Secured User Behaviour Based Access Framework for Web Service

R. Joseph Manoj¹, M.D.Anto Praveena², Manchikalapudi Anvesh³, Murarisetty Pujith⁴

¹Department of IT, St. Joseph’s College of Engineering, Chennai, India.
²³⁴School of Computing, Sathyabama Institute of Science and Technology, Chennai, India
rjmanoj79@gmail.com¹, antopraveena@gmail.com², anveshm123@gmail.com³,
puijth.murarisetty@gmail.com⁴

Abstract. There are many web applications around us providing their services using web services. In order to prevent malicious users from accessing the web services is an important challenge now. Hence this proposed access control model provides the user an effective way to prevent the malicious user by calculating the trust value of every user based on their behaviours such as success rate, failure rate, frequent of access, transaction timeout etc., in the web application. This model also notices and precludes IP address spoofing, SQL injection to allow only the authorized users and access appropriate information. Prototype implementation and simulation results show the proposed model is the effective one when it is compared with existing models.

Keywords: Web Services Access Control, User Behaviour, Trust Based Access

1. Introduction

An Access control framework of web service is to limit the web service user who tries to access the functions of the web services. Each model will have access control policies (rules) which are used to decide the credentials of specific user. Hence this is needed to know the movement of unauthorized users and grant the authorized users to access the web services. The following are the few traditional access control models [1]:

Role Based Access Control (RBAC) is the access control scheme which was widely used in web services. In RBAC, clients will be assigned roles which grants permission in order to provide an authorization to access a particular web service.

Attribute Based Access Control (ABAC) approach allows or denies the web services users based on their attributes such as location, role, attributes related to the environment.

Trust Based Access Control (TBAC) is varying from other access control models. In this model, web services access permission will be given based on the trust value of the user. Trust value will be calculated by the parameters such as user behaviours, success rate, failure rate etc., Hence, violation or misbehaving actions leads to decrease of trust value.

The following sections of the manuscript are organized as follows. In Section II, background concepts of trust based access control scheme is discussed. In Section III, the proposed work is presented. Section IV explains experimentation and performance analysis. Section V concludes the working of proposed access control model and future enhancement of system.

2. Background of Literature Survey

There are many researchers have done their research in web services access control models and it approaches. Some of the researchers follows Naive Bayesian algorithm cantered trust calculation model [Hien et al 2010] for the selection of web service. There are three main sources for trust calculation such as QoS, direct experience and user recognition.
Vivekananth developed a user behaviour based trust calculation model that describes the reliability of user behavior [2]. A trust threshold cost is used to grant permission or ignore the users from accessing the web services. If the value of trust is greater than threshold then the access will be allowed or denied. Trust ranges from 0 to 1.

Stefания Galizia proposed a trust based access scheme to access web service [3]. It has Third Party centered method for the grouping the web services based on the support of internet service tool.

Wang Meng presented a recommendation credibility based trust model [4] for web services access. This method varies honest and dishonest recommendation and dynamically corrects the value of trust.

Shangzhu Jin developed an access scheme in which trust value is computed using the factors time decay and feedback [5]. It is not dealing with assigning trust value for the newly registered users.

Surya Nepal proposed a trust management method based on trust to access web service [6]. But they failed in addressing issues like dishonest, propagation, trust bootstrapping.

Shashi Bhanwar proposed an entrance control demonstrate in light of trust by deciding notoriety and trust of the area on the premise history of past exchanges and appraised value from the feedback [7].

Srivaramangai proposed a model based on reputation of the users to improve consistency of transactions [8]. Consistency may be achieved by forming reciprocal trust between the service provider and requester.

Cesar Ali proposed another trust framework [9] to get to the web administrations in light of setting and part of the administrations requester. Here too they neglected to deal with new client trust esteem viably.

The above web services access models have not handled newly registered users trust value calculation effectively and study few parameters to evaluate the trust value for the user who request service. Also it is not considering the failed transaction due to server errors. Proposed work handles the issues by incorporating the trust value calculation using multivariate such as timeout, access frequency, network conditions, and success and failure rate. Also this proposed system has additional features such as IP address spoofing and SQL injection prevention to tighten the security features.

