INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

INTERNET AND SECURED DATA SHARING

MR. HARSHAL .D. WANKHADE[1], DR. H. R. DESHMUKH[2], PROF A. S. MAHALLE[3], PROF. S.A. KARALE[4]

1. ME Scholar at IBSS COE Amravati.
2. Head and Associate professor, Department of Computer Science And Engineering, IBSSCOE, Amravati Maharashtra, India.
3. Assistant Professor, IBSS COE, Department of Computer Science And Engineering, Maharashtra, India.
4. Assistant Professor, IBSS COE, Department of Computer Science And Engineering, Maharashtra, India.

Accepted Date: 27/02/2014; Published Date: 01/05/2014

Abstract: The Internet is revolutionary and enhancing the way we as humans communicate, both locally and around the globe. Simply put, the internet is a network of linked computers allowing participants to share information on those computers. You should want to be part of it because the internet literally puts a world of information and a potential worldwide audience at your fingertips. Security is an essential part of web applications and should be taken into consideration from the first stage of the development process. Essentially, security is all about protecting your assets from unauthorized actions. You use several mechanisms to this end, including identifying users, granting or denying access to sensitive resources, and protecting the data that's stored on the server and transmitted over the wire. In all of these cases, you need an underlying framework that provides basic security functionality.

Keywords: Deaf and dumb, Hand gesture, Human computer interaction, Sign language.

Corresponding Author: MR. HARSHAL .D. WANKHADE

Access Online On:

www.ijpret.com

How to Cite This Article:

Harshal Wankhade, IJPRET, 2014; Volume 2 (9): 1190-1202

PAPER-QR CODE

INTRODUCTION

Security is an essential part of web applications and should be taken into consideration from the first stage of the development process. Essentially, security is all about protecting your assets from unauthorized actions. You use several mechanisms to this end, including identifying users, granting or denying access to sensitive resources, and protecting the data that's stored on the server and transmitted over the wire. In all of these cases, you need an underlying framework that provides basic security functionality

DATA SHARING

A system for sharing data between two or more sharing system. The system includes sharing of private access data from a private data memory of a first sharing partner, to a first shared data associated with the sharing partner. The system further includes a replication of the shared data from the first shared data memory to another shared data memory associated with a same sharing partner. Private is nothing but the access type use for distinguishing users on the basis of priority set by the partner who want to share data among other partners involve in group.

The data between the two users is shared on the basis of some access specifier. There are three type of access specifier namely private, public and friendly. When data access specifier is public then the data is available for accessing to all valid users. When specifier is private then only owner of the data can access the data. The friendly access type is available only for those users in group for which the owner of the data (table) want to allow view or access of the data.

2. Literature Survey

More than 15 years ago, Tim Berners-Lee performed the first transmission across HTTP. Since then, HTTP has become exponentially more popular, expanding beyond a small group of computer-science visionaries to the personal and business sectors. Today, it's almost a household word. When HTTP was first established, developers faced the challenge of designing applications that could discover and interact with each other. To help meet these challenges, standards such as HTML

(Hypertext Markup Language) and XML (Extensible Markup Language) were created. HTML established simple language that can describe how to display rich documents on virtually any compute platform. XML created a set of rules for building platform-neutral data formats that different applications can use to exchange information. These standards guaranteed that the

Web could be used by anyone, located anywhere, using any type of computing system. At the same time, software vendors faced their own challenges. Not only did they need to develop languages and programming tools that could integrate with the Web, but they also needed to build entire frameworks that would allow developers to architect, develop, and deploy these applications easily.

Understanding Potential Threats:

Creating a secure architecture and design requires that you have an in-depth understanding of your application's environment. You can't create secure software if you don't know who has access to your application and where possible points of attack might be. Therefore, the most important factor for creating a secure application architecture and design lies in a good understanding of environmental

Factors such as users, entry points, and potential possible threats with points of attack. That's why *threat modeling* has become more important in today's software development processes. Threat modeling is a structured way of analyzing your application's environment for possible threats, ranking those threats, and then deciding about mitigation techniques based on those threats. With this approach, a decision for using a security technology (such as authentication or SSL encryption) is always based on an actual reason: the threat itself. But threat modeling is important for another reason. As you probably know, not all potential threats can be mitigated with security technologies such as authentication or authorization.

