discussed in [BERT03] to integrate information securely on the web? That is, what sort of encryption schemes do we need? How do we minimize the trust placed on information integrators on the web? We need a research program to address many of these challenges.

4. Summary and Directions

This paper has provided an overview of the semantic web and discussed security issues. We first discussed the layered framework of the semantic web proposed by Tim Berners Lee. Next we discussed security issues. We argued that security must cut across all the layers. Furthermore, we need to integrate the information across the layers securely. Next we provided some more

details on XML security, RDF security and secure information integration. If the semantic web is to be secure we need all of its components to be secure. We also need secure information integration.

There is a lot of research that needs to be done. We need to continue with the research on XML security. We must start examining security for RDF. This is much more difficult as RDF incorporates semantics. We need to examine the work on security constraints processing and see if we can apply some of the ideas for RDF security. Finally we need to examine the role of ontologies for secure information integration. We have to address some hard questions such as how do we integrate security policies on the semantic web? How can we incorporate policies into ontologies? We also cannot forget about privacy and trust on the semantic web. That is, we need to protect the privacy of individuals and at the same time ensure that the individuals have the information they need to carry out their functions. We also need to combine security research with privacy research. Finally we need to formalize the notions of trust and examine ways to negotiate trust on the semantic web. We have a good start and are well on our way to building the semantic web. We cannot forget about security and privacy. Security must be considered at the beginning and not as an afterthought.

Disclaimer: The views and conclusions expressed in this paper are those of the author and do not reflect the policies of the National Science Foundation or of the MITRE Corporation.

Acknowledgements: I thank the National Science Foundation and the MITRE Corporation for their support of my research on security issues for the semantic web.

References

[BERT02] Bertino, E., et al, Access Control for XML Documents, Data and Knowledge Engineering, 2002.

[BERT03] Bertino, E. et al, Secure Third Party Publication of XML Documents, To appear in IEEE Transactions on Knowledge and Data Engineering.

[FARK03] Farkas, C., Inference Problem for the Semantic Web, Proceedings of the IFIP Conference on Data and Applications Security, Colorado, August 2003.

[LEE01] Berners Lee, T., et al., The Semantic Web, Scientific American, May 2001.

[LAUR00] St Laurent, S., XML, McGraw Hill, 2000

[RDF] Resource Description Framework, www.w3c.org

[SHET90] Sheth A. and J. Larson, Federated Database Systems, ACM Computing Surveys, September 1990.

[THUR94] Security Issues for Federated Database Systems, Computers and Security, 1994.

[THUR95] Thuraisingham B. and W. Ford, Security Constraint Processing in a Distributed Database Management System, IEEE Transactions on Knowledge and Data Engineering, 1995.

[THUR97] Thuraisingham, B., Data Management Systems Evolution and Interoperation, CRC Press, 1997.

[THUR98] Thuraisingham, B., "Data Mining: Technologies, Techniques, Tools and Trends," CRC Press, 1998.

[THUR02a] Thuraisingham, B., "Data and Applications Security" Developments and Directions," Proceedings IEEE COMPSAC 2002.

[THUR02b] Thuraisingham, B., "XML, Databases and the Semantic Web" CRC Press, FL, 2001.

[THUR03] Thuraisingham, B., "The Semantic Web," Encyclopedia of Human Computer Interaction, (Ed: W. Bainbridge, Berkshire Publishers), 2003.

[XML1] http://xml.apache.org/security/

[XML2] http://www.w3.org/Signature/

[XML3] http://www.w3.org/Encryption/2001/