WEB SERVICES DOCUMENT SECURITY

by

G. SELVAKUMAR
Reg. No. 71203405015
of
Kumaraguru College of Technology, Coimbatore 641 006

PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements
for the award of the degree

of

MASTER OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING
JUNE 2005

BONAFIDE CERTIFICATE

Certified that this project report entitled **"WEB SERVICES DOCUMENT SECURITY"** is the bonafide work of **Mr. G.SELVAKUMAR**, who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

GUIDE **HEAD OF THE DEPARTMENT**

The candidate with **University Register No. 71203405015** was examined by us in the project viva-voce examination held on 22/6/05

R. Dinesh
INTERNAL EXAMINER

22/6/05
EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I would like to express my hearty gratitude to our Correspondent, **Dr. K. Arumugam,** and our principal **Dr. K.K. Padmanabhan** for giving me this great opportunity to do this course and accomplish this project.

I express my heartfelt thanks to **Dr.S.Thangasamy,** Professor and Head of Department of Computer Science and Engineering, for his support and encouragement .

I tender my special thanks to **Mr. R. Dinesh, M.S. (Wisconsin),** Assistant Professor and Project Coordinator, Department of Computer Science and Engineering for his excellent support.

I extend my sincere thanks to **Ms. L.S. Jayashree, M.E., Ph.D.,** Senior Lecturer, Department of Computer Science and Engineering for her valuable suggestions and guidance.

I express my deep sense of gratitude to my guide **Ms. V.S. Akshaya , M.E.,** Lecturer, Department of Computer Science and Engineering for her valuable suggestions and guidance throughout the project.

I extend my sincere thanks to **Mr.Karuppasami N(KPN),** Information analyst and **Mr. Arokiya jayakumar** Information associate, Electronic Data Systems, Chennai for their supervision, active involvement and valuable guidance.

I would like to convey my honest thanks to all **members of staff** of the Department for their unlimited enthusiasm, friendship and co operation from which I have greatly benefited.

TABLE OF CONTENTS

1. ABSTRACT

In today's world of extreme competition on the business front, information exchange and efficient communication are the need of the day. This need for information exchange brings in another need to make this information selectively visible, and its visibility to be changed on the fly. Web Services in the business world, in the most simplistic fashion, provides a mechanism of communication between two remote systems, connected through the network of the Web Services. This paper deals with a mechanism for providing security to documents of an enterprise whose applications are web services, by using the principle of Role Based access control – RBAC and XML specification language. With role-based access control, access decisions are based on the roles that individual users take as part of an organization. We need some mechanism to model these RBAC elements and their relationships. For this purpose we use XML to write the specifications. So XML based specification language models the RBAC elements and incorporates the functional specifications according to some RBAC standard. In this paper a framework is going to be used whose primary components are based on XML specification language. The primary aim of the framework is dissemination of secure documents for a single enterprise web service enabled application. The frame work consists of a policy base which contains XML specification files which represents the decision making principles of the organization. The user requests are processed by the framework and the access is controlled according to the privilege of the user. The proposed framework is under test with a primary set of documents of a distributed healthcare system. The future work involves developing the framework for multiple organizations.

ஆய்வுத் திட்டப்பணி சுருக்க வரைவு

தகவல் பரிமாற்றம் என்பது இன்றைய இணையம் சார்ந்த வணிக உலகின் தலையாய தேவையாகும். இணையத்தை ஊடகமாகக் கொண்டு பரிமாறப்படும் தகவல்கள், எந்த வரையறையுமின்றி யாவராலும் உபயோகிக்கப்படுத்தப்பட முடியாததாகவும், தகுதியான பயனாளர்களை மட்டும் சென்றடைவதாகவும் இருத்தல் வேண்டும்.

வெப் சர்வீஸ் என்பது இணையம் மூலம் இணைக்கப்பட்ட கணினிகளில் இயங்கும் மென்பொருள்களை இயங்குதளம், மொழி சார்பின்றி இணைப்பதையும் எளிய முறை தகவல் பரிமாற்றத்தையும் சாத்தியமாக்கும் ஓர் நவீன தொழில் நுட்பமாகும்.

இங்கு விளக்கப்பட்டிருக்கும் திட்டப்பணி, ஒரு நிறுவனத்தின் மதிப்புமிக்க தகவல்கள் இணையத்தில் பாதுகாப்பான முறையில் குறிப்பிட்ட பயனாளர்களை மட்டும் சென்றடைவதற்கான ஓர் வழிமுறையை ஆராய்கிறது. இதில் இரண்டு உத்திகள் பெரும் பங்கு வகிக்கின்றது. முதலாவது, பயனாளர்களை அவர்களது தகுதி, துறை, அனுபவம், பணிபுரியும் இடம் ஆகியவற்றை அடிப்படையாகக் கொண்டு பாகுபடுத்துதல் மற்றும் அவரவர்க்குரிய தகவல் பெறும் உரிமைகளை நிர்ணயித்தல், இரண்டாவது, மேற்படி நிர்ணயிக்கப்பட்ட விவரங்களை எக்ஸ்.எம்.எல் வரையறுப்பு மொழி மூலம் நடைமுறைப்படுத்துதல்.

