1-1-2002

Making Agents Secure on the Semantic Web

Csilla Farkas
University of South Carolina - Columbia, farkas@cse.sc.edu

Michael N. Huhns
University of South Carolina - Columbia, huhns@sc.edu

Publication Info

Published in *IEEE Internet Computing*, Volume 6, Issue 6, 2002, pages 76-79.
http://ieeexplore.ieee.org/servlet/opac?punumber=4236

Making Agents Secure on the Semantic Web

Csilla Farkas • *University of South Carolina* • *farkas@cse.sc.edu*
Michael N. Huhns • *University of South Carolina* • *huhns@sc.edu*

Are you one of those people who never shops on the Web for fear of fraud? Does the thought of online banking make you antsy? Do you get nervous when talk of digital government arises? If so, you're not as paranoid as your friends say — some of your concerns are indeed well founded. With the rapid development of Web technologies and applications built on these technologies, new security risks have emerged. Traditional security models do not provide adequate protection in this dynamic and open environment. Fortunately, as we'll discuss, significant efforts are under way that should make Web services secure.

Securing the Enterprise

Nowadays, an individual still might be able to avoid going online, but any company that wants to compete in the global economy needs a Web presence. That entails risks. To balance Internet access needs with security concerns, the business world needs technologies that provide security with high assurance.

Security leaks can cost a company its competitive advantage, and leaks are rarely accidental. Industrial espionage has been around for ages. In the 16th century, for instance, Venetian merchants used poison and bribery to gain market share. As emerging Web services become preeminent in e-commerce, advanced technologies, such as software agents and ontologies, will allow more subtle attacks.

Security is not solely a problem for human organizations. In a real scenario reminiscent of the cartoon *Spy vs. Spy,* an assassin agent recently connected itself to the JADE (Java Agent Development Framework) agent platform in Italy, killing all of the agents it found there. Although not as serious as it might appear — because the agents in Italy were there for research purposes — the episode nevertheless shows that agents need security. As agents adopt more mission-critical roles within enterprises, this concern will grow in importance.

Web Services and the Semantic Web

The World Wide Web was designed for humans. The envisioned Semantic Web is geared toward agents as well. Semantic constructs — such as ontologies represented in the DARPA Agent Markup Language (DAML), the Resource Description Framework (RDF), and XML — will let agents, as well as people, *understand* a Web page's content.[1,2]

The Semantic Web's success will depend on the implementation and use of Web services, which will likely be agent-based in the future. Using intelligent collaborations, agents can achieve global optimization while adapting to local requirements.[3] This approach will let agents use the large amount of information available over the Web — a task that is beyond human processing power — but this enhanced processing power can be a double-edged sword. Because malicious users and their agents can disclose sensitive information or sabotage the information of others, agents and supporting technologies need to be secure and reliable for safe Web services.

A Secure Semantic Web

Agent technologies were designed with a focus on interoperability, distributed problem solving, and cooperation. Multiagent systems were

 1089-7801/02/$17.00 ©2002 IEEE

intended to be responsive to open environments, such as the Internet, to capitalize on cooperative interactions. But this readiness to interact leaves them vulnerable to agents with harmful intentions. A malicious agent could "kill" an agent that was responsible for selling goods, for example, and by pretend to be that agent, could begin taking sales orders and accepting payments. Such potential for abuse has prompted recent efforts to develop security features for agent-based systems.

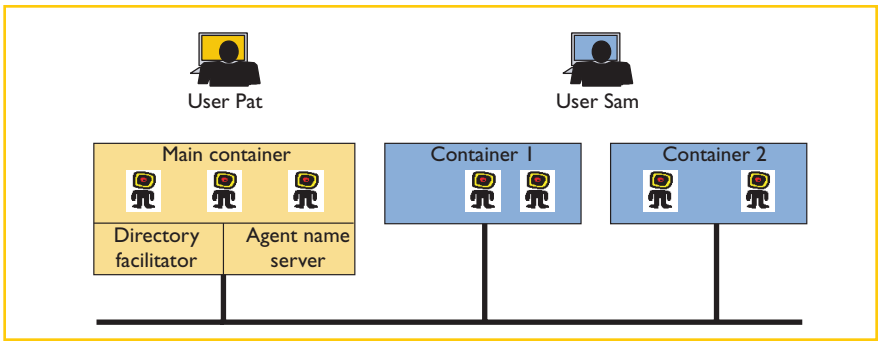

Figure 1. Multiuser JADE security scenario. User Pat owns the main container, two of its agents, and one of the agents in Container-2. User Sam owns the other containers and agents. The users must be authenticated on the JADE platform by providing a valid username and password. A JADE policy, along with the permissions of the owners, determines which actions the agents can perform.