3. Proposed System

The proposed framework design is given in the Fig.1. It is partitioned as 1. Web Service requester is the individual or framework which gets benefits from web services 2. Web service provider will give the support to the requester and deals with the trust calculation. The different sub processes of service provider are given below.
3.1 New User Registration

This component is used for registering new users and it has two sub processes.

1. User Registration: All new users have to register their environment and personal details with the web service provider to access the web service. While the new user registration, service requesters will be given unique id and passcode to use the web service. Once the registration is done, the new user details will be stored and request will be sent to SQL Injection Manager.

2. Registered Users Data: This is a repository where the new users data like id, passcode, IP address of last transaction, date and time of last transaction, requested pages, number of hits, transaction status code like ‘400’, ‘408’ and ‘200 and details of new users such as name, address, contact details etc., Later these data will be used in the authentication process.

3.2 SQL Injection (SQLI) Manager

This SQLI Manager Module detects and prevents SQL Injection occurs at web service provider side. After user registration process is completed the SQL request will be passed to the SQLI Manager. It has following sub processes to detect and prevent SQL Injection [13].

1. XML Generator: This process will be analysing the intercepted SQL query and generate XML file by converting the input values from SQL Query to detect SQL injections are present. This newly generated file will be used in validation process in order to detect the SQLI vulnerabilities. For example, assume the SQL query select * from userdet where uname='Mathew' OR 1=1 - - 'password=' for login into the applications. The generated XML Code after the interruption of SQL query is shown in Fig 2 which has input values given in query. This file will be validated to check the presence of SQL Injection.

```xml
<? xml version="1.0" encoding= "utf-16"?>
<sqlexper>
  <input>Mathew</input>
  <input>'1' ="1</input>
</sqlexper>
```
Fig 2: Sample XML file for the SQL Query

2. XML Validator: The Validator process is validating the XML file. Before validating XML file, XSchema will be created to verify the SQL injection depends upon the input parameters of XML file. Sample XSchema of given XML is given in following Fig.3:

```xml
<?xml version="1.0" encoding="utf-16"?>
<xs: element name="input1">
  <xs: simpletype>
    <xs: restriction base="xs: string1">
      <xs: pattern value="[a-zA-Z0-9]"> </xs: restriction>
    </xs: simpletype>
  </xs: element>
</xs: element>
<xs: element name="input2">
  <xs: simpletype>
    <xs: restriction base="xs: string1">
      <xs: pattern value="[a-zA-Z0-9]"> {8} </xs: restriction>
    </xs: simpletype>
  </xs: element>
</xs: element>
</xml>
```

Fig 3: XSchema File for XML Validation

The above XSchema file as in Fig 3 used to validate the XML file to detect the SQL injection. But this Schema can detect only tautology type SQL injection. If the XML validation is success then process concludes that there is no SQL injection and process is allowed to move the next authentication process. If failure occurs then this process will be terminated the user.

3.3. Authentication Manager

1. Data Categorization: The data such as uid, passcode, URL Requested, service requester IP Address are segregated from server log file and call the authentication process by sending the filtered data.

2. Origin & IP Spoofing Identification: The responsible of this process is to recognize Ingress Packet filtering based IP spoofing. Ingress packet filtering verifies the network range of IP address. If it violates the range of network address then user will not be allowed to access the web service.

3. Authentication: Final authentication will be taken place in this process by evaluating and sieving the details from of server. once it gathers the details of server log of requesters, it recovers the prior history of requester from the registered user data using user id. The authentication process has three cases as shown in the given algorithm [15]. Authorization processes will be initiated by passing user id, passcode, IP address and Requested URL Page once the user is authenticated.
UserAuthent (usrid, pcode, ipadd, requrlpage)
{
  Cat 1: If usrid, pcode are legal and ipadd is registered and no spoofing
      Step 1: Permit to authorization interface.
  Cat 2: If usrid or pcode is not valid and ipadd is registered and not spoofed
      Step 1: Verify the requrlpage is visited
      & Total hits > threshold value &
      Past incorrect req< threshold value If yes
      print “Enter name and password correctly”
      Else
      print “Access Denied”
  Cat 3: If usrid or pcode is not valid and ipadd is not registered
      Step 3: Deny to access web service.
}

Fig 4: User Authentication Algorithm

3.4 Authorization Manager

It is playing a vital role in user authorization which decides whether to permit or deny the user from accessing of web service based on the trust value calculated by trust manager. Trust manager is the process which evaluates the trust value of requesters dynamically. This model defines the values trust from 0 to 10 and threshold level of trust as 0. If each user trust value is greater than the threshold level, then service user may access the web services else will be ignored. The trust manager is described follows [16].