இதன் நம்பகத்தன்மையை சோதித்தறிய பல்வகைப்பட்ட பயனாளர்களையும், தகவல்களையும் உள்ளடக்கிய ஓர் இணையம் சார்ந்த மருத்துவ சேவை நிறுவனத்துக்கான தகவல் தளம் உருவாக்கப்பட்டது. இவ்வமைப்பில் தகவல்கள் பொது உடைமை ஆகாமல், தகுதி வாய்ந்த, முன்னரே வரையறுக்கப்பட்ட பயனாளர்களை மட்டும் சென்றடைவது உறுதி செய்யப்பட்டது.

2. INTRODUCTION

2.1. WEB SERVICES

The revolution of computerizing services of companies gave rise to isolated computer systems. Each company had software developed and customized to its specific needs. However, mergers, acquisitions, and business growths saw the need to share information stored in these isolated computer systems.

The Internet did solve this problem to some extent. However, the Internet also opened many loopholes in security, making the owners of this information uneasy about the scope of their information's availability.

Hence, it became imperative that, for better B2B (Business-to-Business) communication, these systems must have the ability to link up to each other, grant permissions through a system other than the Internet, and which would make all the systems network with each other like an Intranet.

Web Services is probably not the first solution to such a problem. RMI, COM, CORBA, EDI, and ebXML also address the same problem space.
Web Services is based on the already existing and well-known HTTP protocol, and uses XML as the base language. This makes it a very developer-friendly service system. However, most of the above-mentioned technologies such as RMI, COM, and CORBA involve a whole learning curve. New technologies and languages have to be learnt to implement these services.

Also, Web Services is based on a set of standardized rules and specifications, making it more portable. This was not the case with the technologies mentioned earlier.

In a scenario, in which we need to locate a particular pharmacy store in our area, we would not go out on the road and ask every person we met the way to the store. We might, instead, refer the Web site of the pharmacy on the Internet.

If we knew the pharmacy's Web site, we would look it up directly and find the location through the store locator link. If not, we would go to a search engine and type out the name of the pharmacy in the language that the search engine was meant to recognize. After getting the location, we would find the directions to the store, and then go to the store.

The structure of Web Services is also very similar. Web Services provide for each of these previously described activities. If we carefully look at the preceding example, we will see that there is a requestor or a consumer. There is also a service, the pharmacy store. The central database of information is the Internet, through which you find the location of the pharmacy. In the example, when we fire a search in the search engine, our request is wrapped in a structure, whose language is predetermined and localized, and then passed onto the server running the search engine. In Web Services, SOAP, UDDI, and WSDL represent the roles mentioned in these steps.

SOAP (Simple Object Access Protocol) is the method by which we can send messages across different modules.
This is similar to how we communicate with the search engine that contains an index with the Web sites registered in the index associated with the keywords.
UDDI (Universal Description, Discovery, and Integration) is the global look up base for locating the services. In the example mentioned earlier, this is analogous to the index service for the search engine, in which all the Web sites register themselves associated with their keywords. It maintains a record of all the pharmacy store locations throughout the country.

WSDL (Web Services Definition Language) is the method through which different services are described in the UDDI. This maps to the actual search engine in our example.

The following Figure illustrates this concept.

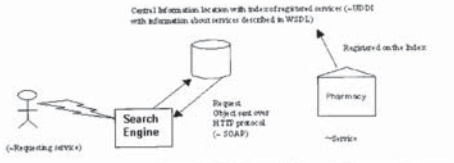

Central Information location with index of registered services (~UDDI with information about services described in WSDL)
Registered on the Index
Pharmacy
Search Engine
Request Object sent over HTTP protocol (~ SOAP)
(~Requesting service)
~Service

Fig 2.1.web services

Web Services in the business world, in the most simplistic fashion, provides a mechanism of communication between two remote systems, connected through the network of the Web Services. For example, in case of a merger or an acquisition, companies don't have to invest large sums of money developing software to bring the systems of the different companies together. By extending the business applications as Web Services, the information systems of different companies can be linked. These business systems then can be accessed by using simple SOAP messages over the normal HTTP Web protocol. For example, a manufacturing company requires some raw materials to be supplied whenever the material in stock reaches the threshold levels. These levels can be constantly monitored by the business system of a trusted supplier, and promptly replenished, without having to wait for a supervisor to notice it and generate a work order.

2.2. THE PROBLEM DOMAIN

When an enterprise application is published in the net as a web service, it is intended to be accessed by various kinds of users for different purposes. To control the visibility and usage of the data a security system is required which comprises of a set of components for authorization administration.

Information access may require restrictions based on the content and context related to the access requests. For example a digital library can contain some information inappropriate for children, and a web service that provides access to such a resource should deny access to users in a certain age group. Similarly for web services in the healthcare industry, relevant parties, including physicians should have access to selective content based patient information.

The access control model should also capture security relevant environmental context and incorporate it in its access control decisions. A digital library service may permit only students of a particular subscribing university intranet. Here the decision making is based on the context, in this case which is an authorized IP address.

Thus, the problem taken in this project work can be defined as "Providing security to documents of an enterprise whose applications are web services, by using the principle of Role Based access control – RBAC and XML specification language".

3. LITERATURE SURVEY

1. Rafae Bhatti, Elisa Bertino, Arif Ghafoor, James B.D.Joshi. "XML Based specification for web services document security", IEEE Computer, April 2004.

