A WEB SERVICES AUTHENTICATION SYSTEM BASED ON WEB SERVER LOG ANALYSIS

M. D. Anto Praveena*, Dr. R. Joseph Manoj2

1Assistant Professor, Sathyabama University, Chennai, India.
2Associate Professor, St. Joseph’s College of Engineering, Chennai, India.

Email: antopraveena@gmail.com

Received on 17-07-2016 Accepted on 15-08-2016

Abstract

Authentication is a method which validates users’ identity prior to permitting them to access the web services. To enhance the security of web services, providers follow varieties of authentication methods to restrict malicious users from accessing the services. This paper proposes a new authentication method which claims user’s identity by analyzing web server log files which includes the details of requesting user’s IP address, username, password, date and time of request, status code, URL etc., and checks IP address spoofing using ingress packet filtering method. This paper also analyses the resultant data and performance of the proposed work.

Keywords: Authentication, Web service Authentication, Web server log, Web service Security.

1. Introduction

Web service is an emerging technology in web arena where web methods can be described and published by service providers and the same can be accessed by other web programs or service requesters over the web. With the emergence of web services technology, vitality and liveness are becoming the nucleus characteristics of interorganizational business processes such as business process integration, distributed auction services and order processing [1]. However such characteristics also bring problems in security of web services such as authentication, authorization and integration. Hence It leads malicious users may violate the system and access the service. This may leads to collapse the part or entire system. To prevent malicious users, web service has different security mechanisms, such as authentication, authorization, access control policies, certification etc. In the security mechanisms, authentication is a crucial security method since it identifies the requesters who claim to access the web service. Since web services are playing important role in commercial
web applications such as online banking, online trading, developing an efficient authentication and authorization system is inevitable.

The security of web services can be split into numerous parts. One of the parts is the authentication, the detection and verification of the authenticity of a user. Traditional authentication was performed by sending a username and password. This method is still used widely, but lack of functionality and liveness required for web services. A difference has to be made to the subparts of security systems. Authentication is, as mentioned before, the detection of the authenticity of a user request to access the resource. Authorization is differing from authentication by controlling the permission of a party. In other words, which parts of the system are accessible by the already authenticated client or authenticated server. Authentication is the first step required to a security system which detects the true identity of the party. After the authentication has been taken place, the second step, authorization can start giving access to the subsystems. The lack of sufficient authentication methods for large web service systems has stalled the development of the services. Recently, authentication methods have been developed by several groups of researchers to solve this problem. However, there is no overview of these methods and they have not been compared to each other yet. All methods seem to have some weaker points in their implementation, but a real comparison has not been made. Web servers also use a range of secondary authentication mechanisms [13], such as sending users an email with an access key, sending users an SMS message with an access key, asking users to answer a security question, asking users to supply an old password and asking a friend or other third party to verify users’ identity to improve the effectiveness of authentication process.

The contents of the paper is ordered as follows, section II explains web log file and its structure, status codes of Hyper Text Transfer Protocol, section III briefs related work in web services authentication, section IV explains proposed authentication system based on web server log file, in section V the experimental results are analyzed and proposed system performance is analyzed in section VI. Conclusions and future work are mentioned in section VI and VIII.

2. Web Server Log

A Web server log, located in web server is the file stores activity details such as IP address, username, password, date and time of request, URL of web user when a request submits to a web server. These data can be combined into a single file, or separated into distinct logs, such as an access log, error log, or referrer log. However, server logs typically do not collect user-specific information. A statistical analysis of the server log may be used to examine traffic patterns by time of
day, day of week, referrer, or user agent. The main source of raw data is the web access log which shall be referred as log file. This information is recorded in chronological order.

### 2.1 Web Log Structure

The following is a fragment from the server logs for loganalyzer.net [2].

```
196.109.55.102  xyz  abc  [08/Oct/2013:04:54:20 -0400] "GET /about.html HTTP/1.1" 200 11179 "Mozilla/5.0(compatible;Googlebot/2.1;+http://www.google.com/bot.html)".
```

This log file structure reflects the information of web user as follows:

- **Remote IP address or domain name**: An IP address is a 32-bit host address defined by the Internet Protocol; a domain name is used to determine a unique Internet address for any host on the internet. One IP address is usually defined for one domain name.
- **Username and password** if the server requires user authentication.
- **Entry and exit date and time**.
- **Modes of request**: GET, POST or HEAD method of Common Gateway Interface.
- **Status codes**: It’s a part of log files which specify error conditions as well as successful communication of data. The HTTP status code returned to the client.

Few of common status codes are listed in Table 1.