Secure Multiagent Systems

Safeguarding multiagent systems requires the development of secure communication protocols, access control models for agents, methods for delegating agent privileges, and distributed trust management. In addition, agent management systems and directory services must be protected from being compromised or destroyed. Developers are securing agent platforms by providing mechanisms for authenticating and authorizing individual agents and platform components.[4]

The approach used with JADE is typical and interesting.[5] It is principal-based, meaning that it relies on Java security policies and mechanisms for human users. Because a JADE agent platform can be distributed across several hosts and can support the interactions of autonomous entities, each component must belong to some user who is responsible for its actions. Hence, a JADE platform must know all users and be convinced of their authentication. Figure 1 shows a multiuser JADE scenario, which has two authenticated owners for all components.

On this platform, each agent has an identity certificate that is digitally signed by a certification authority. The certificate serves to unlock privileges for the agent. Using a delegation mechanism, an agent can also borrow credentials from other agents to get permission to perform other actions. Signed delegation certificates validate the additional credentials.

Such traditional user- and principal-based security systems are likely to be insufficient to secure agent-based systems. Agents can play different roles in different platforms, dynamically change their access requirements, and act on behalf of users with different access privileges. Without relying on a centralized trust management system, agents must be able to decide whether or not to trust another agent or platform. Policies must therefore be developed that focus specifically on agent environments.

Agent Security

Although security needs for multiagent applications have drawn attention recently, few applications deploy security measures.[3,6] Those that do focus either on agent communications or the needs of mobile agents,[4,6] paying less attention to secure multiagent platforms, privilege propagation, and trust management.

Clearly, communication with other agents is a crucial aspect of agent execution and cooperation. Ensuring safe execution first means providing confidential and integrity-preserving communication among the agents.

The use of mobile agents poses two fundamental security requirements:

- Protecting an agent execution environment from a malicious agent.
- Protecting an agent from a malicious platform.

Yet, the threat of agents being monitored, tested on fake data, or supplied with malicious code has not received sufficient attention. Agents could easily be corrupted or destroyed, their Web usage monitored and analyzed, and their communications disclosed and distorted. Researchers must develop and implement techniques to evaluate multiagent platforms' security needs — such as authentication, access control, and inferences — to mitigate such threats.

Stealth Attacks

It is not always necessary to compromise a computer system to gain access to unauthorized information. In addition to making direct attacks, a malicious user might associate and correlate publicly available information to deduce vulnerabilities and secrets. Current Web technologies primarily support human consumption, thus limiting the power of data analysis, but the support for interoperation and machine-enabled information processing increases the danger. Intelligent agents, enabled to access and process large amounts of Web data, will be able to discover information that might be confidential.

Consider a scenario in which a water treatment facility wants to keep its maintenance schedule inaccessible

to the public to prevent tampering. In the same region, a high school student is performing research for a science fair project. To test her hypothesis that maintenance of the water reservoir affects water quality, she takes water samples from the area and uses the water treatment facility's laboratory to analyze the samples. Her results and the descriptions of her experiments, including the sample time and treatment, are posted on her Web site, enabling the facility's schedule to be deduced.

Such security breeches might not always be obvious to people, because different sites might use different vocabularies to describe their information, but agents with ontology translation capabilities might be able to infer disallowed information. To ensure secure Semantic Web applications, the research community will need to provide answers to:

- How can we measure the security of a Web-based information system?
- How can we assess the security threat from large-scale, distributed Web inferences?
- What are the fundamental differences between securing software agents and securing human users?
- What inferences can a malicious user make by observing an agent's responses or monitoring an agent's Web usage?
- How can agents make decisions on the trustworthiness of Web sites and other agents, and detect probing attacks?
- How can access to ontologies be controlled to prevent undesired inferences while supporting agents' cooperation?

Although indirect attacks might not be as obvious as "killing" an agent, the consequences can be just as serious. Silently monitoring an organization's marketing agents might enable its competition to steal customers and gain market share. Addressing both direct and indirect attacks on agents and their platforms is crucial to providing a secure Web environment.

Securing the Supporting Technology

To provide security for (Semantic) Web services, security models and tools for the underlying technologies, such as XML, RDF, DAML-OIL, need to be developed. Security requirements for multiagent systems are driven by functionality, collaboration, and organizational needs. A policy language to express security requirements and techniques to enforce the policy at the level of supporting technologies need to be developed.

Addressing both direct and indirect attacks on agents and their platforms is crucial to providing a secure Web environment.

Ongoing research efforts address the problem of controlling access to XML data: Industry experts focus mainly on developing technologies to enforce security restrictions, such as XML signatures and encryption, while academic researchers develop design principles and access-control models.[7,8]

The Organization for Advancement of Structured Information Standards (www.oasis-open.org) is defining the XML-based Security Assertion Markup Language, for example, to standardize the exchange of authentication and authorization information. One of the major design goals with SAML is *single sign-on* — the ability of a user to authenticate in one domain and use resources in other domains without reauthenticating. This standard will be an important protocol for agents as they engage and interact with resources and other agents in various domains. With it, a security administrator can express advanced security requirements, such as time- or event-based restrictions.

Access Control

Access control forms the first line of defense against misuse. We need flexible models for Web access control that support fine-grained data granularity, accommodate a wide range of policies, are suitable for dynamic, decentralized, and open environments, and are scalable. In addition, they must preserve the semantic consistency of data and limit illegal inferences. Current techniques address some of these concerns, but no full solution is yet available.