Document security in XML based Web services has become increasingly important for managing secure business transactions over the web. The authors propose an XML based access control specification language to address this security challenge. The paper examines information access restrictions based on context and content related to the access requests. It also deals with the subject and object heterogeneity which makes the access decision process more complex.

The secure documents that XML based Web service disseminates encompass diverse subjects and objects related to the applications. Object heterogeneity can exist either as abstract concepts or as knowledge embodied in the information that requires protection. So this work address these heterogeneity and proposes that the Role base access control model will be the suitable one to handle this. The paper says about the primary elements of RBAC model users, roles, permissions, operations and objects. Th: paper also introduces the XML specification language which models the RBAC elements and the framework proposed above.

2. David Ferraiolo and Richard Kuhn," Role-Based Access Controls"
15th National security conference(1992), National Institute of Standards a^nd Technology.

In many organizations in industry and civilian government, the end users do not ``own" the information for which they are allowed access.

For these organizations, the corporation or agency is the actual ``owner" of system objects, and discretionary access control may not be appropriate. Role-Based Access Control (RBAC) is a nondiscretionary access control mechanism which allows and promotes the central administration of an organizational specific security policy.

Access control decisions are often based on the roles individual users take on as part of an organization. A role specifies a set of transactions that a user or set of users can perform within the context of an organization. RBAC provide a means of naming and describing relationships between individuals and rights, providing a method of meeting the secure processing needs of many commercial and civilian government organizations.

Various forms of role based access control have been described and some are used in commercial systems today, but there is no commonly accepted definition or formal standards encompassing RBAC. As such, evaluation and testing programs for these systems have not been established as they have for systems conforming to the Trusted Computer Security Evaluation Criteria. This paper proposed a definition of The requirements and access control rules for RBAC proposed in this paper could be used as the basis for a common definition of access controls based on user roles.

3. "An Introduction to Role Based Access Control", NIST/ITL Bulletin, December 1995.

This article provides background information on Role-Based Access Control (RBAC), a technical means for controlling access to computer resources. While still largely in the demonstration and prototype stages of development, RBAC appears to be a promising method for controlling what information computer users can utilize the programs that they can run, and the modifications that they can make. Only a few off-the-shelf systems that implement RBAC are commercially available; however, organizations may want to start investigating RBAC for future application in their multi-user systems. RBAC is appropriate for consideration in systems that process unclassified but sensitive information, as well as those that process classified information.

The article also explains the primary kinds of access controls which are Discretionary access control, mandatory access control and the evolution of RBAC from those methodologies. Then it starts explaining users and roles. It explains under RBAC framework how users are granted membership into roles based on their competencies and responsibilities in the organization. Then it gives a brief introduction of roles and role hierarchies. Role hierarchies are a natural way of organizing roles to reflect authority, responsibility and competency.

The paper proceeds with roles and operations, and then it ends with stating advantages of RBAC model and current status of RBAC activities.

Further, it is possible to associate the concept of an RBAC operation with the concept of "method" in Object Technology. This association leads to approaches where Object Technology can be used in applications and operating systems to implement an RBAC operation

4. LINE OF ATTACK

The project intends to apply the solution methodology to a simulated environment and test its capability. The line of attack starts from developing a web service for a real time problem and applying the proposed mechanism for that environment.

The real time problem chosen here is a distributed healthcare system. A system of medical practitioners, pharmacists, lab technicians and administrative people working on health related tasks in a medical center. The primary data center is available as web services.

Then a framework will be modeled which consists of various components to enforce the access control of documents. For example the XML policy base and Access control module.

Users are allowed through the framework by proper authentication information and the details trigger the access control module to make the user to browse only the appropriate information. This particular job is done with the help of RBAC – Role based access control system and XML specification language.

So in a consolidated manner we can list down the operations as following.

- Studying the fundamentals of RBAC models
- Learning the Creation of XML documents
- Learning the tools to develop and deploy a web service
- Analysis of application domain
- Developing a Distributed healthcare system

- Studying different models of RBAC
- Determination of the appropriate model for the chosen application The core access policies of the system have to be defined.
- Implementation of the defined policies

5. DETAILS OF THE METHODOLOGY

The proposed solution consists of three important components.

5.1. RBAC- ROLE BASED ACCESS CONTROL

With role-based access control, access decisions are based on the roles that individual users have as part of an organization. Users take on assigned roles (such as doctor, nurse, teller, manager). The process of defining roles should be based on a thorough analysis of how an organization operates and should include input from a wide spectrum of users in an organization.

Access rights are grouped by role name, and the use of resources is restricted to individuals authorized to assume the associated role.
For example, within a hospital system the role of doctor can include operations to perform diagnosis, prescribe medication, and order laboratory tests; and the role of researcher can be limited to gathering anonymous clinical information for studies.

The use of roles to control access can be an effective means for developing and enforcing enterprise-specific security policies, and for streamlining the security management process.

A properly-administered RBAC system enables users to carry out a broad range of authorized operations, and provides great flexibility and breadth of application.

System administrators can control access at a level of abstraction that is natural to the way that enterprises typically conduct business. This is achieved by statically and dynamically regulating users' actions through the establishment and definition of roles, role hierarchies, relationships, and constraints. Thus, once an RBAC framework is established for an organization, the principal administrative actions are the granting and revoking of users into and out of roles.

