Authentication and Access Control in Cloud-Based Systems

Rajeshwari Gadathas Krishna Babu
Institute of Computer Sciences
Georg-August-Universität Göttingen
Email: r.gadathaskrishnab@stud.uni-goettingen.de

Aytaj Badirova
Institute of Computer Sciences
Georg-August-Universität Göttingen
Email: aytaj.badirova@gwdg.de

Faraz Fatemi Moghaddam
Senior Cyber Security Manager
GWDG, Germany
Email: ffatemi@gwdg.de

Philipp Wieder
Deputy head of GWDG
Institute of Computer Sciences
Georg-August-Universität Göttingen
Email: philipp.wieder@gwdg.de

Ramin Yahyapour
Managing Director of GWDG
Institute of Computer Sciences
Georg-August-Universität Göttingen
Email: ramin.yahyapour@gwdg.de

***Abstract*—Every organization places a high priority on maintaining the privacy and safety of sensitive data. Tokens are used to store sensitive information to prove one's authenticity prior to accessing system resources and services. JSON Web Token (JWT) is one such token that contains user information and is generated on the server side and issued to users for authentication purposes. However, there is still a gap in the existing research, which can be addressed with the proposed token revocation scheme in this work. The token revocation scheme ensures that the tokens belonging to deactivated users in the system are invalidated to address the issue of unauthorized users reusing tokens. RBAC is employed in this paradigm to enable access control, which strengthens security by granting users different levels of permission to access system resources. With this introduced framework, users are guaranteed to not use the previous role that was assigned to them.**

***Index Terms*—JWT Token, authentication, revocation, access control, security.**

I. Introduction

A JWT-based authentication is a scalable approach for user access control in large distributed networks. However, there are 2 main issues while using JWT based token authentication scheme. Firstly, there might be a huge loss if a user's role is changed after a token was allocated to them. It could abuse the user's access privileges or permissions. Secondly, JWTs may introduce authentication vulnerabilities. Before a JWT token expires or until the user logs out of the current session, adversaries may try to use the token to gain unauthorized access to resources. This research aims to address the security holes in JWT token-based authentication. A token revocation approach with *Role Based Access Control* (RBAC) has been introduced in this study that would provide a solution to the token expiry issue and user role updation in the database. The primary goal of this study is to propose a solution to the present JWT token expiration problem and prevent unauthorized usage of tokens. The token revocation mechanism ensures that any tokens belonging to inactive users in the system are invalidated and can no longer be utilized by unauthorized individuals.

A. Related Studies

This section presents the studies related to JWT token authentication. Study [1] focused on the integration of timestamps of client and server response time. Although this method has good security and ensures data integrity, the performance is questionable in terms of response time. To enhance the security of the token authentication, the authors in [2] established a scheme that incorporates security features such as session management and cryptographic techniques. In [3], a token revocation solution was proposed using real-world metrics as the expiration validity of a JWT is controlled by cryptographic procedures. They evaluated the performance and efficiency of the model. As a result, in all the related studies, although there is a clear focus on enhancing security, there is still a lack of JWT Token revocation techniques that are necessary to avoid token reuse. In this paper, we will fill the gap by proposing a scheme for JWT token revocation and suggest additional security measures that could be used to avoid current security vulnerabilities associated with JWT tokens.

II. Proposed Model

The recommended design and algorithm have been discussed to address the present-day issues with JWT in this section. The suggested method seeks to address the research problems that have been examined, and the algorithm aims to emphasize the principles and features of the model.

A. Token Revocation Technique

The JWT token revocation method together with RBAC can specify which users can obtain which privileges and under what conditions. However, the security of the current authentication systems using JWT cannot be fully achieved

without a suitable revocation technique. Therefore, we suggest an appropriate token revocation system that helps in preventing unauthorized access to system resources and also controls the privilege levels of the users and organizational policies using RBAC. We proposed two new data structures on the server end called *Token Revocation List* (TRL) and *Token Status Checker* (TSC) to validate the incoming JWT and would safeguard against illegal access. The expiration of the token and the updating of the user role are the current problems with JWT. The algorithm for token revocation and user role updation is presented below.