The basic access authentication was originally defined by RFC 1945 (Hypertext Transfer Protocol – HTTP/1.0) although further information regarding security issues may be found in RFC 2616.

Advantages:

One advantage of the basic access authentication is that it is supported by all popular web browsers. It is rarely used on publicly accessible Internet web sites but may sometimes be used by small, private systems. A later mechanism, digest access authentication, was developed in order to replace the basic access authentication and enable credentials to be passed in a relatively secure manner over an otherwise insecure channel.

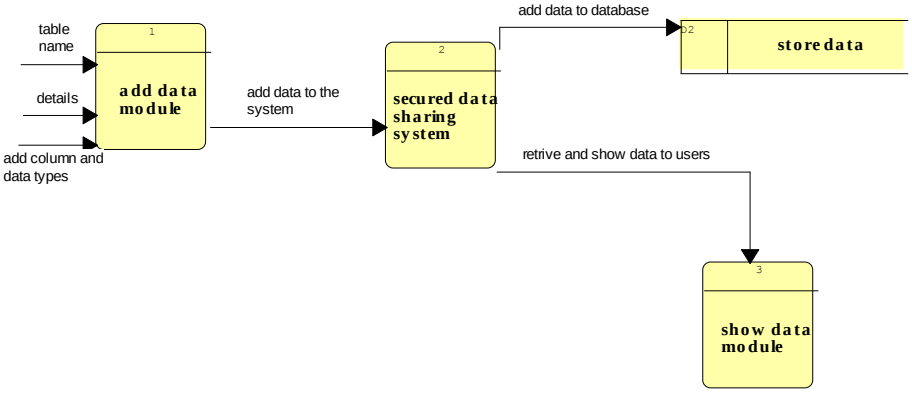
Disadvantages:

Although the scheme is easily implemented, it relies on the assumption that the connection between the client and server computers is secure and can be trusted. Specifically, if SSL/TLS is

not used, then the credentials are passed as plaintext and could be intercepted easily. The scheme also provides no protection for the information passed back from the server. Existing browsers retain authentication information until the tab or browser is closed or the user clears the history. HTTP does not provide a method for a server to direct clients to discard these cached credentials. This means that there is no effective way for a server to "log out" the user without closing the browser. This is a significant defect that requires browser manufacturers to support a 'logout' user interface element or API available to JavaScript, further extensions to HTTP, or use of existing alternative techniques such as retrieving the page over SSL/TLS with an ungues sable string in the URL.

System Design

SYSTEM ARCHITECTURE



The RSA algorithm involves three steps: key generation, encryption and decryption.

Key generation:

RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. The keys for the RSA algorithm are generated the following way:

1. Choose two distinct prime numbers *p* and *q*.

- For security purposes, the integer's *p* and *q* should be chosen uniformly at random and should be of similar bit-length. Prime integers can be efficiently found using a primarily test.

2. Compute $n = pq$.

- n is used as the modulus for both the public and private keys

3. Compute $\varphi(pq) = (p - 1)(q - 1)$. ($\varphi$ is Euler's totient function).

4. Choose an integer e such that $1 < e < \varphi(pq)$, and e and $\varphi(pq)$ share no divisors other than 1 (i.e. e and $\varphi(pq)$ are cop rime).

- e is released as the public key exponent.

- e having a short bit-length and small Hamming weight results in more efficient encryption. However, small values of e (such as $e = 3$) have been shown to be less secure in some settings.

5. Determine d (using modular arithmetic) which satisfies the congruence relation $de \equiv 1 \pmod{\varphi(pq)}$.

- Stated differently, $ed - 1$ can be evenly divided by the totient $(p - 1)(q - 1)$.

- This is often computed using the extended Euclidean algorithm.

- d is kept as the private key exponent.

The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the private (or decryption) exponent d which must be kept secret.