The important component of the proposed solution is a specification language to draft the access control decisions which is based on xml.

5.2. XM BASED SPECIFICATION LANGUAGE

The RBAC model has five primary elements users, roles, permissions, operations and objects. These elements are related through set relations and functions. Permissions are composed of an object to operations mapping.

We need some mechanism to model these RBAC elements and their relationships. We use XML to write the specifications. So XML based specification language models the RBAC elements and incorporates the functional specifications according to some RBAC standard.

In this project work a framework is going to be used whose primary components are based on XML specification language. The primary aim of the framework is dissemination of secure documents for a single enterprise web service enabled application.

The frame work consists of a policy base which contains XML specification files which represents the decision making principles of the organization. The framework in its entirety is given below with access control module, XML policy base and all other components.

5.3. THE WEB SERVICES APPLICATION

5.3.1. The background

The simulated system is a Distributed Healthcare system. A system of medical practitioners, pharmacists, lab technicians and administrative people working on health related tasks in a medical center. The primary data center is available as web services.

The several kind of users of this environment are group of clinicians, nurses, doctors, accountants, administrative people of various grades, a Dean and a chief administrator and patients.

When it comes to operations we can consider the following to be undergone in the system.

5.3.2. Operations

Under Resource Administration,
Finances, personnel, Materials and support services, assets and facility management.
Under evaluation and planning
Activities, medical care, clinical research, quality assurance
Logistics of care delivery
Patient identification, Admission, Discharge, Transfer, appointment scheduling, service scheduling
Under Records Management
Medical records, insurance and legal documentation, clinical audit.

5.3.3. Choosing RBAC model

When we decide to deploy the concept of Role based access control model in our system, we have four options to choose from.

The options and their details are given below.
RBAC – 0 Basic
RBAC – 1 Basic + Hierarchy
RBAC – 2 Basic + Constraints
RBAC – 3 Basic + Hierarchy + Constraints
Selected one: RBAC – 3

We have selected the model RBAC 3. Since we have hierarchy of roles and some constraint based access in our system. The hierarchical arrangement of roles in our system is depicted in the figure below.

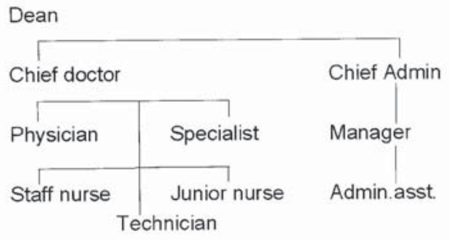


Fig.5.3.1. Hierarchy of roles

The constraint is a limitation when a particular role tries to access some specific part of a document. For example when a dental surgeon tries to browse the documents related with patient payments the system may not allow him to do so. It is a constraint here. In our system hierarchy and constraints are common aspects. That is why we have decided to choose RBAC model 3 for our system.

5.3.4. The XML Documents.

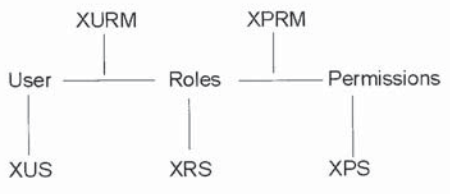


Fig 5.3.2. The Document structure

The above diagram depicts the XML documents used in the system and the nature of them.

XUS – XML User Sheet provides the details about the users and their credentials.

XRS – XML Role sheet describes the roles involved in the system and defines the users who are all takes that role.

XPS – XML Permission sheet explains the nature of permissions. That means the kind of objects which are allowed to read, write and modify by some permission ids are described here.

XURM – XML User Role Mapping is a document describes the association between users and roles of the system.

XPRM – XML Permission Role Mapping is a document provides the association between permission and roles, that is, which roles are allowed to do specific permissions.

5.3.5. The functioning of the system.

The Functioning of the system is based on the policy base created by the administrator. Policy base is a collection of xml documents listed above. Collectively the documents define the access control policies of the system. Here we describe the nature of every user, the role he can play, and the access permissions of the user.

So whenever a user login the system, based on his username and password we classify him to a role. The description of the user is available in XUS. The corresponding role played by the user is available in the XRS. We have the document XURM to map the user to a role. Every role has pre defined permissions which is determined by the document XPRM.

We deploy a module for access control that determines the content and context information. For example a cardiologist of the health care system will not be allowed to access the details of payment done by the patients. Similarly a clerical staff is not allowed to access the blood test results of any patient.

5.4. THE PROGRAMMING ENVIRONMENT

The dot net development environment has been chosen to develop the imaginary health care system. While have few more choices like java, we prefer dot net because of the availability of numerous tutorials and the user friendliness of the software.

The .NET Framework is a development and execution environment that allows different programming languages & libraries to work together seamlessly to create Windows-based applications that are easier to build, manage, deploy, and integrate with other networked systems.