### Table 1: Status Codes of HTTP.

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Description</th>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK</td>
<td>408</td>
<td>Request Time-out</td>
</tr>
<tr>
<td>201</td>
<td>Created</td>
<td>409</td>
<td>Conflict</td>
</tr>
<tr>
<td>202</td>
<td>Accepted</td>
<td>413</td>
<td>Request Entity Too Large</td>
</tr>
<tr>
<td>203</td>
<td>Non-Authoritative Information</td>
<td>414</td>
<td>Request-URL Too Large</td>
</tr>
<tr>
<td>305</td>
<td>Use Proxy</td>
<td>500</td>
<td>Server Error</td>
</tr>
<tr>
<td>400</td>
<td>Bad Request</td>
<td>502</td>
<td>Bad Gateway</td>
</tr>
<tr>
<td>401</td>
<td>Unauthorized</td>
<td>503</td>
<td>Out of Resources</td>
</tr>
<tr>
<td>404</td>
<td>Not found</td>
<td>504</td>
<td>Gateway Time-Out</td>
</tr>
</tbody>
</table>

- **Bytes**: The content-length of the document transferred.
• Remote log and agent log.

• Remote URL and Requested URL

• “request:” The request line exactly as it came from the client

Log files are plain text (ASCII) files that are independent from the server platform. There are some distinctions between server software, but traditionally there are four types of server logs: Transfer Log, Agent Log, Error Log and Referrer Log. The first two types of log files are standard. The Referrer and Agent Logs may or may not be “turned on” at the server or may be added to the Transfer log file to create an “Extended” Log File format.

3. Related Work

This section presents the Extract of web services authentication schemes which were proposed by different researchers. Hada.S et al (2002) [3] present a design for a session-oriented, multi-party authentication protocol based on standard Web service technologies such as SOAP, XML-Signature/Encryption, and SOAP-DSIG. The protocol consists of a message authentication protocol and a session management protocol.

Yu Shung et al (2008) [4] discuss the issue of online user authentication and propose a method for online user authentication employing trusted computing technology. These works describe a browser extension scheme, which transparently produces a certificate for each user, improving web authentication security and defending against password phishing and other attacks. Since the scheme combines the password entered by the user, the password associated with private key protected by trusted platform module, and user certificate provided by trusted computing platform, stealing only the password at web will not have an effect on user security. And no changes on the server side are required in the scheme. The proposed approach could be proved to protect against phishing attacks.

Dacheng Zhang et al (2004) [5] present a new protocol design for multiparty authentication in which each service instance of a given session is provided with a unique identifier. The coordinated atomic action scheme is exploited for achieving an improved level of threat containment. They evaluate the scalability of their design by means of both experiments and an analytical model. Hung-Min-sun et al (2012) [6] design a user authentication protocol named oPass which leverages a user’s cellphone and short message service to thwart password stealing and password reuse attacks. oPass only requires each participating website possesses a unique phone number, and involves a telecommunication service provider in registration and recovery phases. Through oPass, users only need to remember a long-term password for login on all
websites. Shim, S.S.Y et al (2005) [7] describes access these resources, either via the public Internet or private intranets, users must verify their digital identity. This can range from a simple user-name-password combination to biometric data such as fingerprints to physical objects like hardware tokens and smart cards. Federated identity management would enable individuals to interact with various service providers or Web sites with trust relationships by signing in just once.

Nian Lu et al (2008) [8] propose a security mechanism that deals with the requirements of authentication, integrity, no reputation, and confidentiality across the communication process based on WS-Security and the two security tokens.

Barry Nijkamp (2006) [9] proposed Multiparty Authentication is clearly the support of multiple clients and Services. Multiparty Authentication is the only frequently used protocol that supports the delegation of both the client as the service role.

There is a problem with the Multiparty Authentication. As every delegation to another client or server is done with the Session Key, a big security issue rises. If the key is lost to an untrusted third party, that party can take over the complete session, either the client or the Service. Such an intruder can act as a new server or a new client and take over the complete handling of that side. This way the other side is communicating directly with an untrusted party, without any possibility to detect this session intruder.

Unaware of this unwanted user, the uncompromised party continues to exchange data and possible private data can be lost to this intruder. Another big disadvantage of the Multiparty Authentication algorithm is the required Session Authority. These systems will not function without the Session Authority handing out the session secrets.

The central Session Authority needs to be a trusted at any time. If the Session Authority is compromised, illegal session keys can be handed out. Skogsrud et al (2004) [10] proposed an algorithm for a lifecycle mechanism primary built Trust management. It however also includes a way of Authentication and this authentication is handled a different way as conventional user authentication.

As OWL-S is based on ontology’s [9] clients connecting to a service provider can be easily placed into a group corresponding section of the ontology. This method is easy and straight-forward, making it easy to implement. However, the ontologisms need to be detailed to accommodate all possible users or they need to be vague. The vagueness introduces a new security issue, because non-authorized clients can fit into the properties of a vague class and get authorized wrongfully.

