Performance Evaluation of Add-On Security Services Incorporated on Business Services

Thirumaran.M
Assistant Professor
Dept. Of CSE
Pondicherry Engg College

Dhavachelvan.P
Professor &HOD
Dept. Of CSE
Pondicherry University

Aishwarya.D
PG Scholar
Dept. Of CSE
Pondicherry Engg College

Shanmugapriya.R
PG Scholar
Dept. Of CSE
Pondicherry University

ABSTRACT

As more and more services come under the purview of internet, online businesses have also made a flourishing start, so much so that a bulk of the business transactions have become web- based. Web services have found their way into sectors like banking, finance, construction, airlines, marketing, and telecommunication and so on. We propose to design an add-on security model which provides interoperable security for the business services. It will be flexible enough to adapt itself to the security requirements of a specific business. This add-on security service models will increase the security levels of the business services. The model designed is evaluated after integrating the add-on security services with the business services.The fundamental approach we follow, for creating our project is Model Driven Security Service Enforcement, SLA based Security Service Negotiation, Automatic Security Service Integration, Performance Evaluation, and Impact Analysis. This is a definite way to achieve better performance, consistency and cost effectiveness in business services by making it invulnerable to hacker attacks.

Keywords Web Services, Security Negotiation, SLA attacks , Hacker Attacks

1. INTRODUCTION

Web services are application components which can communicate using open protocols. Web services are independent in nature which can be discovered using UDDI (Universal Description, Discovery and Integration). In short, web services employ a distributed computing technology and allow us to create heterogeneous client server applications. Due to the augmentation of businesses being carried out via web services, there is a drastic increase in the consumer portals being deployed. The service consumption rate has also been increasing radically. As of December 2005, the eBay service has more than 25,000 members who created 1,900 live applications. Approximately 47% of eBay.com listings are through eBay Web Services. Nearly 50% are from third-party developer tools created by companies. The first section of this paper deals with the related works of this domain. A service-oriented architecture (SOA) is a collection of services communicating with each other. The communication can involve either simple data passing or it could be two or more services coordinating some activity. Some means of interconnecting services to each other is needed. SOA is a versatile tool providing assistance to consumers of services. There are three levels at which the services operate: Service providers, Consumers, and End users. In recent times, all the end users have come to depend on the online services. More than the service providers, it is the consumers who are expected to satisfy the requirements of the end users as they interact directly with the users. When the users are in need of a particular service they place a request to a consumer who dispatches the work to either the service providers or to the 3rd parties depending on the efficiency and cost effectiveness with which the task is completed. With the increase in the rate of cyber crimes, classified information must be protected. The services offered by the Service Providers are subject to many risks and hence they are treated as being untrustworthy by the end users the web tool offered by the service providers is not reliable because it does not adequately address the security issues involved. The security challenges presented by the Web services are formidable and unavoidable. Many of the features that make Web services attractive include greater accessibility of data, dynamic application-to-application connections, and relative lack of human intervention which are at odds with the traditional security models and controls. The identity, integrity, and security of the data and the user must be preserved in most situations. More than one encryption key may be used. The primary purpose of this paper is to devise a method to secure Web services by using the five models of authentication, authorization, integrity, confidentiality and non-repudiation. So we propose an approach to express security goals at the business process level. Then the model designed with these security criteria are incorporated with the already existing business services as an add-on feature. Then this integrated model is evaluated to verify its performance, consistency and cost effectiveness. The proposed models and evaluations of their performance consistency are enumerated in Section 2.And in the last Section, the method by which the security models are designed and incorporated are described.

2. RELATED WORKS

Christian Wolter discuses about Various types of security goals, such as authentication or confidentiality, can be defined as policies for service-oriented architecture [1] . Claudio A. Ardagna, Sabrina De Capitani di Vimercati discuses about a simple and effective formalization of novel concepts that have to be supported for enforcing the new access control paradigm needed in open scenarios, toward the

aim of providing an expressive solution actually deployable with today's technology [2]. Christian Wolter discuses about. [3] ShaziaWasimSadiq discuses about process design is primarily driven by process improvement objectives. Major technical and organizational challenges is identified.[4] Michiaki Tatsubori discuses about emerging tool for security configuration of service-oriented architectures with Web Services. The users must construct their own mental models of how the security configurations actually relate to business policies. [5] N. Nagaratnam discuses about Business-driven development, management of secure applications and solutions. [6]

3. EXISTING SYSTEM

In the systems currently existing the sequence diagrams are used to depict the flow of activities taking place inside the system. But these sequence diagrams are not in a machine processible format. In the mechanism for representation of sequences so far the design document and code utilized are designer dependent and platform dependent repectively.So a more effective method to represent the sequence of activities will be through an XML (Extensible Markup Language).XML is platform-independent international standard and thus immune to changes in technology.XML is widely used in the representation of Data Structures like Web Services. It is a tag oriented language where all the entities may be easily defined. The assessments may be easily done through the XML since it has a more flexible processible format. The documents are encoded in a machine readable pattern.

In our paper we propose to keep the log in feature as our root tag and the entities depicting the transitions from the source to the destination of the metamorphosis. In an XML file the input and output is given in a specified format with the operation name as the function name. The applications that are devised by this method are dynamic in nature. When the output of this XML file is a string then that implies that the text has not been encrypted. And when it is found to be a long string then it deduced to be in an encrypted format. XML is a universal format for representation of the data in a methodical format. Its self-documenting format describes structure and field names as well as specific values. In applications like MICROSOFT WORD or PDF the contents may be fancy but they are not processible by nature. The Auto run format which is a distinguished feature is supported in XML language.