1. Trust Assessment Point (TAP): This is the initial process of trust manager which verifies the trust value initially and avoids the user. TAP allows or denies user from accessing web services based on current trust value. If the user is new to the system, TAP will assign 0 as trust value and allows the requester to avail the web service.

2. Trust Managing Point (TMP): It is dynamically computes the value of trust based on below parameters that are accessed from the database of registered users and the types of users.

1. Success Rate (St) is the count of transactions of a user which are success during the specific period of time. Service provider will define the range of time.

2. Failure Rate (Ft) is the count of failed transactions of a user for a quantified period of time.

3. Frequency of Access (Af) is the frequency value of accessing the service by the requester. If the Af value of requester is lesser than threshold level which was assigned by service provider, then the service requester can be considered as lazy requester.

4. Time-out (To) bethe count of time-out occurs while requester is accessing the resource. It is used to identify the honest users.

5. Average Time (At) is the time spent while requester accessing the web service. It is also used to identify the honest requesters.
TMP classifies the requesters into four kinds in light of their conduct in the framework. They are 1.Honest and Active 2.Dishonest and Active 3.Dishonest and Active 4.Dishonest and idle. In the event that the time out (To) and Average time (At) are more noteworthy than its edge esteem (Vo), at that point requester is accepted as genuine else untrustworthy requester. In the event that recurrence of access (Af) is more noteworthy than its limit level, at that point the requester is considered as dynamic requester else idle requester. pf determines punishment factor. Framework expect punishment factor as 0.5. The proposed framework has following trust value computing techniques.

**Type 1** If the requester is perceived as genuine and active, then requester’s trust value (dt) will be increased as given below:

\[
dt_{t+1} = \frac{(dt_t + (St+Af)*dt_t)}{(St+Ft)}
\]  

**Type 2:** On the off chance that the requester is perceived as fair and idle or deceptive and dynamic, at that point requester’s time decay value (dt) will be diminished as below:

\[
dt_{t+1} = \frac{(dt_t - (St+Af)*dt_t)}{(St+Ft)}
\]  

**Type 3:** In this type the user is realized as inactive and dishonest, then user’s decayed trust value (dt) will be decreased by subtracting penalty factor as follows:

\[
dt_{t+1} = \frac{(dt_t - (St+Af)*dt_t)}{(St+Ft)} - pf
\]

TMP checks the whether any server disappointments happens in past exchange. On the off chance that there is a server disappointment, TMP sends demand to TNP for trust esteem transaction with the goal that legit clients won’t be rebuffed.

3. **Trust Negotiation Point (TNP):** This process is to negotiate the trust value of service requester who is affected with server failures. The following two factors will cause the trust value to negotiate.

1. **Server Error rate (Se)** is the value to identify the count of time-out or disconnect occurs in the transaction occurs due to Server failure. Since the server error occurs sometimes, the value of trust will be lessened to honest requester. To evade the problems, TNP checks whether the past access has correspondence failures because of server.

2. **User class (Uc)** identifies the class of user and reward them based on their consistency. Platinum user is a user who maintains trust value of 7 and above consistently for last 3 months.

Consequently if there are any server disappointments or customer continues with platinum customer class, TNP will counsel by including Negotiate Factor (Nf) of 0.5 with the trust esteem got from TMP. Game plan process will orchestrate honest to goodness and dynamic customers for 3 times to stimulate their honest to goodness and liveliness. At long last TNP sends last trust an incentive to TDP which sends the incentive to approval interface. Finally Authorization Interface picks whether to allow or deny the requester to get to the entrance of web benefit in light of the incentive from the Trust Manager.