Encryption:

Alice transmits her public key (n,e) to Bob and keeps the private key secret. Bob then wishes to send message M to Alice.

He first turns **M** into an integer $0 < m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the cipher text c corresponding to:

$$c = m^e \bmod n$$

This can be done quickly using the method of exponentiation by squaring. Bob then transmits c to Alice.

Decryption:

Alice can recover *m* from *c* by using her private key exponent *d* by the following computation:

$$c^d \equiv m \pmod{n}.$$

Algorithm for adding data:

1. Start.

2. Create the data table by defining the TableName.

3. Select the appropriate access specifier private, public, friendly.

4. Add the details about table data in “details” textarea”.

5. Repeat the loop until user wants to add columns.

    5.1. Add the column name.

    5.2. Add the data type of added column.

    5.3. Goto step 5

6. Finally create table using “create table” button.

7. Stop.

3.3 Secured data sharing algorithm:

1. Start.

2. Enter the data table name you want to search.

3. If (exists) Goto step 4 else Goto step 2.

4. Check the TableName, TableOwner, details, access specifier.

5. If (access specifier==public)

Then

View the data directly.

6. If (access specifier==private)

Then

If (user==owner)

Then

View the data.

Else Display the appropriate error message.

7. If (access specifier==friendly)

If (user==owner)

Then

View the data.

Else if (user==friend)

Then

View the data.

Else

Display the appropriate error message.

8. Stop.

IMPLEMENTATION

4.1 Server Controls:

ASP.NET *server controls* are a fundamental part of the ASP.NET architecture. Essentially, server controls are classes in the .NET Framework that represent visual elements on a web form. Some of these classes are relatively straight forward and map closely to a specific HTML tag. Other

controls are much more ambitious abstractions that render a more complex representation from multiple HTML elements.

Types of Server Controls:

ASP.NET offers many different server controls, which fall into several categories.

HTML server controls:

These are classes that wrap the standard HTML elements. Apart from this attribute, the declaration for an HTML server control remains the same. Two examples include Html Anchor (for the <a> tag) and Html Select (for the <select> tag). However, you can turn any HTML tag into a server control. If there isn't a direct corresponding class, ASP.NET will simply use the Html Generic Control class. To change an ordinary HTML element into a server control, simply add the runat="server" attribute to the element tag.

Web controls:

These classes duplicate the functionalities of the basic HTML elements but have

A more consistent and meaningful set of properties and methods that make it easier for the developer to declare and access them. Some examples are the HyperLink, List Box and Button controls. In addition, several other types of ASP.NET controls (such as rich controls and validation controls) are commonly considered to be special types of web controls. In Visual Studio, you'll find the basic web forms controls in the Standard tab of the Toolbox.

Rich controls:

These advanced controls have the ability to generate a large amount of HTML markup and even client-side JavaScript to create the interface. Examples include the

Calendar, Add Rotator and Tree View controls. In Visual Studio, many rich controls are also found in the Standard tab of the Toolbox.

Validation controls:

This set of controls allows you to easily validate an associated input control against several standard or user-defined rules. For example, you can specify that the input can't be empty, that it must be a number that it must be greater than a certain value, and so on. If validation fails,

you can prevent page processing or allow these controls to show inline error messages in the page. In Visual Studio, these controls are found in the Validation tab of the Toolbox. These include the following:

Data controls:

These controls include sophisticated grids and lists that are designed to display

Large amounts of data, with support for advanced features such as templating, editing, sorting, and pagination. This set also includes the data source controls that allow you to bind to different data sources declaratively, without writing extra code.

Navigation controls:

These controls are designed to display site maps and allow the user to navigate from one page to another.

Login controls:

These controls support forms authentication, an ASP.NET model for authenticating users against a database and tracking their status. Rather than writing your own interfaces to work with forms authentication, you can use these controls to get prebuilt, customizable login pages, password recovery, and user-creation wizards.

4.2 The Server Control Hierarchy:

All server controls derive from the base Control class in the System. Web. UI namespace. This is true whether you're using HTML server controls, using web controls, or creating your own custom controls. It also applies to the Page class from which all web forms derive. Figure illustrates the main branches of this inheritance chain.