3.1 Model Driven Security Service Enforcement

The challenge in security of complex distributed systems does not anymore lie in encryption or access control of a single middleware platform, but in the protection of the system as a whole. This includes the definition of correct security policies at various abstraction layers, and also in the unified and correct management and enforcement of the correct security policy at all relevant places in the system. The authors have learned in the development even of comparatively simple distributed systems that this is not possible anymore by a manual definition of encryption properties and access control rules. Human security administrators are not able to define all these fine grained rules with sufficient assurance, to distribute them to all Policy enforcement Points and to check many log files or admin consoles. This is especially impossible in highly distributed and agile service oriented or data driven systems. In this paper we will illustrate the approach and architecture behind Model Driven Security Management and provide a healthcare regulatory compliance case study.

3.2 SLA based Security Service Negotiation

We use simple agent technology to enable the client to negotiate the best Qos agreement from a set of service providers. Our macroscopic negotiation is thus between a single client and many service providers. The service provider infrastructure employs a Qos manager to perform microscopic negotiation and work out the best reservation possible to offer to the client.

Automatic Security Service Integration

Security in SQL Server Integration Services consists of several layers that provide a rich and flexible security environment. These security layers include the use of digital signatures, package properties, SQL Server database roles, and operating system permissions. Most of these security features fall into the categories of identity and access control.

Performance Evaluation

It provides for the establishment and communication of employees' performance plans and procedures for evaluating employees' performance. Impact Analysis is a technique designed to unearth the "unexpected" negative effects of a change on an organization. It provides a structured approach for looking at a proposed change, so that you can identify as many of the negative impacts or consequences of the change as possible. Firstly, this makes it an important tool for evaluating whether you want to run a project. Secondly, and once the decision to go ahead has been made, it helps you prepare for and manage any serious issues that may arise. All too often organizations do not undertake Impact Analysis. This is one reason that so many projects end in failure, as unforeseen consequences wreak havoc.

4. CASE STUDY

The travel industry is a big business in the world and travel agencies make up an important part of the industry. Agencies can access details about hundreds of package holidays, flights, accommodations and excursions. Travel agents have the expert knowledge to help their customers find and book a holiday package. Special agencies for booking family holidays, city sightseeing, tours, business travels are also present. Travel agencies are trained to give advice on destinations worldwide and make arrangements for transportation, hotel accommodation, car rental and sightseeing tours.

Our project is to create services for the complete travel domain starting from reservation to managing food and travel services and ending with the cancellation of services if required. The services we have decided to provide are Registration, log in, availability checking, reservation, room booking, food services, cab booking, mode of payment, cancellation and messaging services i.e. confirmation mail.

The five entities are airline, hotel, local transport, bank and agency. These five entities are responsible for providing services. The airlines provide two major services of reservation and cancellation. Hotel is responsible for Room booking and availing of food services, the local transport for obtaining the mode of transport within the place like cab booking. The bank is the means by which the user pays the agency for the services that they are providing so hence it holds the mode of payment service. The last and most important entity agency is capable of providing registering service, login, checking the availability of tickets, messaging

service. These are the services that are usually provided by any travel domain.

There are a lot many problems occurring in such travel services offered by an agency. There may be issues of security which result in the customer feeling insecure. This insecurity results ultimately in the loss of customers. This is a major setback for any agency whose goal is to increase the amount of customers.

We propose to enhance the security of these sites. And we intend to do this by designing five add-on services (authentication, authorization, integrity, confidentiality and non repudiation).The services designed using Net BeansIDE6.1 are easy to create and to use. These services can be created within a matter of minutes. However the major disadvantage of Net Beans is that it employs several built in features which while making it faster also makes it difficult to incorporate and integrate the add on features.

Services created using Net Beans make it impossible to integrate the security services that we have designed. Creation of new services and integrating our services results in the time delay during which our customers may decide to move onto some other agency providing a constant service. This results in the loss of customer faith and trust in our service. Customer satisfaction is our key goal which is vital. So we must find a way to create the services using a different application and integrate our add on services which is designed into those services. This ultimately results in the usage of Sun Application Server 8.0.This application has its own merits. The services that are created using this application can be modified to accommodate our services. Even though the creation of these services takes a lot of time the services designed are more adaptable and flexible. The services of registration, log in, checking availability requires the add -on feature of authentication. Reservation, room booking, availing food services, cab booking require the authorization service. Mode of payment via the bank needs the confidentiality as criteria. Authorization and integrity both are add -on feature for the cancellation service. Non-Repudiation is a necessary add on for the messaging or confirmation service.

AUTHENTICATION

The weakness in this system for is that passwords can often be stolen, accidentally revealed, or forgotten. The necessity for the service providers to have the password of the end users is nil. So we tend to overcome these drawbacks of the existing system by proposing a new model.

In the previous systems when a user gives a password during the sign up process it gets automatically stored in the database as such in its original form. On subsequent usage, when the user types his password in the log in page it is checked with the one stored in the database. If they exist then the password is checked with the one provided by the user. If an exact match is found then the user is declared to be the authentic person to access the site. If they do not match then the user is declared to be an imposter.