4. **Simulation and Result Analysis**

To check the performance of the proposed system and obtain the experimental data, a framework was created. An e-Library application was developed on top level of the framework and implemented the functionality of the proposed system. A web service was created at low level of framework to search and download the necessary books. The e-library web application was tested in LAN environmental setup for 15 days with about 250 registered users. Simulation result was recorded and compared with existing models.
4.1. Experiment 1

In experiment 1, how the system is responding to handle malicious user based on ingress packet filtering technique for definite days. Hence the proposed work was purposely assigned few specific numbers of attackers to spoof IP address. The resultant data of experiment have been given in Table 1.

Table 1: Fake Fatalities Identification

| Days | Victims Count | Days | Victims Count |
|------|---------------|------|---------------|
| 1    | 21            | 8    | 21            |
| 2    | 25            | 9    | 23            |
| 3    | 25            | 10   | 24            |
| 4    | 31            | 11   | 27            |
| 5    | 43            | 12   | 40            |
| 6    | 43            | 13   | 40            |
| 7    | 38            | 14   | 36            |

4.2. Experiment 2

In the analysis 2, consider the values [12] of \( dt=0.68 \), \( Ft=95 \), \( St = 75 \) and Frequent of access (Af) ranges from 15 to 100. Frequent of access (Af) edge an incentive for a particular period is accepted as 100. Here To, Af are defined that they are not as much as their limit esteem and Penalty factor (Pf) value is 0.5. Resultant trust esteems in view of frequent of access (Af), Time out (To), Access Time (At) and Penalty Factor (Pf) are recorded in Table 2.

Table 2: Result of Including Multifactor in Trust Calculation

| Access Frequency (Af) | Trust Value Based on Af | After including To, At with Af | After Including Af with To, At, Pf |
|-----------------------|-------------------------|---------------------------------|---------------------------------|
| 15                    | 0.43                    | 0.40                            | 0.39                            |
| 25                    | 0.47                    | 0.46                            | 0.41                            |
| 35                    | 0.43                    | 0.39                            | 0.37                            |
| 45                    | 0.48                    | 0.41                            | 0.39                            |
| 55                    | 0.50                    | 0.59                            | 0.47                            |
| 60                    | 0.54                    | 0.51                            | 0.47                            |
| 70                    | 0.57                    | 0.54                            | 0.50                            |
| 80                    | 0.61                    | 0.59                            | 0.54                            |
| 90                    | 0.63                    | 0.60                            | 0.57                            |
| 100                   | 0.65                    | 0.76                            | 0.62                            |
From the data shown in table 2, it is concluded that value of trust is increased as value \(Af\) is increased. It implies that if the client is dishonest and inert, trust esteem will be influenced. The qualities from this table have been taken for the further investigation done beneath and demonstrated this proposed framework is limiting vindictive client effectively.

From the chart in Fig 6, we can see that the trust estimation of noxious requesters will be consistently diminished if the requester's frequent of access (\(Af\)) is lower than edge esteem generally trust esteem will be expanded. Henceforth we can reason that the lower the estimation of \(Af\) is the quicker decrease for the estimation of \(Dt\), which viably urges the requester to give legit get to and keep away from noxious access. Nonetheless, those pernicious practices can't be illegal essentially by Access recurrence.

### 4.3. Experiment 3

In this experiment, Trust values were ascertained utilizing the recipe and recorded in Table 3 in light of estimations of \(St\), \(Ft\) of the client. Likewise estimations of \(Af\), \(To\), \(At\) and \(Pf\) are considered to figure the esteem.

| \(St\) | \(Ft\) | \(Af\) | \(St,Ft,Af\) To Based (\(Dt\)) | \(Af,To\) At Based on \(Dt\) | \(Af,To, At, Pf\) Based \(Dt\) |
|------|------|------|-----------------|-----------------|-----------------|
| 15   | 55   | 15   | 0.328           | 0.31            | -0.233          |
| 25   | 45   | 25   | 0.556           | 0.521           | 0.05            |
| 35   | 35   | 35   | 0.834           | 0.81            | 0.42            |
| 45   | 25   | 45   | 1.16            | 0.92            | 0.52            |
| 55   | 15   | 55   | 1.46            | 1.19            | 0.79            |