Algorithm for Token Revocation

For Users i=1 to n1 ;
if((Tid(t)==Tid(rc)) **then**
Token has already been Revoked
else
if(((Tid(t)==Tid(s)) **then**
Do the following for n records in JSC
else
if(((Tiat(t)!=Tiat(s))[or]**if**(Texp(t)!=Texp(S))[or]
if(Tnbf(t)!=Tnbf(s))[or]if((Taud(t)!=Taud(S))[or]
Trole(t)!=Trole(S))) **then**
Move the corresponding token to TRL
else
Grant access to resources
if((Trole(t)==L1) **then**
Assign user with Basic user rights
else if ((Trole(t)==L2) **then**
Assign the user with Administrator Rights
else if(Trole(t)==L3) **then**
Assign the user with Super Administrator Rights
else
Request Denied

When a JWT token 't' arrives at the server, it checks to see if the token IDs 'tid(t)' present match those in the list of tokens that have been revoked. If the token has already been revoked and is present in the TRL, it is presumed that the token has already been invalidated by the server or has expired, and the access request will be immediately refused. If the token ID is not listed in the revocation list, the system then confirms that the token is present in the JWT status checker until its expiry time is met, which keeps complete records of all issued JWTs. Then it validates the claims present in the with those in the TSC. JWT Tiat(t), Tnbf(t), Texp(t), Trole(t) are the token issuing time, not before, expiry time, user roles. If there is any mismatch in the value of the claim, then it invalidates the token and sends it to the TRL and therefore discards the request. A JWT with the token ID 'TKID3' was issued in this depicted scenario, upon request by the user during the initial phase. This token ID is subsequently added to the token header and transmitted to the server with the authentication requests. The token ID is verified by the server using these two tables when it receives

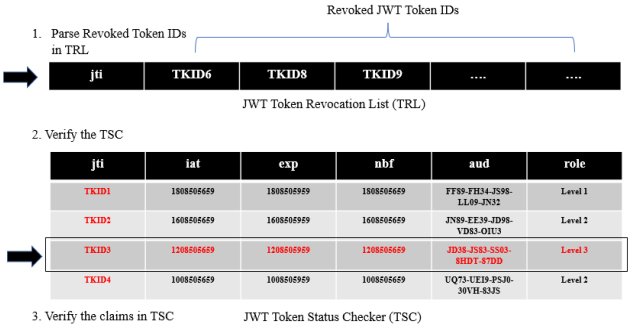

Fig. 1. Token Revocation List and Token Status Checker

this request. The token ID is initially checked against the token revocation list using the "jti" claim. It is believed that the token has already been revoked and cannot be reused if it is already present. In the absence of a token ID, the server then parses the JWT token status checker table to validate the token claims and guarantee that the claims are legitimate. It verifies that the token has not expired or been invalidated previously, the period of issuance, the intended audience for the JWT token, and their access permissions by verifying each claim value. The server provides access depending on user roles and RBAC policies if the claims are true and legitimate.

III. Evaluation and Discussion

The suggested model is assessed using two evaluations: Security Evaluation and Comparative Evaluation in this section. We will assess a few user-experience situations and a comparative analysis of this design concerning the other related studies to determine their benefits and drawbacks.

A. Security Evaluation

Theorem 1: *The token that was provided to a user can no longer be utilized by any other authorized users in the system if the user is no longer active in the system. The token is rendered invalid and ineffective once a user account is deleted from the system.*

Proof: Assume Bob's account with 'Role' as 'Super Administrator' has been terminated from the system but the JWT token he was granted would stay operational until its expiration. This token might be accessed by another user Alice to access and utilize Bob's intended super administrator resources. In this situation, the token revocation mechanism is applied to invalidate the tokens that belong to a deactivated user and send them to a token revocation list. This ensures that if any of the token claims such as a token expiry time or date is mismatched, the corresponding token would be moved to a token revocation list to prevent unauthorized access to system resources. Alice presents this token to the server to obtain access to resources. The server looks up the database and identifies that Bob's account is no longer available in the active directory and invalidates the request from Alice by moving the token to the revocation list.

Theorem 2: *An authenticated user cannot utilize a previously assigned user role issued in a token to access resources in the system.*

Proof: User Bob was previously assigned with the 'Super Administrator' role and has now been modified to a 'Basic User'. This circumstance will be handled with the aid of the revocation technique presented in this work. Assuming Bob is a malicious user trying to utilize 'Basic User' privileges to access the critical resources in the system which could only be accessed by a super administrator. The token with the "Super Administrator" role will initially be checked to see if it is present in the TRL. If not present in TRL, the TSC is verified and the records are inspected based on the Token ID. The issued claims are then verified, and the user's current role is established. The user is given access if the role is the same. The request will be rejected if there is a role mismatch since it is assumed that the token is being used for an unapproved function.

