Chapter IX
Protecting ASP.NET Web Services

Konstantin Beznosov
University of British Columbia, Canada

ABSTRACT

This chapter reports on our experience designing and implementing an architecture for protecting enterprise-grade Web service applications hosted by ASP.NET. Security mechanisms of Microsoft ASP.NET container—a popular hosting environment for Web services—having limited scalability, flexibility, and extensibility. They are therefore inadequate for hosting enterprise-scale applications that need to be protected according to diverse or complex application-specific security policies. To overcome the limitations of ASP.NET security, we developed a flexible and extensible protection architecture. Deployed in a real-world security solution at a financial organization, the architecture enables integration of ASP.NET into the organizational security infrastructure with reduced effort on the part of Web Service developers. Throughout this report, we discuss our design decisions, suggest best practices for constructing flexible and extensible authentication and authorization logic for Web Services, and share lessons learned.

INTRODUCTION

Although Web Services security standards and technologies are close to being ready for wide implementation, there is little feedback in the literature from those who are starting to use these building blocks to make Web Service applications fully secured and trusted. This chapter provides such feedback by reporting on our experience of designing an architecture for protecting enterprise-grade distributed applications hosted by the popular Web Services container, ASP.NET. The chapter also contributes the best practices on constructing flexible and extensible authentication and authorization logic for Web Services by employing Resource Access Decision (RAD)
Protecting ASP.NET Web Services

(Beznosov, Deng, Blakley, Burt, & Barkley, 1999; Beznosov, Espinal, & Deng, 2000; Object Management Group [OMG], 2001), and Attribute Function (AF) (Beznosov, 2002a) architectural styles. We implemented our architecture in just over 4 KLOCs of C#.

ARCHITECTURE OVERVIEW

ASP.NET container is a popular hosting environment for Web Services built and run on Microsoft Windows and .NET platforms. However, the ASP.NET security architecture (Microsoft, 2001, 2002), as provided “out-of-the-box,” lacks sufficient flexibility and extensibility to be adequate for enterprise applications (Beznosov, 2002b). As we describe in Hartman, Flinn, Beznosov, and Kawamoto (2003), ASP.NET supports limited authentication and group/user-based authorization, both being bound to Microsoft proprietary technologies (Windows domains and Passport). If a Web Service application needs to be protected via a third-party authentication or authorization services available in the enterprise security infrastructure, the real-world developers have two options. The first is to develop home-grown container security extensions, which are hard for average application developers to get right. The second is to program the security logic into the Web Service business logic, making the resulting application expensive to maintain and modify. In both cases, the development of security-specific components by average application developers is commonly believed to result in high vulnerability rates, due to security-related bugs that are hard to avoid and catch.

Due to its flexibility and extensibility, our protection architecture enables integration of ASP.NET into the organizational security infrastructure with reduced effort on the part of Web Service developers. The architecture is flexible because it allows configuring machine-wide authentication and authorization functions, and overriding them for a subtree of the Web Services (up to an individual application) in the directory-based ASP.NET hierarchy. Its extensibility is revealed through the support of a wide variety of authentication and authorization (A&A) logic, provided that the logic can be translated into a .NET class or accessed (possibly via a proxy) through a predefined .NET API. Furthermore, one can reuse instances of such logic by combining authorization decisions from them according to predefined or custom rules.

We achieved these by:

1. separating custom SOAP (World Wide Web Consortium [W3C], 2002a, 2002b, 2002c, 2002d) extension modules (which act as ASP.NET-specific A&A enforcement logic) from the A&A decision logic;
2. following the RAD architecture style (Beznosov et al., 1999; Beznosov et al., 2000; OMG, 2001), which makes customization of access control decision logic easier and avoids the need for a generic policy evaluation engine;
3. taking advantage of the extensibility, inheritance, and caching features of ASP.NET web.config configuration mechanism; and
4. separating the logic of retrieving attributes of Web Service implementations from the authorization and business logic by following the AF approach (Beznosov, 2002a).

In the next section, we discuss the requirements that drove the design, and present the architecture. To illustrate the architecture’s capabilities, we also present two examples of policies and configurations for supporting these policies. Intertwined with the requirements and the architecture descriptions in the preceding two sections, our reflections on designing the architecture, and the lessons we learned, are mentioned in the discussion and summarized at the end of the chapter.
Related Content

Applied Cryptography in E-mail Services and Web Services
[www.igi-global.com/chapter/applied-cryptography-mail-services-web/46240?camid=4v1a](www.igi-global.com/chapter/applied-cryptography-mail-services-web/46240?camid=4v1a)

A More Secure Image Hiding Scheme Using Pixel Adjustment and Genetic Algorithm
[www.igi-global.com/article/a-more-secure-image-hiding-scheme-using-pixel-adjustment-and-genetic-algorithm/95139?camid=4v1a](www.igi-global.com/article/a-more-secure-image-hiding-scheme-using-pixel-adjustment-and-genetic-algorithm/95139?camid=4v1a)

Analyzing Information Security Goals
[www.igi-global.com/chapter/analyzing-information-security-goals/65764?camid=4v1a](www.igi-global.com/chapter/analyzing-information-security-goals/65764?camid=4v1a)

E-Mail Worm Detection Using Data Mining
[www.igi-global.com/article/mail-worm-detection-using-data/2470?camid=4v1a](www.igi-global.com/article/mail-worm-detection-using-data/2470?camid=4v1a)