It can be concluded that the rate of success will be increased as trust value also increased or value will be decreased. In Table 3 above, \(Dt\) is computed based on the factors \(St\), \(Ft\), \(Af\), \(To\) and \(Pf\) and values are recorded.
The chart in Fig 7 demonstrates the adjustments in trust an incentive as indicated by multifaceted, for example, St, At, Af, To and At. This diagram has been drawn in light of 3 sets of information 1. In view of St, Ft, Af and To 2. In light of St, Ft, Af with To, At 3. In the view of St, Ft, Af with To, At, Pf. This diagram presumes that untrustworthy and idle requester's trust esteem is relentlessly diminished when trust esteem is computed utilizing the multifaceted, for example, St, Ft, Af with To, At, Pf. Likewise discovers that if service user is straightforward and dynamic esteem develops gets expanded. In the wake of including Penalty factor, untrustworthy and latent client trust esteem will be corrupted tremendously.

![Fig 7. Multifactor Effect in trust computation](image)

In any case, Honest and dynamic users might be viewed as dishonest and idle user in view of server disappointments. To maintain a strategic distance from this issue rebuffed requester trust esteem will be included with Negotiate factor (Nf) so it will energize requesters. In our framework, 0.5 was set as Negotiate factor (Nf). Likewise if requesters achieve most extreme estimation of 10 and keep up a similar incentive for one month, those users may be identified as platinum client. Transaction process will likewise be improved the situation these clients for three times to support their genuine and liveliness.

### 4.4 Performance Analysis

This exhibits a correlation of proposed approach with prevailing web administrations get to control access models. Table 4 delineates the qualities and highlights upheld in the proposed access method and prevailing methods. The accompanying examination mirrors that our approach utilizes a system to appoint get to percentile to web administrations and utilizations. The chart in Fig 7 demonstrates the adjustments in trust an incentive as indicated by multifaceted, for example, St, At, Af, To and At. This diagram has been drawn in light of 3 sets of information 1. In view of St, Ft, Af and To 2. In light of St, Ft, Af with To, At 3. In the view of St, Ft, Af with To, At, Pf. This diagram presumes that untrustworthy and idle requester's trust esteem is relentlessly diminished when trust esteem is computed utilizing the multifaceted, for example, St, Ft, Af with To, At, Pf. Likewise discovers that if service user is straightforward and dynamic esteem develops gets expanded. In the wake of including Penalty factor, untrustworthy and latent client trust esteem will be corrupted tremendously.
In any case, Honest and dynamic users might be viewed as dishonest and idle user in view of server disappointments. To maintain a strategic distance from this issue rebuffed requester trust esteem will be included with Negotiate factor (Nf) so it will energize requesters. In our framework, 0.5 was set as Negotiate factor (Nf). Likewise if requesters achieve most extreme estimation of 10 and keep up a similar incentive for one month, those users may be identified as platinum client. Transaction process will likewise be improved the situation these clients for three times to support their genuine and liveliness.

Since it prevents SQL attack (Injection) and IP spoofing initially, the access to resources is lessened which brings about diminished approval work amid serving of access demands.

Table 4: Existing and Proposed Model Comparison

| Characteristics                        | RBA | ABAC | Proposed system |
|----------------------------------------|-----|------|-----------------|
| Distributed environment support        | Yes | Yes  | Yes             |
| Trust level assignment                 | No  | No   | Yes             |
| Reduced authorization work for access requests | No  | No   | Yes             |
| Detect SQL Injection                   | No  | No   | Yes             |
| Detect IP Address Spoofing             | No  | No   | Yes             |
| Policies specification & Maintenance   | Simple | Complex | Simple |
At long last the execution investigation of proposed framework was led with present conventional service control model, such as RBAC, ABAC and result [13] is appeared in Fig 8. Execution examination says that how effectively all the current models are confining malevolent clients or conduct. The investigation inferred that proposed show confines a larger number of clients than customary access control models.

![Comparing Security Parameter With Existing System](image)

**Fig 8:** Comparing Security Parameter With Existing System

5. **CONCLUSION**

In this framework, a novel approach of web administrations get to control access of web services is proposed. Likewise talked about the energy of multifaceted, for example, Access recurrence, Time of access and Time out on trust an incentive to get to web benefit are examined. In the wake of presenting the SQL Injection and IP address parodying checking, security highlight is fix in the proposed framework. As an outcome, the security of web administrations got enhanced and this new model is proficient to rouse benefit requesters to participate in get to control framework effectively and sincerely. At last by investigating the execution of the proposed framework it is reasoned that the proposed demonstrate capacity to recognize and keep the malignant clients or practices is superior to any conventional access models.