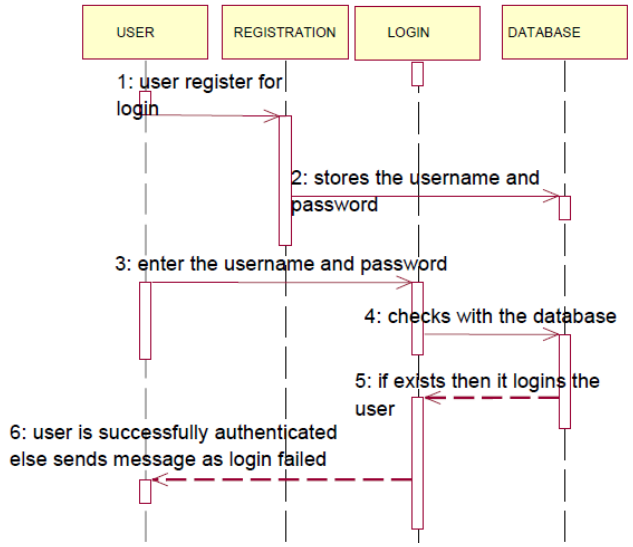

Figure1. Authentication

```
<? xml version="1.0" ?>
< sequence>
<server>authentication</server>
 <entity name="user">
 <action id="1">Registration</action>
<inputname="" type="text" />
<inputage="" type="text" />
<inputaddress="" type="text" />
<inputusername="" type="text" />
<inputpassword="" type="text" />
<action id="2">Login</action>
<inputusername="" type="text" />
<inputpassword="" type="text" />
</entity>
</sequence>
```

AUTHORIZATION

The cancellation of tickets is a huge process requiring a lot of steps and the tickets may be cancelled accidentally by the consumers or the agency. When a user wants to use a certain service of an agency he has to log in using his user name and password. Then password provided is checked with the one existing in the database. When the log in is a success the user is allowed to book the tickets and to avail further services. All the details of these transactions are stored in the database. When the same user wants to cancel the services he has availed, he tells the agency, the agency in turn contacts the service provider. The service provider checks the database if the user has already booked the tickets. And then the ticket is cancelled if they have been priory booked.

```
<? xml version="1.0" ?>
 <Sequence>
<servername>Authorization</servername>
<entity name="user" id="1">
        <action id="1">Reservation</action>
        <inputname="" type="text" />
        <inputaddress="" type="text" />
        <inputemailid="" type="text" />
        <inputairlinename="" type="text" />
        <inputfromplace="" type="text" />
        <inputtoplace="" type="text" />
        <inputnooftickets="" type="text" />
        <inputdate="" type="date" />
```

```
<inputtime="" type="time" />
 <entity name="agency" role="customer"
access="false" id="2">
<action id="2">Cancellation</action>
<inputname="" type="text" />
<inputticketnumber="" type="text" />
<inputairlinename="" type="text" />
<inputNooftickets="" type="text" />
</entity>
</entity>
</sequence>
```

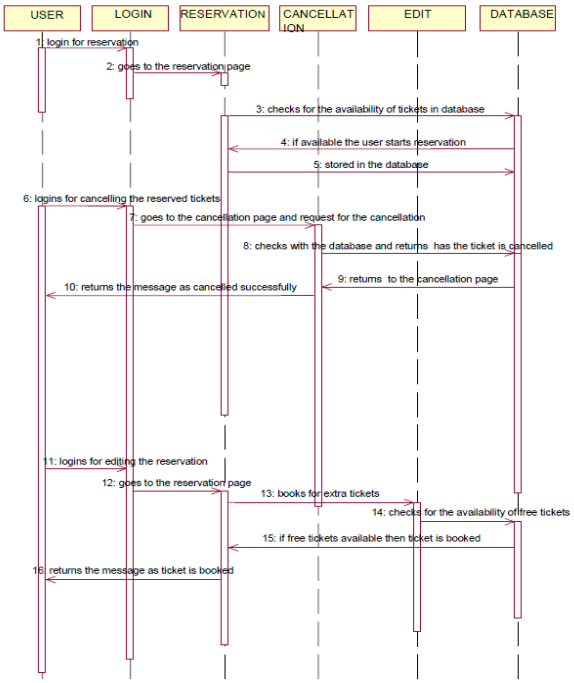

Figure2. Authorization

CONFIDENTIALITY

In this model the credit card number which is classified information is stored as such, which may lead to theft of money by the outsiders. There is not a need for the agencies or the service providers to have information about the credit card details of the user. In the systems designed so far when the user wants to make a payment for the reservation of tickets he gives his credit card number to the agency the agency takes the number as such and checks the balance for the particular credit card number provided. If the balance is adequate the amount is subtracted from the account and the expenses are paid for. Confirmation mail is sent to the user that the amount has been successfully taken from his account.

```
 <? xml version="1.0" ?>
 <sequence>
<servername>confidentiality</servername>
<entity name="user">
<action id="1">internalagent</action>
<inputcreditcardno="" type="text" />
<inputpinno="" type="text" />
<entity name="bank">
<action id="2">externalagent</action>
<inputcreditcardno="" type="text" />
<inputpinno="" type="text" />
</entity>
</entity>
</sequence>
```

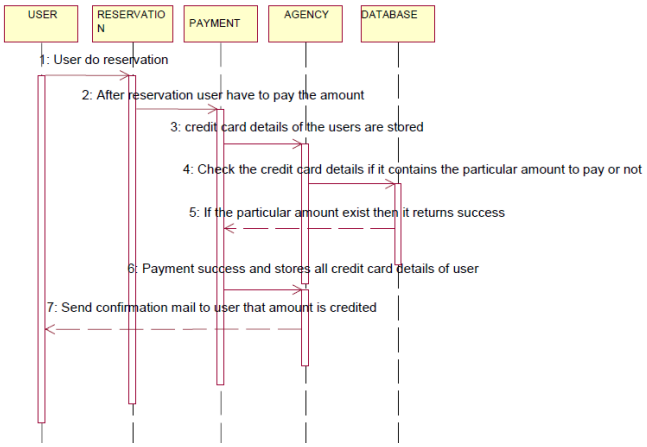

Figure3. Confidentiality

INTEGRITY

This delays the time of transactions and when a person has not decided to use the main services offered by the agency he must also be not allowed to use the sub services.