Theorem 3: *The JWT token may withstand security attacks during transmission without modifying the payload or token content.*

Proof: The suggested approach in this study effectively secures a JWT token in transit employing a *Transport Layer Security* (TLS) *Version* V 1.3 connection which is encrypted, thus making it impossible for any individual to intercept. It is highly recommended to use the 'HttpOnly' cookie so that the header content is unseen and prevents stop Cross-Site Scripting attacks. For instance, a man-in-the-middle attacker tries to intrude and capture the token and tries to modify the JWT claims present in the payload and gain access to the system token creation. Here, the connection is secured using *Hypertext Transfer Protocol Secure* (HTTPS) protocol and therefore is encrypted. Therefore, the information inside the channel would be encrypted packets making it impossible to decrypt the packet and obtain access to the system resources.

B. Comparative Evaluation

A comparative analysis of the proposed design is performed. The comparison factors include user role updation, access control mechanism, data integrity check, and resistance against security attacks. The suggested methodology also has the advantage of invalidating obsolete or expired tokens.

Token Revocation Method: The Token Revocation method is performed to make it possible for clients to sign off which enables the server to erase any security credentials connected to the permission [4]. The token revocation method illustrated in this study helps in revoking obsolete tokens.

User Role Updation: Users are assigned security roles such as Basic User, Administrator, and Super Administrator. The current existing role in the claim differs from the role in the status checker record when a user's role is changed to a new role in the active directory. Therefore, by using the suggested strategy in practice, the token claims presently assigned to the user will be updated with the most recent role.

Access Control Mechanism: In this research analysis, RBAC is used as a policy standard to provide users access to key resources based on their security levels [5] The role-based security paradigm eliminates the need to manually alter access control lists throughout the active directory and grant users rights based on the job positions they have inside the organization.

Data Integrity: Tokens are digitally signed using cryptographic algorithms making it harder for attackers to compute the signatures, especially when implemented with asymmetric cryptographic algorithms. Even if the token's payload content has been altered, the server will still recognize it while computing the signature, and the signatures will not match and discards the token. Therefore, by suggesting a token status checker that verifies each claim on the incoming JWT with the registered records held in the server's database before granting access to a resource, this model passes the data integrity check.

Resistivity against Man-in-the-Middle: This model can prevent man-in-the-middle attacks(by using HTTPS TLS V 1.3 Protocol) and phishing attacks (by using digital signatures). Therefore, we can conclude that the proposed model in this paper takes into consideration all the relevant factors, including a token revocation procedure, user role change effects, attack resistance, the use of access control, and data integrity assurance.

IV. Conclusion

Current JWT research has some challenges that need to be addressed. Three major issues with JWT-based authentication are JWT token expiration, user role updation, and security. The proposed framework addresses these issues by implementing a token revocation scheme. Through the use of the TLS V 1.3 protocol, which encrypts the communication channel and defends the network against attacks, this technique also guarantees confidentiality. Access control further enhances the security of the model while maintaining flexibility. As a result, the security element is separated into three distinct levels (Basic User, Administrator, and Super-Administrator). This design is governed by the RBAC policy. The token is signed using cryptographic means which guarantees data integrity. Future work may involve implementing the illustrated methodology and architecture in real-time cloud-based systems and integrating multi-factor authentication to improve security. This will result in a robust and secure method to access control and authentication for cloud-based systems.

References

[1] Ahmed, Salman and Qamar Mahmood. "An authentication-based scheme for applications using JSON web token." 22nd international multi-topic conference (INMIC). IEEE (2019).

[2] Yang, Shulin, et al. "Implementation of Permission Management Framework Based on Token through Shiro."International Conference on Computer Technology, Electronics and Communication (ICCTEC). IEEE (2017).

[3] Jánoky, László Viktor, Péter Ekler, and János Levendovszky. "Evaluating the Performance of Novel JWT Revocation Strategy." Acta Cybernetica (2021).

[4] RFC 7009: Token Revocation. OAuth Documentation (2020).

[5] Xu, Jian, et al. "Role-based access control model for cloud storage using identity-based cryptosystem." Mobile Networks and Applications (2021).