R. Joseph Manoj et.al (2014) [14] developed a model of accessing web service based on user trust calculation. In this method they used a web server analysis based authentication system.

4. Proposed Work

In this paper, a new authentication method based on web server log is introduced. The system architecture consists of two major parts 1.Service requester 2.Service provider. The proposed authentication system is implemented at service provider location. In the authentication system there are different components to authenticate the requested users based on log file data. The overview of proposed system architecture is given in Fig.1.

4.1 System Architecture

In the following architecture, Data filtration, User authentication, IP address spoofing are the important processes which are involved in the service authentication process. The protocol of the architecture is given below:

4.2 Analyzing web server log

**Registered Users Data:** It is a data storage area of the authentication system which maintains the details such as user id, username, password, Remote IP address of last request, date and time of last request, previously requested page, number of hits (entry of visited time), Number of status codes ‘200’, ‘400’ and ‘408’ of various service requesters. Because
number of users may access the same service from same IP address or domain. These data will be used by later authentication processes.

**User Registration:** This process initially checks any requisition for new user registration. If process receives requisition for registration then new requester of service then has to register their details with their IP Address with service provider. During the registration process, requesters will be providing unique user id and password to access the web service. Consequently new requesters’ details such as user id, password, Requested URL, IP Address will be stored in the database. If the requester is already registered, it pass request to data filtration process with user id and request details.

**Data Filtration:** Data Filtration is the process in which data such as user id, password, Requested URL, IP Address of service requester who currently claims the service from service provider are filtered from the log file and pass the data to the user authentication process.

**IP Spoofing and origin Identification:** This process supports the user authentication to identify IP address spoofing by using Ingress packet filtering and route based filtering methods and prevent the attackers to access the system. Ingress packet filtering checks close to the source whether the address belongs within an assigned network range. Route based filtering check in the core, whether the route is correct for the given source address.

**User Authentication:**

This is the process where actual user authentication is takes place by analyzing the web server log details. Once it receives log details of requesters, it retrieves the requesters’ previous history from the registered user data based on user id. In the Fig.2, Authentication process is working as three cases. In case 1, if uid and pwd are valid and IP address is registered then checks spoofing and origin of address. In case 2, if uid or pwd is invalid and IP address is registered then it checks user is already visited the requested page, Number of bad request and number hits or visits with its threshold value. In case 3, if uid or pwd is invalid and IP address is not registered then system simply deny to access web service. The authentication algorithm is given below:

```
UserAuth (uid, pwd, ipaddr, reqpage)
{
  Case 1:
    Step1. If uid and pwd are valid and IP address is registered
    1(a) Checks IP address spoofing using Ingress packet filtering
```
If no address spoofing then Allow to access web service.

Else Deny to access web service.

Case 2:

Step 2: If uid or pwd is invalid and IP address is registered

2(a) Checks if requested page is visited & Number of hits > threshold value & Previous bad request < threshold value.

If yes then Allow three times to send request to claim service

Else Deny to access web service.

Case 3:

Step 3: If uid or pwd is invalid and IP address is not registered

3(a) Deny to access web service.

Fig.2: User Authentication Algorithm.

Web service gives response to requester directly after it receives service to be claimed and its value. Hence analyzing log files will make the authentication system more dynamic.

5. Experimentation and Resultant Data Analysis

In order to practice experiment data sets and to implement the procedures needed by the proposed authentication system from client registration to service request, an application-independent framework was developed. On top of the architecture layer, an e-library application was implemented to test proposed system operability and performance. At the lower level of architecture a web service for search and download the books was developed with authentication components such as user registration, data filtration, user authentication and IP spoofing and all these modules were located at the service providers area. IP Address spoofing identification uses Ingress packet filtering method to prevent the malicious user to access the system.

Ingress Packet filtering will be working as follows:

IF packet’s source address from within the prefix range

THEN Allow to access (Forward as appropriate)

Else IF packet’s source address is outside the prefix range
THEN Deny packet

END IF.

The e-library application was simulated in LAN environment with for more than 2 months and about 100 users were registered and involved in the experiment. During the implementation, intentionally the system was tested for IP address spoofing by form of a common security violation known as a man in the middle (MITM) attack. In the attack, a malicious party intercepts a legitimate communication between two friendly parties. The malicious host then controls the flow of communication and can eliminate or alter the information sent by one of the original participants without the knowledge of either the original sender or the recipient. In this way, an attacker can fool a victim into disclosing confidential information by “spoofing” the identity of the original sender, who is presumably trusted by the recipient.

The Information of requesters is analyzed and extract of web server log file is given in Fig 3. In the server log extraction, actual server IP address, client IP address, date of request, username, password, browser details, port number have been found.