In the systems used so far cancellation of a service is not guarantee the cancellation of other services that come along with it. When the domain is for travels, the main service is booking the tickets and the subsidiary services are accommodation, site seeing and so on. When the user cancels a certain number reservation of seats, the agency requests the service provider to cancel only the tickets and not the additional sub services reserved by the user. Then the confirmation mail is sent to the user about the cancellation.

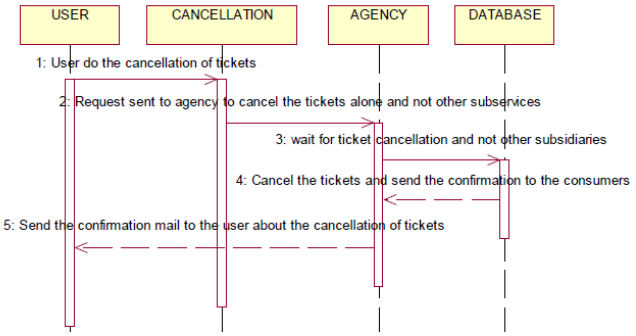

Figure4. Authentication

```
<Sequence>
<servername>integrity</servername>
 <entity name="user">
<Action>cancellation</action>
<inputname="" type="text" />
<inputticketnumber="" type="text" />
<inputairlinename="" type="text" />
<inputNooftickets="" type="text" />
<Action>roombooking</action>
<inputroomtype="" type="text" />
<inputroomnumber="" type="text" />
 <Action>cabbooking</action>
<inputdate="" type="date" />
<inputtime="" type="number" />
  <Action>foodservices</action>
```

```
<inputdate="" type="date" />
<inputtime="" type="number" />
<inputhotelpreferred="" type="name" />
 </entity>
</sequence>
```

NON REPUDIATION

After a user has successfully logged in into his account and books the services that he wants, he waits for the confirmation from the agency. The agency in turn waits for the confirmation from the service providers. The service providers take a long time to check for the availability of tickets and they send their reply to the end users via the agency. The user receives the confirmation of his services booked from the agency after a long time.

This usually results in the amount of time spent by the users waiting for confirmation of the services that they have availed. The identities of the receiver and sender are known but they are known only after a long delay in time.

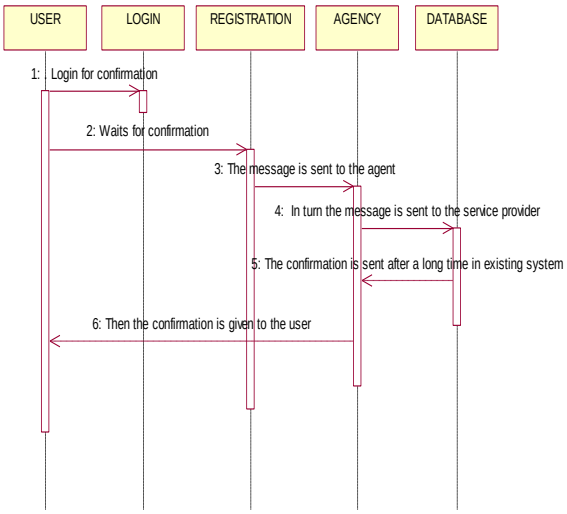

Figure5. Authentication

```
<? xml version="1.0" ?>
 <Sequence>
<entity name="agency">
<Action>confirmation mail</action>
<inputusername="" type="text" />
<inputmessage="" type="text" />
</entity>
</sequence>
```

5 ADD-ON SECURITY SERVICES

5.1 Authentication

Authentication is defined as the process of determining if a person is whom he claims to be. Authentication is done through the use of log in names and passwords. Knowledge of the password guarantees, the user's authenticity. Each user initially registers using a username and password of his choice or the one provided by the system. In our model, when the user gives a password it is stored in an encrypted form in the database. When the user enters his and password in the log in page, it is checked with the encrypted password stored in the database. Only if the password matches, the user is allowed to perform further reservations. If they do not match then the user is redirected to the log in page again.
The password is protected from the service providers by storing them in an encrypted manner in the database. The chances of password getting into the hands of unauthorized users are negligible.

5.2 Authorization

Authorization is the method of specifying access rights to the particular resources. It is the power or rights given to a person or group of persons to perform that particular task. This method ensures that an authenticated entity can access only those services they are allowed to access. Access control lists are used to implement this. In the systems existing so far minimum features like file authorization are provided. This performs access checks for requested ASP.NET resources using the original caller's security context. The original caller must be granted at least read access to the desired file. Windows ACLs on resources accessed by Web services (files, folders, Active Directories) include an Access Control Entry (ACE) that grants read access to the original caller. These services providing the minimum authorization features do not enhance the security of the site. The main feature in our model is that only the end users are provided with the authority to cancel the tickets that they have reserved. When any person attempts to cancel the tickets, the authorization of the person for that service is verified from the database. If the role of the person is a user then he is allowed to cancel the reservations. Otherwise he is denied from using that service. The time taken to cancel a service is greatly reduced and the rate of errors occurring during cancellation is also minimized.

5.3 Confidentiality

Confidentiality is the protection against the interception of sensitive information and transactions. This method of protection against information theft is known as encryption. The purpose of encryption in Web services security is to make the transmitted data inaccessible while it is being transmitted, ensuring the confidentiality of the data transmission.

Without confidentiality, attackers can eavesdrop to intercept messages and read all of the information. Classified information and transactions are frequently transmitted using Web services. Therefore, it is important to maintain a secure transmission so that eavesdropping by unauthorized parties is eliminated.