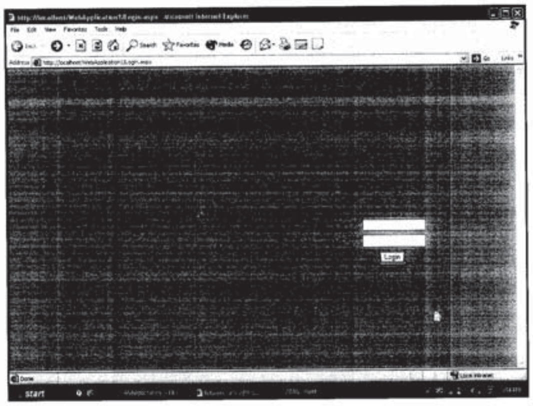
The .NET Framework consists of:

- The Common Language Runtime (CLR)
  A language-neutral development & execution environment that provides services to help "manage" application execution
- The Framework Class Libraries (FCL)
  A consistent, object-oriented library of prepackaged functionality
- The .NET Framework provides the basic infrastructure that Windows-based applications need to make Microsoft's .NET vision of connecting information, people, systems, and devices a reality:
- Support for standard networking protocols & specifications
  The .NET Framework uses standard Internet protocols and specifications like TCP/IP, SOAP, XML, & HTTP to allow a broad range of information, people, systems, and devices to be connected

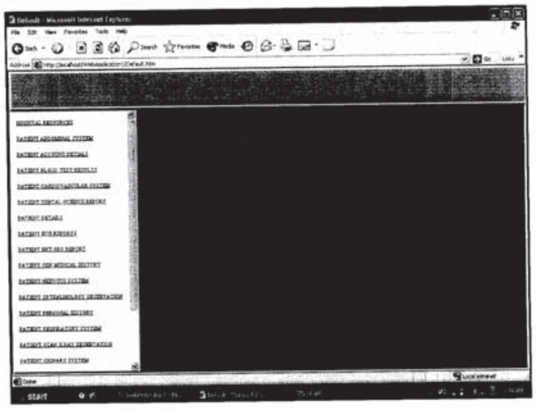
- Support for different programming languages
  The .NET Framework supports a variety of different programming languages so developers can pick the language of their choice
- Support for programming libraries developed in different languages
  The .NET Framework provides a consistent programming model for using prepackaged units of functionality (libraries) which makes application development faster, easier & cheaper
- Support for different platforms
  The .NET Framework is available for a variety of Windows platforms, which allows people, systems, and devices to be connected using different computing platforms. E.g. People using desktop platforms like Windows XP or device platforms like Windows CE can connect to server systems using Windows Server 2003.

6. RESULTS

After developing the framework appropriate for a simulated environment the interface is developed. The following figure shows the primary interface to the system, developed as an ASP page in dot net environment.

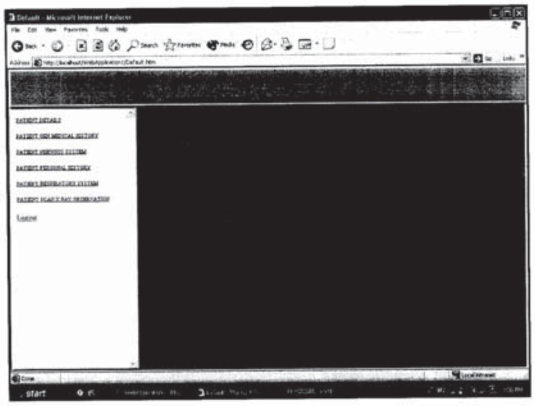


The next page provides a model for the page that provides navigation to the information pool according the role played by a user in the environment. In the following snapshot the user is Dean of the health care system. The information that he is authorized to access is listed in the left side of the page in the form of links.


HOSPITAL RESOURCES
PATIENT ABDOMINAL SYSTEM
PATIENT ACCOUNTS DETAILS
PATIENT BLOOD TEST RESULTS
PATIENT CARDIOVASCULAR SYSTEM
PATIENT DENTAL SCIENCE REPORT
PATIENT DETAILS
PATIENT ECG RESULTS
PATIENT ENT REPORT
PATIENT GEN MEDICAL HISTORY
PATIENT NERVOUS SYSTEM
PATIENT OPTHALMOLOGY OBSERVATION
PATIENT PERSONAL HISTORY
PATIENT RESPIRATORY SYSTEM
PATIENT SCAN XRAY OBSERVATION
PATIENT URINARY SYSTEM

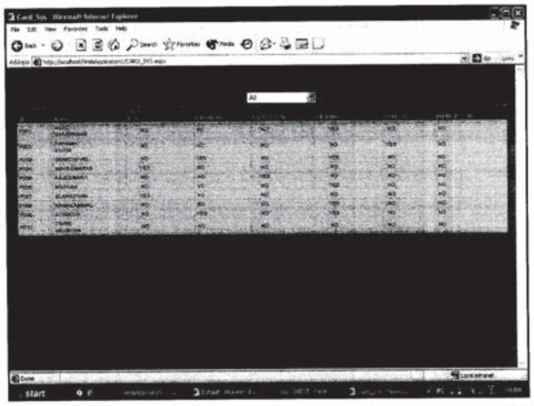
Similarly, whenever a user login the system, the role played by him is identified and the information he can access is shown in the form of links in the left side.

The appearance of the navigation screen for a neuro physician will appear like this.