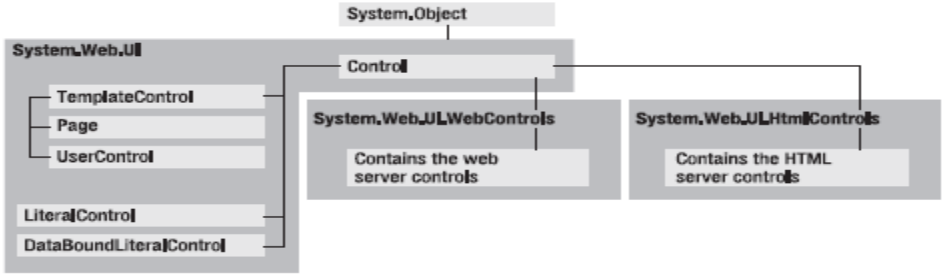


Figure 4.2.1: Server control inheritance

Because all controls derive from the base Control class, you have a basic common denominator that you can use to manipulate any control on the page, even if you don't know the specific control type.

Connection-based objects:

These are the data provider objects such as Connection, Command, DataReader, and DataAdapter. They allow you to connect to a database, execute SQL statements, move through a read-only result set, and fill a DataSet. The connection-based objects are specific to the type of data source, and are found in a provider-specific namespace (such as System.Data.SqlClient for the SQL Server provider).

Content-based objects:

These objects are really just "packages" for data. They include the DataSet, DataColumn, DataRow, DataRelation, and several others. They are completely independent of the type of data source and are found in the System.Data namespace.

Each provider has its own Namespace and generic classes such as the DataSet are stored in the System.

Table 7-1. *The ADO.NET Namespaces*

Namespace	Description
System.Data	Contains the key data container classes that model columns, relations, tables, datasets, rows, views, and constraints. In addition, contains the key interfaces that are implemented by the connection-based data objects.
System.Data.Common	Contains base, mostly abstract classes that implement some of the interfaces from System.Data and define the core ADO.NET functionality. Data providers inherit from these classes (such as DbConnection, DbCommand, and so on) to create their own specialized versions.
System.Data.OleDb	Contains the classes used to connect to an OLE DB provider, including OleDbCommand, OleDbConnection, OleDbDataReader, and OleDbDataAdapter. These classes support most OLE DB providers, but not those that require OLE DB version 2.5 interfaces.
System.Data.SqlClient	Contains the classes you use to connect to a Microsoft SQL Server database, including SqlDbCommand, SqlDbConnection, SqlDataReader, and SqlDBDataAdapter. These classes are optimized to use the TDS interface to SQL Server.
System.Data.OracleClient	Contains the classes required to connect to an Oracle database (version 8.1.7 or later), including OracleCommand, OracleConnection, OracleDataReader, and OracleDataAdapter. These classes are using the optimized Oracle Call Interface (OCI).
System.Data.Odbc	Contains the classes required to connect to most ODBC drivers. These classes include OdbcCommand, OdbcConnection, OdbcDataReader, and OdbcDataAdapter. ODBC drivers are included for all kinds of data sources and are configured through the Data Sources icon in the Control Panel.
System.Data.SqlTypes	Contains structures that match the native data types in SQL Server. These classes aren't required but provide an alternative to using standard .NET data types, which require automatic conversion.

The Connection Class:

The Connection class allows you to establish a connection to the data source that you want to interact with. Before you can do anything else (including retrieving, deleting, inserting, or

updating data), you need to establish a connection. The core Connection properties and methods are specified by the IDbConnection interface, which all Connection classes implement.

Connection Strings:

When you create a Connection object, you need to supply a *connection string*. The connection string is a series of name/value settings separated by semicolons (;). The order of these settings is unimportant, as is the capitalization. Taken together, they specify the basic information needed to create a connection. Although connection strings vary based on the RDBMS and provider you are using, a few pieces of information are almost always required:

How the database should authenticate you:

The Oracle and SQL Server providers give you the choice of supplying authentication credentials or logging in as the current user.