In the model we are going to design the password which will be provided to the agency will be stored in an encrypted format. Only the user is allowed to know the password and use it. The service providers and the agency will handle with only the encrypted credit card number. This number will be stored in an encrypted format in the database and decrypted when it is retrieved. The security issues are greatly addressed and the credit card number a classified information is handled in a secure manner. The access by unauthorized users is avoided.

5.4 Integrity

Integrity means that data cannot be modified undetected. It can be viewed as a special case of Consistency as in the ACID model. Integrity is violated when a message is

actively modified in transmission. Information security systems typically provide message integrity in addition to data confidentiality. In the systems used so far cancellation of a service is not guarantee the cancellation of other services that come along with it. When the domain is for travels, the main service is booking the tickets and the subsidiary services are accommodation, site seeing and so on. When the user cancels a certain number reservation of seats, the agency requests the service provider to cancel only the tickets and not the additional sub services reserved by the user. Then the confirmation mail is sent to the user about the cancellation. This delays the time of transactions and when a person has not decided to use the main services offered by the agency he must also be not allowed to use the sub services. In our model, when a user wishes to cancel his tickets he requests the agency, The agency in turn tells the service provider to cancel the main service as well as the subsidiary services. Confirmation mail is sent to the end users via the agency. So when a user cancels his service with the agency, the agency informs the cancellation to the service provider. So changes done at the user end are reflected in the provider end. Thus integrity is ensured in our model providing better security.

5.5 Non Repudiation

Both sender and receiver can provide legal proof to a third party (e.g. judge) that

- the sender did send the transaction, and
- the receiver received the identical transaction

Non repudiation guarantees that the message sender is the same as the creator of the message. This requirement prevents a party from denying it sent or received a message. Usually its implemented using Xml Digital Signatures.

6. PROPOSED WORKS

The business domain is a big industry in the world and travel agencies make up an important part of our economy. Agencies can access details about hundreds of package holidays, flights, accommodations and excursions. Travel agents have the expert knowledge to help their customers find and book a holiday package. Special agencies for booking family holidays, city sightseeing, tours, business travels are also present. Travel agencies are trained to give advice on destinations worldwide and make arrangements for transportation, hotel accommodation, car rental and sightseeing tours. Our paper is to create services for the complete travel domain starting from reservation to managing food and travel services and ending with the cancellation of services if required. The services we have decided to provide are Registration, log in, availability checking, reservation, room booking, food services, cab booking, mode of payment, cancellation and messaging services i.e. confirmation mail. The five entities are airline, hotel, local transport, bank and agency. These five entities are responsible for providing services. The airlines provide two major services of reservation and cancellation. Hotel is responsible for Room booking and availing of food services, the local transport for obtaining the mode of transport within the place like cab booking. The bank is the means by which the user pays the agency for the services that they are providing so hence it holds the mode of payment service. The last and most important entity agency is capable of providing registering service, login, checking the availability of tickets, messaging service. These are the services that are usually provided by any travel domain.

There are a lot many problems occurring in such travel services offered by an agency. There may be issues of security which result in the customer feeling insecure. This insecurity results ultimately in the loss of customers. This is a major setback for any agency whose goal is to increase the amount of customers. We propose to enhance the security of these sites. And we intend to do this by designing five add-on services (authentication, authorization, integrity, confidentiality and non repudiation).The services designed using Net BeansIDE6.1 are easy to create and to use. These services can be created within a matter of minutes. However the major disadvantage of Net Beans is that it employs several built in features which while making it faster also makes it difficult to incorporate and integrate the add on features. Services created using Net Beans make it impossible to integrate the security services that we have designed.

Creation of new services and integrating our services results in the time delay during which our customers may decide to move onto some other agency providing a constant service. This results in the loss of customer faith and trust in our service. Customer satisfaction is our key goal which is vital. So we must find a way to create the services using a different application and integrate our add on services designed into those services. This ultimately results in the usage of Sun Application Server 8.0.This application has its own merits. The services that are created using this application can be modified to accommodate our services. Even though the creation of these services takes a lot of time the services designed are more adaptable and flexible.

The services of registration, log in, checking availability requires the add on feature of authentication. Reservation, room booking, availing food services, cab booking require the authorization service. Mode of payment via the bank needs the confidentiality as a criterion. Authorization and integrity both are an add on feature for the cancellation service. Non Repudiation is a necessary add on for the messaging or confirmation service. The table indicating the responsibility of each organization to providing the desired add on service is given below

6.1 Authentication

Authentication is defined as the process of determining if a person is whom he claims to be. Authentication is done through the use of log in names and passwords. Knowledge of the password guarantees, the user's authenticity. Each user initially registers using a username and password of his choice or the one provided by the network. There afterwards, the user must use the password chosen by him to log in to his account. In our model, when the user gives a password it is stored in an encrypted form in the database. When the user enters his and password in the log in page, it is checked with the encrypted password stored in the database. Only if the password matches, the user is allowed to perform further reservations. If they do not match then the user is redirected to the log in page again. The password is protected from the service providers by storing them in an encrypted manner in the database. The chances of password getting into the hands of unauthorized users are negligible. We have evaluated the security of the service after integrating it with the add-on model and have found the user satisfaction to be high.