Default - Microsoft Internet Explorer
PATIENT DETAILS
PATIENT NEW MEDICAL HISTORY
PATIENT NERVOUS SYSTEM
PATIENT PERSONAL HISTORY
PATIENT RESPIRATORY SYSTEM
PATIENT SCAN X RAY INFORMATION
Logout

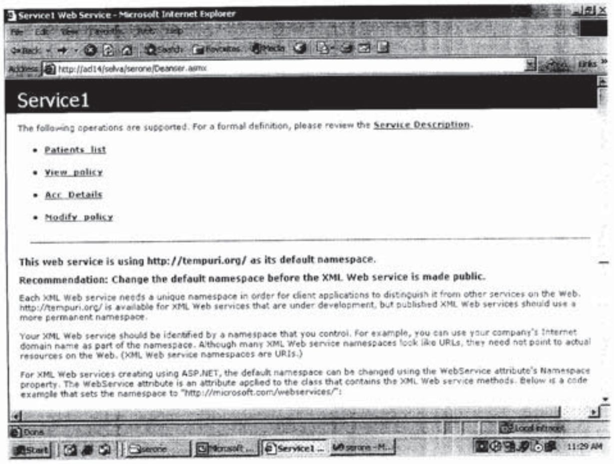
Here we can see, the information he can access are limited and only the details relevant to his expertise and nature of work are available.

This screen shot shows the information returned by a web service on a request by a cardiologist.



Here we provide the details of the patients on individual basis, as well as the entire data. For that purpose we show a combo box. So the user can select the details of all the patients of any particular user.

The next screen shots shows the description of a developed web services for illustration purpose. Here we see the list of web methods created for a web service.


Service1 Web Service - Microsoft Internet Explorer
http://acl14/selva/serone/Deanser.asmx
Done
Local intranet
11:29 AM

Service1

The following operations are supported. For a formal definition, please review the **Service Description**.

- **Patients_list**
- **View_policy**
- **Acc_Details**
- **Modify_policy**

This web service is using http://tempuri.org/ as its default namespace.

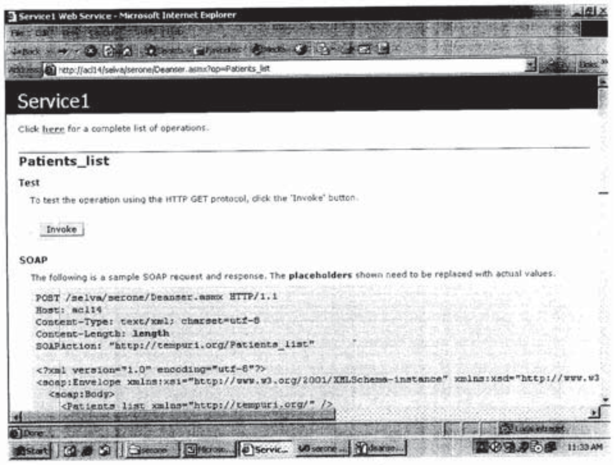
Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. http://tempuri.org/ is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services creating using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "http://microsoft.com/webservices/":

When a web service is developed, the details about the web methods, the web service intended to execute are displayed as below. This example screen shows the details of a web method patients list in a web service called dean service.


Service1 Web Service - Microsoft Internet Explorer
http://acl14/selva/serone/Deanser.asmx?op=Patients_list
Invoke
Done
Local intranet
11:33 AM

Service1

Click here for a complete list of operations.

