Security Considerations in Orientation Project

Marianna Sipos

John von Neumann Faculty of Informatics, Budapest Tech
Bécsi út 96/B, H-1034 Budapest, Hungary
sipos.marianna@nik.bmf.hu

Abstract: There is a research in John von Neumann Faculty of Informatics, Budapest Tech about the indoor orientation and guidance. The security decisions of project are based on security principles and on the tree tier development model. This contribution explains the Visual Studio ASP.NET framework services to make our Web application foolproof.

Keywords: security, principles, Web application, Web server, application server, tree tier model, deployment model, process.

1 Introduction

There is a research in John von Neumann Faculty of Informatics, Budapest Tech about the indoor orientation and guidance. The research is divided into five parts. This parts are: 1) How to make connections and keep connections with different devices. 2) Orientation within doors and within buildings. 3) Help for handicapped and disabled people in orientation. 4) Information storage and pass 5) Security decisions.

In the research are working 3 colleagues from the Softwaretechnology Institute 2 PhD student, 1 student of “Politehnica” University of Timisoara with SOCRATES/ERASMUS scholarship and 3 students of Neumann Faculty. The next documentations are published from the project: Marianna Sipos, Orientation and Guidance within Doors [1], Imola Szemes, Design and Implementation of an Indoor Guidance and Information System Using .NET and SVG [2]. The teams are working independent, but with the full system. The security decisions team work has the most tightly contact to other teams. Their decisions influence most the work of other teams.

Students can participate in the research as research assistants and can publish their results as student research work or as dissertation. They work on a small section of the chosen part.

The next headings prepars security part.

2 Security in Orientation Project

2.1 Applied Design Principles

2.1.1 Principle of Least Privilege

To limit the potencial damage processes that run script or execute code should run under a least privileged account. The privileges granted to that project determine which operations is able to perform a malicious user.

In Visual Studio ASP.NET the default account is ASPNET. The ASPNET account runs with least privileges. This change was implemented for the initial relese of the .NET Framework.[1]

2.1.2 Principle of Use Defense in Depth

Check the rights before step into the next layer or subsystem. The check points are the gatekeepers that ensure that only authenticated and authorized users are able to access the next layer.

It is necessary to run authentication and authorisation process befor access a new layer.

2.1.3 Principle of don't Trust User Input

Applications should validate all user input before performing operations with that data.

The ASP.NET Framework provides validator controls to check user input in an ASP.NET Web page. This controls let us to set client-side validation or server-side validation. Client-side validation performs client script. It can enhance user experience through immediate feedback, but it is working with browsers that support DHTML (dynamic HTML). Server-side validation is performed on the server even if it was already performed on the client. It is against malicious users bypassing client-side validation.

The controls automatically detect if the browser supports DHTML and perform their checking accordingly. [3]

[1] Beta releases ASP.NET runs as SYSTEM!!!

2.1.4 Principle of Check at the Gate

‘You don’t always need to flow a user’s security context to the back end for authorization checks. Often, in a distributed system, this is not the best choice. If you design solid authentication and authorization strategies at the gate, you can circumvent the need to delegate the original caller’s security context all the way through to your application’s data tier.’ [4]

2.1.5 Principle of Assume External Systems are Insecure

If you don’t own a service, don’t assume security is taken care of for you.

The ASP.NET Framework provides techniques for testing, supports automatic test.

2.1.6 Principle of Fail to a Secure Mode

Avoid exposing information that is not required. Also, do not provide too much detail in error messages, meaning don’t include details that could help attacker exploit vulnerability in your application. Do not leave sensitive data unprotected during exception handling. [4][2]

Mind that you are only as secure as your weakest link.

2.2 Security Model for ASP.NET Web Application

2.2.1 Logical Tiers

- User Services

  User Services are responsible for client interaction with the system. They are associated most often width interactive users, on the other hand they also provide processing of programmatic requests from other systems. Authentication and authorisation are typically performed within this layer.

- Business Services

  Business Services provide the core functionality of the system and encapsulate business logic.They are independent from the user interface or data sources. This permit of flexibility and scalability and provide enhance the system security.