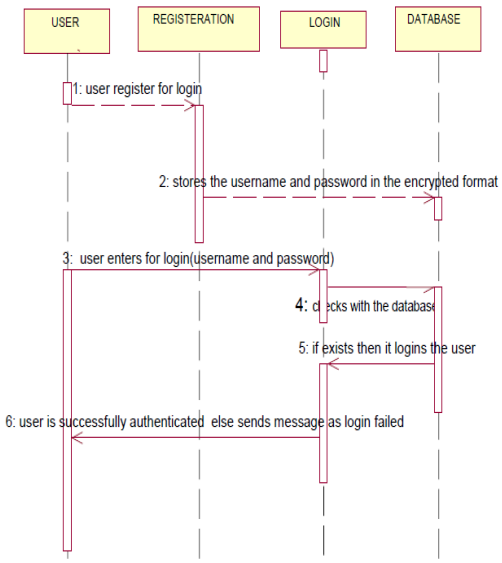

Figure6. Authentication

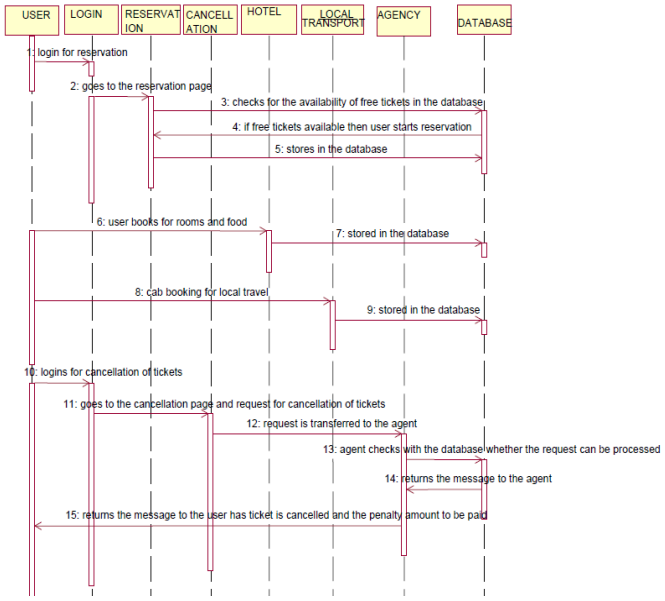

Figure7. Authorization

```
<? xml version="1.0" ?>
<Sequence>
<Server>authentication</server>
<entity name="user">
<action id="1">Registration</action>
<inputname="" type="text" />
<inputage="" type="text" />
<inputaddress="" type="text" />
<inputusername="" type="text" />
<inputpassword="" type="encryptedtext" />
< action id="2">Login</action>
<inputusername="" type="text" />
<inputpassword="" type="encryptedtext" />
</entity>
</sequence>
```

6.2 Authorization

Authorization is the method of specifying access rights to the particular resources. It is the power or rights given to a person or group of persons to perform that particular task. This method ensures that an authenticated entity can access only those services they are allowed to access. Access control lists are used to implement this. The time taken to cancel a service is greatly reduced and the rate of errors occurring during cancellation is also minimized. The main feature in our model is that only the end users are provided with the authority to cancel the tickets that they have reserved. When any person attempts to cancel the tickets, the authorization of the person for that service is verified from the database. If the role of the person is a user then he is allowed to cancel the reservations. Otherwise he is denied from using that service. After integrating the authorization constraint as add-on feature with the existing model, we have found the user satisfaction to be high.

```
<? Xml version="1.0"?>
<Sequence>
<servername>Authorization</servername>
<entity name="user" id="1">
<action id="1">Reservation</action>
<inputname="" type="text" />
<inputaddress="" type="text" />
<inputemailid="" type="text" />
<inputairlinename="" type="text" />
<inputfromplace="" type="text" />
<inputtoplace="" type="text" />
<inputnooftickets="" type="text" />
<inputdate="" type="date" />
<inputtime="" type="time" />
<entity name="agency" role="customer" access="true"
id="2">
<action id="2">Cancellation</action>
<inputname="" type="text" />
<inputticketnumber="" type="text" />
<inputairlinename="" type="text" />
<inputNooftickets="" type="text" />
</entity>
</entity>
</sequence>
```

6.3 Integrity

Integrity means that data cannot be modified undetected. it can be viewed as a special case of Consistency as in the ACID model. Integrity is violated when a message is actively modified in transmission. Information security systems typically provide message integrity in addition to data confidentiality. So when a user cancels his service with the agency, the agency informs the cancellation to the service provider. So changes done at the user end are reflected in the provider end. Thus integrity is ensured in our model providing better security. In our model, when a user wishes to cancel his tickets he requests the agency, The agency in turn tells the service provider to cancel the main service as well as the subsidiary services. Confirmation mail is sent to the end users via the agency.

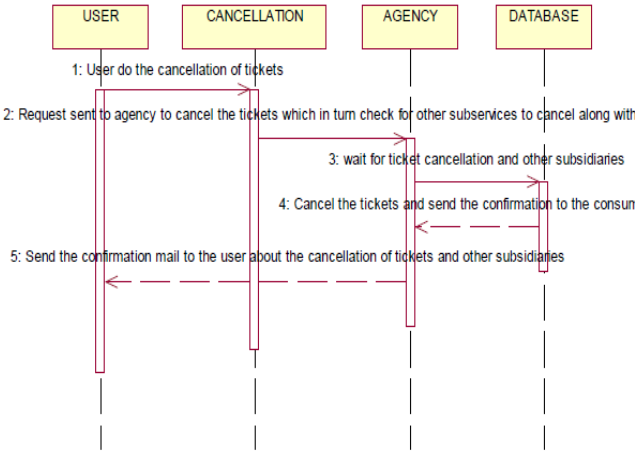

Figure8. Integrity

```
<sequence>
<servername>integrity</servername>
<entity name="user">
<action executionorder="1">cancellation</action>
<inputname="" type="text" />
<inputticketnumber="" type="text" />
<inputairlinename="" type="text" />
<inputNooftickets="" type="text" />
<action executionorder="2">roombooking</action>
<inputroomtype="" type="text" />
<inputroomnumber="" type="text" />
<action executionorder="3">cabbooking</action>
<inputdate="" type="date" />
<Inputtime="" type="number" />
<action excecutionorder="4">foodservices</action>
<inputdate="" type="date" />
<inputtime="" type="number" />
<in <inputhotelpreferred="" type="name" />
</entity>
</sequence>
```

The user satisfaction is found to be high when evaluating the security of the service after integrating it with this add-on model.