Patients_list

Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

SOAP

The following is a sample SOAP request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /selva/serone/Deanser.asmx HTTP/1.1
Host: acl14
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/Patients_list"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3
  <soap:Body>
    <Patients_list xmlns="http://tempuri.org/" />
```

7. CONCLUSION

This project work deals with a means to provide security to documents of an enterprise whose documents are available as web services. Here I have taken a modern approach which uses an effective access control mechanism and the implementation is on .NET platform. While web services are getting much attention, so many access control mechanisms are in market to protect the web documents. The first part of the project analyses an access control mechanism called RBAC- Role Based Access Control which is efficient and effective to protect web documents. To represent RBAC we need a mechanism. Here we use XML Specification Language for that purpose. We restrict every user based on the role played by him. The nature of the user, the role taken by him, the access permissions of the role are clearly defined by following any one of RBAC standard. Then they are represented in XML. The collections of these XML documents are known as XML policy base which plays the central role in restricting access to the documents. To validate the functioning of this idea, we build a imaginary health care system whose users are from Dean to Lab technician. RBAC 3 model is chosen to represent the policies of the system. Then the xml policy base is constructed. A web interface is created and whenever a user enters into the system the role played by him is identified and the appropriate portions of the documents are available for him to access. Thus the proposed mechanism is tested and functioning well. This mechanism is found comparatively simple and effective then other mechanisms in the market today. It is because of the advantages in RBAC and the power of the XML language.

8. SCOPE FOR FUTURE WORK

This project work deals with a means to provide security to documents of an enterprise whose documents are available as web services. At the end of the project work, it has a good scope for future work. When it comes to access control the project handles only read access control of documents. The write and modify access controls have not been done. But it requires a much less modification in the work. We need to change the xml policy base to incorporate few more access specifications like write and modify. Then the web service has to provide web methods appropriate for write and modification. The major scope for enhancement for the project lies in its size. The done work is meant for documents of a single organization. We have the scope to modify it to service multiple organizations. While we have documents of multiple organizations in one side and users of those organizations in other side, the intermediate policy base and access control mechanism becomes so complex and huge. The similarity between users of different organization has to be found, and the documents have to be organized with utmost care.

9. REFERENCES

1. Rafae Bhatti, Elisa Bertino, Arif Ghafoor, James B.D.Joshi. "XML Based specification for web services document security", IEEE Computer, April 2004.

2. David Ferraiolo and Richard Kuhn," Role-Based Access Controls"
15th National security conference(1992), National Institute of Standards and Technology.

3. "An Introduction to Role Based Access Control", NIST/ITL Bulletin, December 1995.

4. Web Services Security, Mark O'Neil, et al. Tata McGraw-Hill Edition, 2003.

5. Step by Step XML, Michael J.Young. Prntice Hall of India,2001.

6. www.webservicesarchitect.com

7. www.webservices.org

8. www.dev.systinet.com (Developers corner tutorial.)

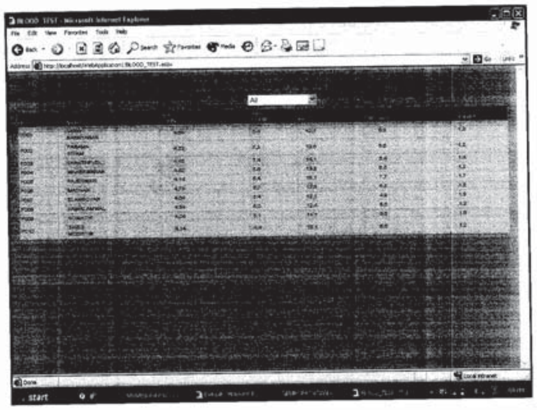
9. www.msdn.microsoft.com

10. www.java.sun.com

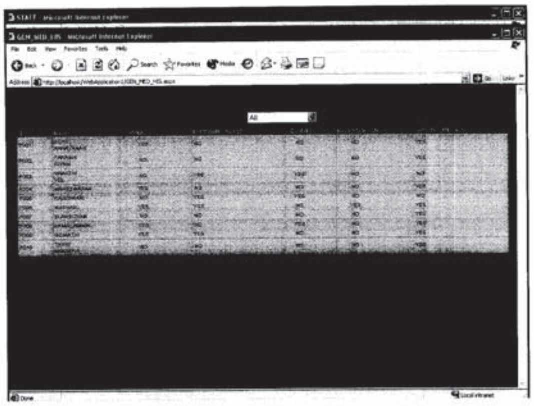
10. APPENDIX

9.1 More snapshots.

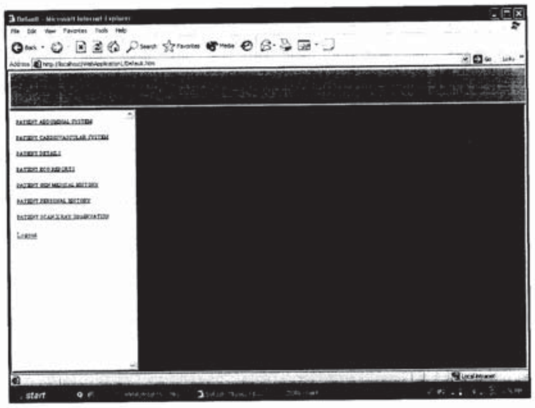
9.1.1 Blood test report of the patients.



9.1.2 Medical history of the patients



9.1.3 The screen for a physician


Default - Microsoft Internet Explorer
PATIENT ABDOMINAL SYSTEM
PATIENT CARDIOVASCULAR SYSTEM
PATIENT DETAILS
PATIENT ECG REPORTS
PATIENT GEN MEDICAL HISTORY
PATIENT PERSONAL HISTORY
PATIENT SCAN X RAY INVESTIGATION
Logout

9.2 Code samples

The following XML specification document shows how the credential of an user in a Health Care Organization could be defined.

```
<credentials>
        <credential>
                <cred type cred_type_id='c100'>
                Nurse
                </cred type>
           <cred_expr>
                <att_value_list>
                        <att_value_pair>
                                <att_name>
                                        user_id
                                </att_name>
                                <att_value>
                                        John
                                </att_value>
                        </att_value_pair>
                        <att_value_pair>
                                <att_name>
                                        age
                                </att_name>
                                <att_value>
                                        30
                                </att_value>
                        </att_value_pair>
                        <att_value_pair>
                                <att_name>
```

```
                                        level
                                </att_name>
                                <att_value>
                                        5
                                </att_value>
                        </att_value_pair>
                </att_value_list>
        </cred_expr>
</credential>
</credentials>
```

This document shows the credential of a user whose user identification is Nurse. The attribute value pair is an XML element which provides the name, age and level of access given to that particular user. Collectively it is known as credential expression which is (Nurse, {(user_id, John),(age,30),(level,5)})
So the above document is known as XML user sheet - XUS which contains the details about the user.

Similarly the roles of an organization can be defined by the XML role sheet which is given below.

```
<roles>
        <role>
                <role_name>Doctor
                </role_name>
                <duty_role_id>dri 1
                <duty_role_id>
        </role>
</roles>
<dutysets>
        <dutyset ds id='dri 1'>
        <priv>prescribe
        </priv>
        </dutyset>
</dutysets>
```

The above XML specification document denotes the authorization of a role Doctor in terms of his privileges. So each role will be assigned to some specific privileges or permissions. And each permission is assigned for some objects to which the access is permitted.

