Grenze International Journal of Engineering and Technology, Jan Issue

Session Tracking with User Authentication

Yeswanth Reddy Kondamadugula[1], Amara Sai Krishna Kumar [2], Prudwi Adapa[3],
Hemanth Reddy Bheemireddy [4] and Chandra Sekhar Kolli[5]
[1-5]Department of Computer Science and Engineering,
Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur District, Andhra Pradesh

***Abstract*—Many of us using may web applications in our daily life but many web applications are insecure because they use session-based authentication by overcoming this drawback we are using Token- Based Authentication, which is more secure than Session-based Authentication solution is, in the token-based application the server makes JWT with a unique key and sends the JWT to the customer. The customer stores the JWT (as a rule in nearby capacity) and incorporates JWT in the header with every request. Then the server would approve the JWT with each request made by the client and sends reaction. To keep the client from login on totally different frameworks or net browsers you want to produce a token on every login has to be completed to check the token on each page. The possible chance that the token does not coordinate, at that point in the SESSION and the client will logout automatically and message will display on the screen. The possibility that the client tries to login for the second time, he needs to logout the previously opened session which has been provided at the first login time and also, we check the client with the validate Token. For example, google authenticator or by means of confirmed email address in this manner Authentication made sure about.**

***Index Terms*— Token, authentication, security, privacy, sessions.**

I. INTRODUCTION

If a client has an active session being used and a second login happens, the first login will be ended right away. We do this as opposed to prevent the second login to prevent a dormant program left open, or client who finished their program without logging off from inadvertently keeping themselves out of their own account. The authentication system is the one that permits a user to access a resource only after supplied credentials checked with that stored in the database and found to be the same.

A. Why we need to prevent simultaneous logins (User Stories)?

The principle use case for forestalling synchronous logins is to forestall account sharing – numerous individuals utilizing a paid account. By empowering this setting, you are making account sharing practically trivial as though someone else signs in then the current client's session will be shut down (they will be logged out).

B. How authentication is made?

During the years, many verification strategies were presented and applied, every one of them attempting to offer safer assistance than the other. First secret word-based validation was offered, later meeting based, biometric verification and later token-based confirmation which was most dependable considering its structure. Additionally, in it is depicted validation utilizing testaments. In this paper will depict meeting and token-based validation in complete, their burdens and advantages.

Grenze ID: 01.GIJET.7.1.501_4

II. Literature Survey

From the publication Session Management Vulnerabilities in Today's Web [2] the author says Now a day's cyber-crimes are increasing day by day. To reduce that they are lot mechanism are introduced in web application which improves the security of data and user. Session management is one technique Session management tracks a user's activity across sessions of interaction with a website. They are also one existing mechanism called cookies. Which is not an effective method when compared to sessions. Session management is utilized to encourage secure interactions between a client and some service or application and applies to a sequence of requests and responses related with that particular user. By using the Http protocol which is stateless protocol which take the request from client to server in a secure fashion.

From the publication Token-Based vs Session-Based Authentication A survey [1] In the current world condition, there is a fast expansion in the internet services. In a considerable lot of these administrations which has delicate information of the clients must be made sure about by verifying the clients. So, there is a necessity of the authentication, to verify the actual user. Generally maximum systems will be using cookies and session-based authentications to validate user.

Here in this research paper the creator discussed and analysed briefly about token-based authentication and explained it works. The author also compared old techniques like session-based authentication with his proposed technique (token-based authentication) and explained why to choose token-based technique over existing session-based techniques.

III. Token-Based Authentication

Token-based validation is one of many web confirmation techniques used to make a safer check measure. Token verification expects clients to acquire a server produced code (or token) before they are allowed network section.

A. Where is token-based authentication is utilized?

Token-based confirmation is only one of many web validation techniques used to make a safer check measure. Other web validation strategies incorporate biometric verification and secret key confirmation. While every verification strategy is unique, all strategies fall under one of the accompanying three classes: information (something you know), legacy (something you are), and ownership (something you own).

Secret key confirmation falls inside the information classification since clients depend on a word or expression, they have recently made to check their personality. Biometric verification is an illustration of "something you are" because of its utilization of organic qualities, like fingerprints. What is more, last, however unquestionably not least, token-based validation has a place in the ownership class. Token verification expects clients to acquire a server produced code (or token) before they are conceded network passage. Token confirmation is regularly utilized related to secret word validation for an additional layer of security.

This is what we allude to as two-factor confirmation (2FA). That means even if an attacker will be implementing a brute force attack to take out any secret key, they should also have to bypass the token authentication layer. Without admittance to the token, accessing the organization turns out to be progressively troublesome. This extra layer debilitates assailants and can spare organizations from possibly deplorable penetrates.