6.4 Confidentiality

Confidentiality is the protection against the interception of sensitive information and transactions. This method of protection against information theft is known as encryption. The purpose of encryption in Web services security is to make the transmitted data inaccessible while it is being transmitted, ensuring the confidentiality of the data transmission. Without confidentiality, attackers can eavesdrop to intercept messages and read all of the information. Classified information and transactions are frequently transmitted using Web services. Therefore, it is important to maintain a secure transmission so that eavesdropping by unauthorized parties is eliminated. In the model we are going to design the password which will be provided to the agency will be stored in an encrypted format. Only the user is allowed to know the password and use it. The service providers and the agency will handle with only the encrypted credit card number. This number will be stored in an encrypted format in the database and decrypted when it is the security issues are greatly addressed and the credit card number classified information is handled in a secure manner. The accesses by unauthorized users are avoided.

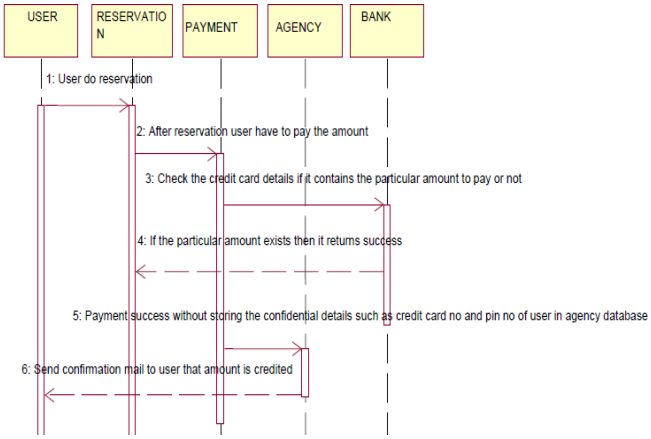

Figure9. Confidentiality

```
<? Xml version="1.0"?>
<Sequence>
<servername>confidentiality</servername>
<entity name="user">
<action id="1">internalagent</action>
<inputcreditcardno="" type="text" />
<inputpinno="" type="text" />
<entity name="bank" mode="hidden">
<action id="2">externalagent</action>
<inputcreditcardno="" type="text" />
<inputpinno="" type="encrypted text" />
</entity>
</entity>
</sequence>
```

We have evaluated the security of the service after integrating it with the add-on model and have found the user satisfaction to be high.

6.5 Non Repudiation

Non repudiation guarantees that the message sender is the same as the creator of the message. This requirement prevents a party from denying it sent or received a message. Usually it's implemented using Xml Digital Signatures.
Both sender and receiver can provide legal proof to a third party that

- the sender did send the transaction, and
- the receiver received the identical transaction

In this model, when the user reserves the tickets he needs he tells the agency the number of tickets needed. This request is immediately accompanied by the agency sending request to the service providers. The service providers after checking for the availability of tickets inform the agency immediately and this processed information if conveyed to the end user in a matter of few seconds. This results in the identity of the sender and the receiver being known at once. There is no time delay in the sending and receiving of transaction which greatly enhances the security of web services by correctly identifying the sender and receiver immediately.

```
<? Xml version="1.0"?>
<sequence>
<entity name="agency"messagehandler="email">
<action>confirmation mail</action>
<inputusername="" type="text" />
<inputmessage="" type="text" />
</entity>
</sequence>
```

We have evaluated the security of the service after integrating it with the add-on model and have found the user satisfaction to be high.