For example, some of the XML access control models require that sensitivity of data items increase downward in an XML tree. Users can be prevented from seeing the lower nodes of the tree. While this model might be enforceable during the design of a new XML document, it might not be applicable to existing ones.

Other models, which allow mixed sensitivity along a path in the XML document — for example, a low-sensitivity tag might be under a high-sensitivity tag — provide confidentiality by not releasing the value of the sensitive tags. They either release "blank" values as place holders for the real values, or delete the sensitive tags and values and link the lower-level tags to their nearest permitted tag. The first solution creates an inference channel, allowing an unauthorized user to infer the existence of a disallowed data item. The second solution does not reveal the existence of the data item, but might violate semantic consistency of the tags that

were originally linked to the disallowed tags.

The Inference Problem

Access-control models focus on security requirements of direct data accesses. However, they fail to address indirect accesses based on the underlying data semantics. While several research papers and prototypes explore the problem of RDF inferences, none considers the possible security implications, let alone recommends ways to safeguard sensitive information.

The *inference problem*, as defined by database researchers, is where sensitive information is disclosed by combining nonsensitive data with metadata (such as database dependencies and integrity constraints). Its avoidance lies in how to express a user's knowledge (metadata) and how to trace possible collaborations among malicious users.

Development of the Semantic Web reopens the inference problem from a new perspective. We can hardly assume that the Web user — or even the person whose data is stored on the Web — knows all available information. Sensitive data about users could quite possibly be available at a site unknown to them.

In one of our favorite homework assignments, for example, we ask students to find a professor's home phone number, which is listed under a different name in the local telephone directory. The number is not posted anywhere, and all listings on the Web have been tracked to ensure that the number is confidential. By combining information about the professor's other family members, however, students eventually can infer the correct number.

In a Web environment, it is difficult to detect data replication with inconsistent security classifications. It is even more difficult when the use of ontologies makes it possible to find related data stored at other sites. Ontologies also support semantic correlation between related data to derive new information (similar to traditional database inferences in an open environment). The derived information might be sensitive and should not be derivable from released, nonsensitive data.

Conclusions

Agents were designed to collaborate and share information. While highly desirable for interoperability, this feature is scary from the security perspective. Illegal inferences, supported by Semantic Web technology and ontologies, might enable users to access unauthorized information. In addition to semantic associations and replicated data with different sensitivity, malicious agents could also exploit statistical inferences. Although each agent in a system might behave in a desired and secure way, their combined knowledge could be used to disclose sensitive data. The research community must therefore develop and implement techniques that allow control over released data.

The research community must develop and implement techniques that allow control over released data.

To answer the questions related to information availability (scalability), data correctness (integrity), and access control in the presence of illegal inferences and undesired collaborations (confidentiality), researchers in Semantic Web technologies (XML, RDF, DAML, and multiagent systems) and information system security need to collaborate. Indeed, given the Web's openness, dynamic nature, and diverse user population, developing secure Web services will require the collaboration of experts in different fields from both industry and academia. In turn, the "Intelligent Web" of the future will facilitate unheard-of support for collaborations and information management.

Acknowledgements

The US National Science Foundation supported this work under grants number IIS-0083362 and IIS-0112874.

References

1. T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific Am.*, vol. 284, no. 5, May 2001, pp. 34–43.
2. J. Heflin and J.A. Hendler, "Dynamic Ontologies on the Web," *Proc. American Association for Artificial Intelligence Conf. (AAAI)*, AAAI Press, 2000, pp. 443-449.
3. M.N. Huhns, "Agents as Web Services," *IEEE Internet Computing*, Vol. 6, No. 4, Jul.-Aug. 2002, pp. 93–95.
4. J.J. Tan, L. Titkov, and C. Neophytou, "Securing Multi-Agent Platform Communication," *Working Notes Second Int'l Workshop on Security of Mobile Multiagent Systems*, 2002, pp. 66–72; www.dfki.de/~kuf/semas/semas-2002/WorkingNotes/semas-2002.zip.
5. G. Vitaglione, "*Jade Tutorial: Security Administrator Guide*," Sept. 2002, http://sharon.cselt.it/projects/jade/doc/tutorials/SecurityAdminGuid.pdf.
6. V. Roth, "Empowering Mobile Software Agents," *Proc. 6th IEEE Mobile Agents Conf.*, IEEE Computer Soc. Press, Los Alamitos, Calif., 2002, pp. 238–244.
7. E. Bertino et al., "Specifying and Enforcing Access Control Policies for XML Document Sources," *World Wide Web J.*, vol. 3, no. 3, 2000, pp. 139–151.
8. B. Dournaee, *XML Security*, McGraw-Hill, New York, 2002.

Csilla Farkas is an assistant professor of computer science and engineering at the University of South Carolina, where she teaches and conducts research in information security.

Michael N. Huhns is a professor of computer science and engineering at the University of South Carolina, where he also directs the Center for Information Technology.