After processing the server log data file such as shown above the following is sample output that obtain after it is processed by user authentication is given below in Table 2.

**Table 2: Sample Data from Registered User Database.**

<table>
<thead>
<tr>
<th>Used Id</th>
<th>Pwd</th>
<th>Client IPAddr</th>
<th>Prev. Req Date</th>
<th>NooStat Code</th>
<th>ReqPage</th>
<th>NooF Hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>A001</td>
<td>Ab01</td>
<td>197.168.1.66</td>
<td>30-09-2013</td>
<td>30</td>
<td>/supportvoice/_vti_book/_vti_aut/book.asmx</td>
<td>80</td>
</tr>
</tbody>
</table>
The Proposed system also was compared the difference between with and without ingress packet filtering technique using following experiments.

**Experiment 1:**

In the experiment 1, system is checked for how effectively it handles fraudulent victims without Ingress filtering technique for specific days. So the system was implemented without Ingress filtering technique and deliberately assigned specific number of attackers to spoof IP address. Since the system was based on role based authentication, it did not check the IP address is spoofed or not. So it allowed attackers to access the system.

**Experiment 2:**

In the experiment 2, system is checked for how effectively it handles fraudulent victims based on Ingress filtering technique for specific days. So the system was implemented based on Ingress filtering technique and deliberately assigned specific number of attackers to spoof IP address. Since the system was based on ingress filtering technique based authentication, it did check the IP address of requested users and allowed if there is no IP address spoofing. The above two experiments resultant data have been illustrated in Table 3 below:

**Table 3: Denial of Fraudulent Victims.**

<table>
<thead>
<tr>
<th>Day(s)</th>
<th>No. of Victims</th>
<th>Day(s)</th>
<th>No. of Victims</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>4</td>
<td>23</td>
</tr>
<tr>
<td>5</td>
<td>41</td>
<td>5</td>
<td>26</td>
</tr>
<tr>
<td>6</td>
<td>41</td>
<td>6</td>
<td>32</td>
</tr>
<tr>
<td>7</td>
<td>35</td>
<td>7</td>
<td>29</td>
</tr>
</tbody>
</table>
From the Table 3, it is concluded that ingress filtering technique effectively finds spoofing of IP address and authenticate the users correctly. The following Fig.4 shows the victim trace fraction of the system for a particular period. This Figure shows tracing of fraudulent users based on ingress filtering techniques as days are increases.

![Diagram showing Denial of Fraudulent Victims](image_url)

**Fig 4: Over all victims trace fraction.**

### 6. Performance Study

Finally the performance analysis of authentication system was conducted with existing traditional authentication system [14] such as checking username and password, third party authentication, attribute based authentication etc., In the basic user name and password authentication system, service access permission will be provided based on correctness of the user name and password. In the attribute based access control, users attribute are used to decide the permission. Performance analysis says that how efficiently all authentication systems are restricting fraudulent users from accessing web service.

The analysis was conducted in the proposed model with different parameters as shown in Table 3.

**Table 3: Parameter metrics comparison with existing schemes.**

<table>
<thead>
<tr>
<th>Factor</th>
<th>Username/Pwd Authentication</th>
<th>Attribute Based System</th>
<th>Third-Party Authentication</th>
<th>Proposed System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victim Trace Fraction (%)</td>
<td>74</td>
<td>80</td>
<td>91</td>
<td>94</td>
</tr>
<tr>
<td>Gain (%)</td>
<td>60</td>
<td>74</td>
<td>87</td>
<td>89</td>
</tr>
</tbody>
</table>
From the above table we can conclude that the proposed system protects about 94% of the fraudulent attacks from the attacker. The next parameter deals with the gain that authentication method offer to its users. The users with this scheme achieve more gain than the other existing system.

7. Conclusion

In the proposed system, the new authentication system based on server log files analysis was implemented. Unlike other web service authentication systems, it detects IP Address spoofing based on ingress packet filtering method. The proposed system results were obtained and performance analysis was conducted.

Performance analysis concluded that the system ability to restrict the fraudulent users better than other existing authentication systems such as username and password, third party authentication, attribute based authentication systems. As a result, the security of web services gets improved and proposed model is able to encourage requesters to take part in access control actively and honestly.

8. Future Enhancement

In future work, proposed system will explore to find the authentication method for new service requesters so that accessing web services will be more effective. Even though proposed model provides better authentication method to restrict fraudulent users, restricting attackers will be the big challenge.

Hence In future, system will be tested and rectified for various attacks like SQL Injection, Eavesdropping Further research of the proposed system will lead to complete and more practical solution to manage trust level in access control.

References


9. Barry Nijkamp; Authentication in Web Services, 5th Twente Student Conference on IT, Enschede, 2008


**Corresponding Author:**

**M. D. Anto Praveena**

**Email:** antopraveena@gmail.com