7.EVALUATION RESULTS

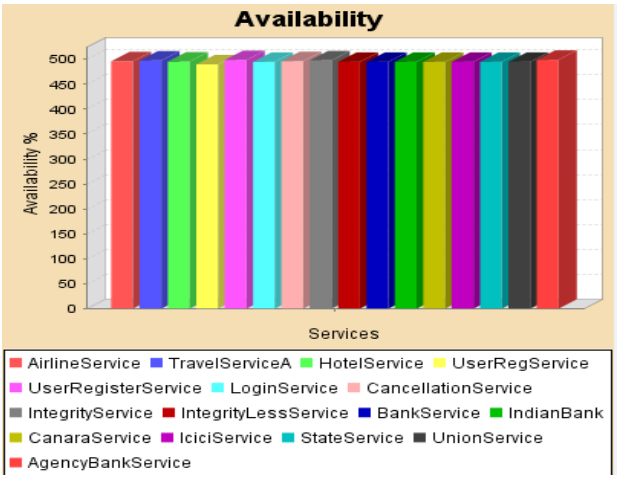

Figure10. Availability

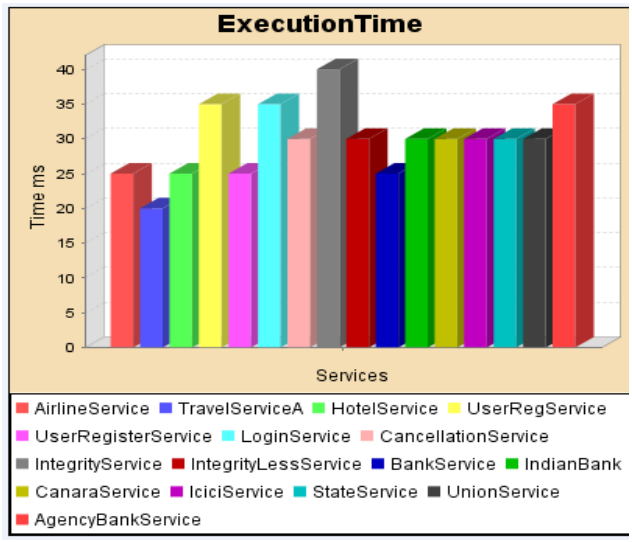

Figure11. Execution Time

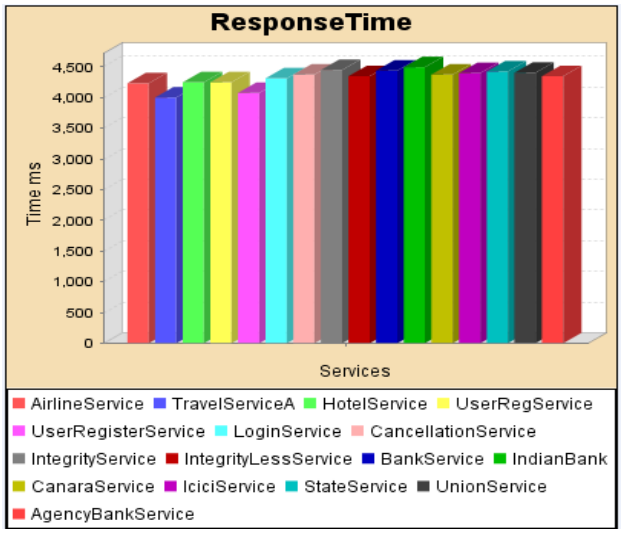

Figure12. Response Time

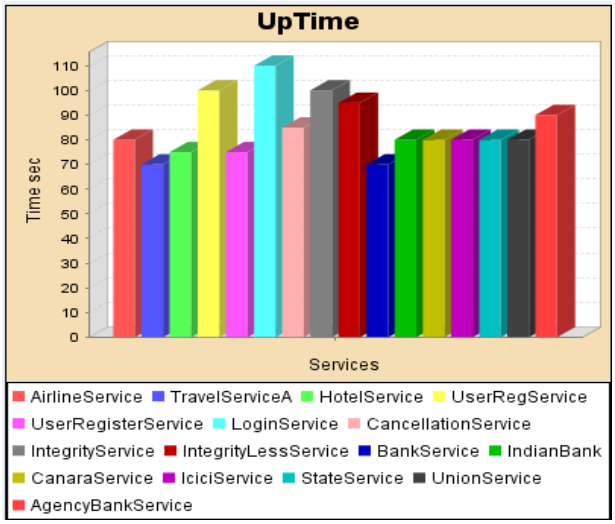

Figure 13.UpTime

	Services	Up time	Execution Time	Cost	Reliability	Response Time	Availability
EXISTING SERVICES WITHOUT SECURITY	LOGIN SERVICE	20	7	5200	0.881	863	99.0
	RESERVATION	15	5	5150	0.850	850	98.7
	INTEGRITY	19	6	5190	0.872	890	99.0
	BANK SERVICE	18	7	6564	0.860	890	98.0
	NON-REPUDIATION	16	8	5990	0.880	880	98.6
EXISTING SERVICES WITH SECURITY	LOGIN SERVICE	22	6	5500	0.880	850	99.7
	RESERVATION	20	7	5200	0.880	815	99.6
	INTEGRITY	20	8	5200	0.895	870	99.6
	BANK SERVICE	22	9	7654	0.888	888	99.0
	NON-REPUDIATION	19	7	6000	0.890	860	99.9

Figure14. Comparison of services with or without security

8 CONCLUSION

Implementing this model will turn out to be a boon not only to the travel agencies but also other business establishments engaged in similar consumer-driven leisure industries. The model driven approach to implement the security models as an add-on service greatly enhances the efficiency of the business services by making it secure, trustworthy and effective. The user's response to our approach has been tremendous and has been verified before and after integrating the security model with the business model.

9. REFERENCES

[1] Model-driven business process security requirement specification. Christian Wolter , Michael Menzel Andreas Schaad , Philip Miseldine , ChristophMeinel

[2] Expressive and Deployable Access Control in Open Web Service Applications.Claudio A. Ardagna, Sabrina De Capitani di Vimercati, Member, IEEE, Stefano Paraboschi , Eros Pedrini, Pierangela Samarati, Senior Member, IEEE, and Mario Verdicchio

[3] Christian Wolter, Michael Menzel, Andreas Schaad, Philip Miseldine and ChristophMeinel, "Model-driven Business Process Security Requirement Specification", (ELSEVIER) Journal of Systems Architecture 55, (2009) 211-223.

[4] ShaziaWasimSadiq, Guido Governatori, KioumarsNamiri," Modeling Control Objectives for Business Process Compliance", in BPM, 2007, pp. 149–164.

[5] Michiaki Tatsubori, Takeshi Imamura, Yuhichi Nakamura," Best-Practice Patterns and Tool Support for Configuring Secure Web Services Messaging", in ICWS, IEEE Computer Society, 2004. pp. 244–251

[6] N. Nagaratnam, A. Nadalin, M. Hondo, M. McIntosh, P. Austel, "Business-driven Application Security: from Modeling to Managing Secure Applications", IBM Syst. J. 44 (4) (2005).