[2] Microsoft: Building Secure ASP.NET Applications, Design Principles, p. 6

- Data Services

  Data Services provide access to data and to other systems through generic interfaces, which are convenient to use from components within the Business Services layer. They encapsulate specific access rules and data formats.

2.2.2 Physical Deployment

We can deploy our system in two ways:

1. **The Web Server is the Application Server**

   In this deployment pattern are the business and data access components on the Web server.

   Advantage of this solution is to reduce network communication so provides faster execution for users. The disadvantage is that it can lead additional vulnerabilities, because there is no firewall in front of the business and data access layer.

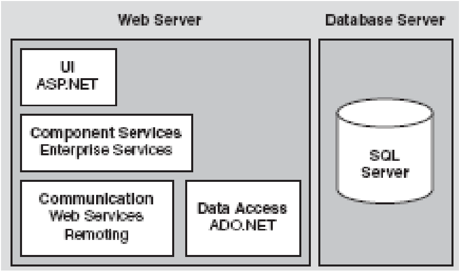

Figure 1
The Web server as an application server deployment pattern [4]

2. **Remote Application Tier**

   This is a common deployment pattern where the Web tier is self contained and is separated from the application server. The Web server is protected by Internet Information Services (IIS). IIS provides a gate when you authenticate users (that is, you disable Anonymous authentication). IIS Web permissions can be used as an access control mechanism to restrict the capabilities of Web users to access specific files and folders. These restrictions apply to individual users or groups. IIS checks Web permissions, followed by NTFS file permissions. A user must be authorized by both mechanisms for them to be able to access the file or folder.

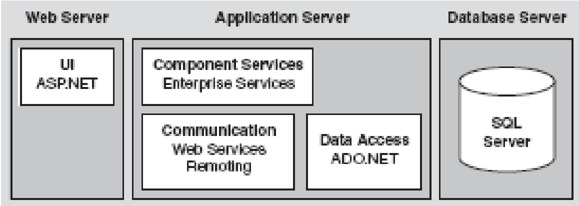

Figure 2
The remote application tier [4][3]

Advantage that we can use the Web tier as a demilitarized zone (DMZ) so we can give more access for information archived on Web server, but we can protect our business tier with the separation from end users. The disadvantage is that it needs more resources and the deeper communication can be slow.

2.3 Using Separate Processes

Separate Processes can provide different access for users to different operations. They can run on the same server or on physically separated servers.

If two tasks are performed in different tiers, or in different servers we have to separate them in different processes.

If we adopt the priciple of least privilege, some parts of code – which need more access –, can not perform. Code that requires additional trust should be isolated within separate processes. This processes of course will have raised privileges.

Conclusions

If we choose for the impelementation of our project Visual Studio ASP.NET application, we can take different security decisions. This framework provides tools for apply security principles. We can implement our application in three tier Model. The Visual Studio aids authentication authorisation process and secure communication among tiers and subsystems. So we can implement our project using security decisions.

[3] Microsoft: Building Secure ASP.NET Applications, Security Model for ASP.NET Applications p. 9

References

[1] Marianna Sipos: Orientation and Guidance within Doors, Proceedings of 3rd Serbian-Hungarian Joint Symposium on Intelligent Systems, Subotica, August 31- September 1, 2005, pp. 243-247

[2] Imola Szemes: Design and Implementation of an Indoor Guidance and Information System Using .NET and SVG, Dissertation on “Politehnica” University of Timisoara Faculty/Department of Automation and Computer Science and Budapest Tech, John von Neumann Faculty of Informatics, Institute of Softwaretechnology, 2006

[3] Microsoft Visual Studio Documentation, Client-Side Validation for ASP.NET Server Controls, Microsft, Redmond, 2005 ms-help://MS.VSCC.v80/MS.MSDN.v80/MS.VisualStudio.v80.en/dv_aspnetcon/html/35c73907-2928-4087-980b-cbd837c75b22.htm

[4] Microsoft: Building Secure ASP.NET Applications, Authentication, Authorization, and Secure Communication, Microsoft Press, Redmond 2003