The latter choice is usually best, because you don't need to place password information in your code or configuration files.

4.8 The WebControl Base Class:

All the web controls inherit from the WebControl class. The WebControl class also derives from Control. As a result, many of its properties and methods—such as Controls, Visible, and FindControl ()—are similar to those of the HTML server.

Conclusion & Future Scope

Secured data sharing system is an intelligent application for finding people on the web and sharing data between them. This web application have greatly ease the pain of web users when they try to find the person who on this web application.

Thus, we have design an intelligent data sharing system which has strong features of finding people on the web and providing user with security by giving them right to keep their data private, public and friendly. User can join the application by registration on the web site and can find the people around by choosing appropriate option. Once the user finds the people he/she wants to, and then the data sharing comes into place. Sharing of data is totally depends upon the access type chosen by the user. User can keep his data available to all, available to certain groups or communities OR totally private. Thus this data sharing system can satisfy the need of user in more accurate and secured way.

Future scope

Secured data sharing system (SDSS) are growing rapidly, and offer a tantalizing set of data for users. Not only are there millions of users and connections between them, but users are often allowed to augment their relationships with information such as type, strength, or duration. This project sets out to present a comprehensive survey of these networks and their properties. We begin with criteria and definitions to determine which sites qualify for inclusion in this survey.

The fascination of data sharing is that it does not let you go. Once you have logged in you cannot just leave it. One would have never known that leaving data for friends and acquaintances is such connecting feel.. After a person joins the application slowly and slowly more people start discovering that person. There are a few from some organization, few from some other organization, few college mates and few school friends and it keeps rolling. The list of friends in always growing and as you interact with each other there comes a time when you feel that you have so many things common with many people. This compulsion to find likeminded people can be done through this application and so no one can resist it.If a person is an outgoing, extrovert, fun loving, looking for friends, then, yes the application is an addiction. This is friendship based site, where people create their profile, few would be genuine and upload their pictures as a profile photograph to share with those friends who are far, and cannot be contacted. On this application one starts searching friends from one friend, then go on to peep in their profile, search for community that interest you. There are variety communities with such diversity as from cars to condoms or dolls to dialysis. You join a community and then you become active, people will scrap some comment and then it goes on and goes on, and hence the first thing as soon as you are on the net is that you would do is to login to application check new guy / gal or weather your invited or not.

This application is so addictive that one would love to sit on it for hours together, for such frivolous reasons like looking for friends, creating data, sharing tables changing profile picture etc. The application can make reached new levels; people can spend their free time reading and writing friend's table/data on this so called social networking site. It is one of the more addicting things on campus - but the biggest benefits of it is that it really does put you back in touch with old friends and you can see what they're up to and all that. It can be made available many capabilities including online dating, people who share the same hobbies and interests as well as possible business connections. Once the line of your interest is known you will be able to keep in touch with people having same interest and also create your own community or special interest arena. The special interest community can be created by providing appropriate

access type to the data and lets you have many sharing on many topics the scope runs deep with the system as there are many variations available for creation of data in tabular format.

REFERENCES

1. MacDonald, Matthew; Szpuszta, Mario (2005). Pro ASP.NET 2.0 in C# 2005 (1st edition Ed.). Apress. ISBN 1-59059-496-7.

2. Professional IIS 7 and ASP.NET Integrated Programming by Dr. Shahram Khosravi.

3. ASP.NET MVC download site: http://asp.net/mvc/

4. Rhino Mocks: http://www.ayende.com/projects/rhino-mocks.aspx

5. Mock Object page on Wikipedia: http://en.wikipedia.org/wiki/Mock_object

6. "Netcraft Web Server Survey, February 2010". http://news.netcraft.com/archives/2010/02/22/fe bruary_2010_web_server_survey.html.

7. Dave Kramer (December 24, 1999). "A Brief History of Microsoft on the Web". Microsoft. http://www.microsoft.com/misc/features/feature s_flshbk.htm.

8. Vulnerability Report: Microsoft Internet Information Services (IIS) 6". http://secunia.com/advisories/product/1438/?tas k=statistics.