```
<permissions>
        <permission>
                <perm_id>
                        p1
                </perm_id>
                <obj_type>
                        med_report
                </obj_type>
                <operation>
                         read
                </operation>
                <perm_id>
                        p2
                </perm_id>
                <obj_type>
                        prescription
                </obj_type>
                <operation>
                        all
                </operation>
 </permission>
```

```
</permissions>
```

The above is the XML permission sheet which defines permissions for objects and associated operations in a given system. The permission component usually consists of system dependent operations such as read write, delete or modify.

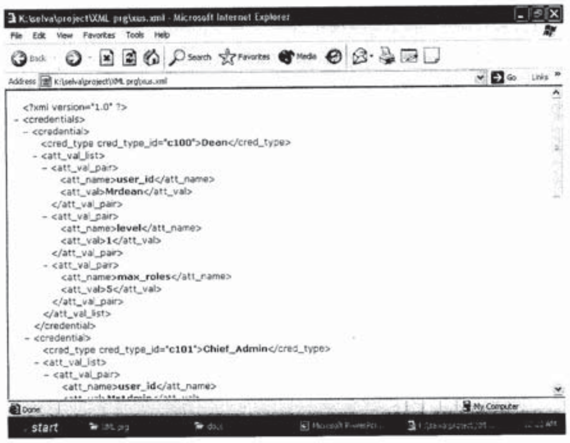
Those above documents forms the base for the decision making process. They are used in advanced XML specification documents which provides the mapping. The code given below illustrates the mapping of permission to roles. XPRM stands for XML specification for Permission Role Mapping.

```
<xprm>
        <prm prm_id='prm1'>
                <role_name>
                        Eye_Doctor
                </role_name>
                <permissions>
                <perm_id>
                        p2
                </perm_id>
                </permissions>
        </prm>
        <prm prm_id='prm2'>
                <role_name>DBA
                </role_name>
                <permissions>
                <perm_id>
                        p1
                </perm_id>
                <perm_id>
```

```
                        p2
                </perm_id>
                </permissions>
        </prm>
<xprm>
```

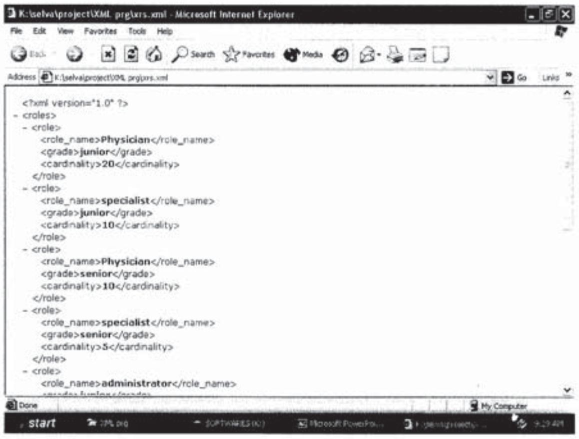
This permission role mapping shows what are all the permissions of assigned for a role. Already we would have defined what users will take what roles by another mapping known as user role mapping- XURM. Along with that specification the above XPRM resolves permissions are allowed to users of various kinds. All these XML specification documents forms the XML policy base, the major component of the framework.

The created XML specification documents appear like this. This document is known as XUS. XML User sheet.


K:\selva\project\XML prg\xus.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
K:\selva\project\XML prg\xus.xml
Go
Links
Done
My Computer
start

```
  <?xml version="1.0" ?>
- <credentials>
  - <credential>
      <cred_type cred_type_id="c100">Dean</cred_type>
    - <att_val_list>
      - <att_val_pair>
          <att_name>user_id</att_name>
          <att_val>Mrdean</att_val>
        </att_val_pair>
      - <att_val_pair>
          <att_name>level</att_name>
          <att_val>1</att_val>
        </att_val_pair>
      - <att_val_pair>
          <att_name>max_roles</att_name>
          <att_val>5</att_val>
        </att_val_pair>
      </att_val_list>
    </credential>
  - <credential>
      <cred_type cred_type_id="c101">Chief_Admin</cred_type>
    - <att_val_list>
      - <att_val_pair>
          <att_name>user_id</att_name>
```

The XML document that defines the roles associated with users is known as XRS. XML Role sheet. The appearance of an XRS is shows below.


K:\selva\project\XML prg\xrs.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address K:\selva\project\XML prg\xrs.xml
Go
Links
Done
My Computer
start

```
<?xml version="1.0" ?>
- <roles>
  - <role>
      <role_name>Physician</role_name>
      <grade>junior</grade>
      <cardinality>20</cardinality>
    </role>
  - <role>
      <role_name>specialist</role_name>
      <grade>junior</grade>
      <cardinality>10</cardinality>
    </role>
  - <role>
      <role_name>Physician</role_name>
      <grade>senior</grade>
      <cardinality>10</cardinality>
    </role>
  - <role>
      <role_name>specialist</role_name>
      <grade>senior</grade>
      <cardinality>5</cardinality>
    </role>
  - <role>
      <role_name>administrator</role_name>
```

9.3 Abbreviations

RBAC – Role Based Access Control

SOAP - Simple Object Access Protocol

UDDI - Universal Description, Discovery, and Integration

WSDL - Web Services Definition Language

XUS – XML User Sheet

XRS – XML Role sheet

XPS – XML Permission Sheet

XURM – XML User Role Mapping

XPRM – XML Permission Role Mapping