This framework will find new ideas to implement the authentication and authorization process more efficiently. It will also find the new parameters to calculate the requesters trust value more effectively. Framework will locate the diverse method to isolate the clients as indicated by their trust esteems with the goal that legit and dynamic clients will get more advantages.

**References**

[1] R. Joseph Manoj, Dr.A.Chandrasekar, “An Literature Review on Trust Management in web services access control”, International Journal on Web Services Computing, Vol 4, No 3, pp.1-19, Sep 2013.

[2] Vivekananth, “A Behavior Based Trust Model for Grid Security”, International Journal of Computer Applications, Volume 5– No.6, 2010

[3] Stefania Galizia, Alessio Gugliotta and John Domingue,”A Trust Based Methodology for Web
Service Selection”, International Conferences on Semantic Computing, 2007.
[4] Wang Meng, Hongxia Xia, Huazhu Song, “A Dynamic Trust Model Based on Recommendation Credibility in Grid Domain”, International Conference CiSE, 2009.
[5] Usha Nandini, Saravanan M, Albert Mayan J, Murari Devakannan Kamalesh, Mohana Prasad K (2018), "Automatic traffic control system using PCA based approach", International Conference on Energy, Communication, Data Analytics and Soft Computing, pp.2387-2392.
[6] Surya Nepal, Wanita Sherchan and Athman Bouguettaya, “A Behaviour-Based Trust Model for Service Web, IEEE international Conference on Service Oriented Computing and Applications 2010.
[7] Bhanwar, S, Bawa, S, “Establishing and Evaluating Trust in a Grid Environment”, 10th International symposium ISSPAN’09, 2009.
[8] Renagaramanujam, Srinivasan, Srivaramangai P, “A Comprehensive Trust Model for Improved Reliability in Grid.”, International Journal of Computer Applications Vol 5, No.7, August 2010
[9] Cesar Ali “CATRAC: Context Aware Trust and Role based Access Control for composite web services” 10th IEEE international Conference on computer and information technology (CIT 2010).
[10] Hien Trang Nguyen, Weiiang Zhao, Jian Yang, “A Trust and Reputation Model Based on Bayesian Network for Web Services”, 2010 IEEE International Conference on Web Services.
[11] Asha P, Albert Mayan J, Canessane A (2018),“Efficient Mining of Positive and Negative Itemsets Using K-Means Clustering to Access the Risk of Cancer Patients”, Communications in Computer and Information Science, ICSCS 2018, Kollam, 2018, pp.373-382.
[12] X. Wang, J. Luo, A. Song and T. Ma, “Semantic Access Control in Grid Computing”. Proc. 11th International Conference on Parallel and Distributed Systems, 2005
[13] Shanshan Song, Kai Hwang, and MikinMcwan, “Fuzzy Trust Integration for Security Enforcement in Grid Computing” NPC ‘04, LNCS 3222, pp. 9-21.
[14] R. Joseph Manoj, Dr.A.Chandrasekar, M.D.Anto Praveena, “An Approach to detect and tautology type SQL injection in web services based on web services based on XSchema Validation”, International Journal of Engineering and computer science, Vol 3, No 1, pp.3695-3699, Jan 2014.
[15] R. Joseph Manoj, Dr.A.Chandrasekar, “An Authentication System of web services based on Web Server Log Analysis” International Journal of Engineering and Technology,Vol 5, No 6, pp.4786-4793, Jan 2014.
[16] R. Joseph Manoj, Dr.A.Chandrasekar, “An Access control model of web services based on multifactor trust based management, International Review on Computers and software”, Vol 8, No 10, pp.2460-2466, 2013.
[17] Albert Mayan J, Anto Praveena M D, Telkar Bharath Rao, Uravakonda Uday Sagar (2018), "Optimized test data generation over suspicious implementation of oracle problem", IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, pp.2559-2563.