B. Process

Step 1: Initially while a user account has login in one system then in the database one session is created for that user in data base.

Step 2: User need to do some activity in that account or else it will be logout after some point of time.

Step 3: If one session is active but user forgot to logout where he has some work in that account, but he cannot access that account directly because that account active session is running in server.

Step 4: So, to destroy the active session, we need to validate weather, the user is correct user or not, so that we are introduced an authentication mechanism by using google authenticator.

Step 5: Where we need to enter a mail id or phone number which is already registered in the database with that specific account.

Step 6: An otp will be sent to the registered account. After entering the otp the active session will logout or destroy.

Step 7: Then new session will be created in database. Where this mechanism improves security and authenticity of the user's account by using the sessions and google authenticator or mail authentication technique.

C. Existing process

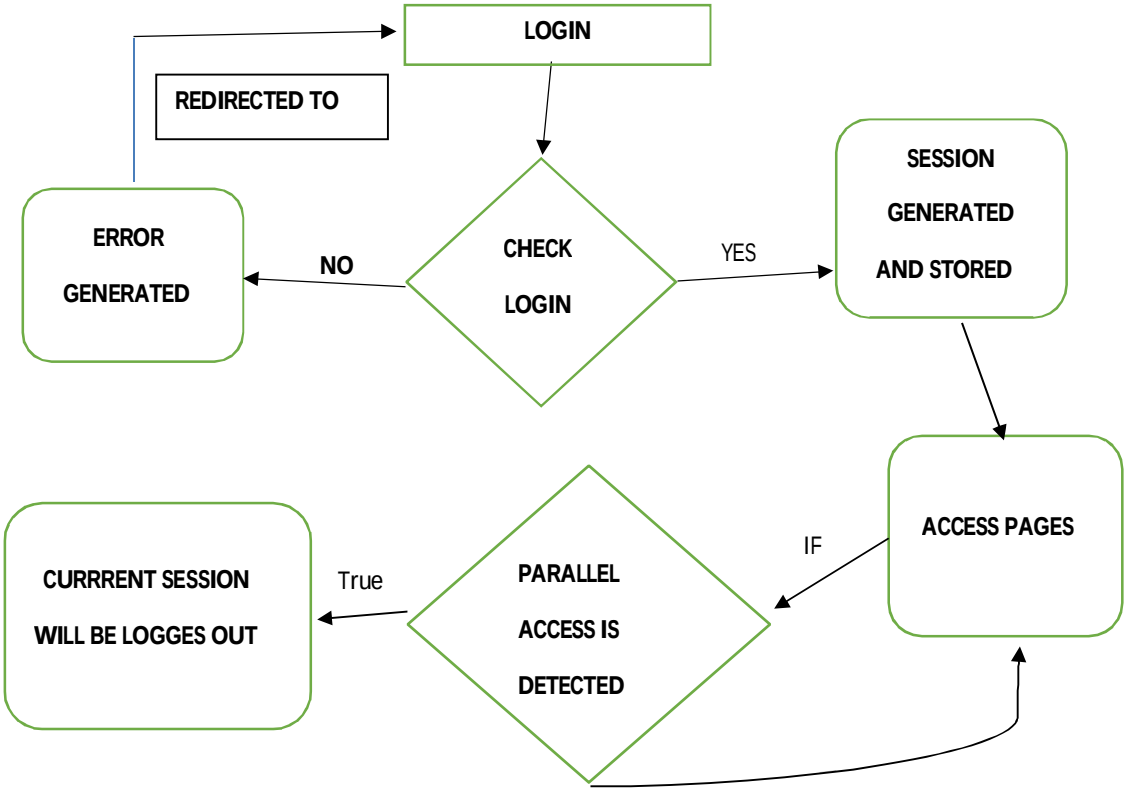

Fig 1. Existing process

This is the pictorial representation of existing process. This starts from login where it has a decision making of whether to login successful or failure. If it is a failure it will be redirected to login page else, it will create a session for current user and stored that session after that it will redirect after successful to access pages and that after accessing it will be detected weather no session is not yet active then it will continue else it will logout the active session. If no active session it will stay in access page.

IV. RESULTS

Now and again expected to restrict the customer to simply sign-in to a single session at a time. To allow the client from login and accessing on various systems or web programs you should create a token on each productive login attempt.
We need to validate the token on every page. If the token does not coordinate, at that point obliterate the SESSION and log out the client. Utilizing the above PHP content, we can forestall the various logins of the client and naturally log out from different spots when it gets signed in.

V. CONCLUSION

We have proposed and build session tracking with user authentication in my website, which can be used to authenticate users and prevent double logging prevention and by using authenticators when one session is active a second session with same credentials want to login prior to the first session is not over or closed it will ask the user to enter the otp to check the client is authorized or not by this project we enhanced both authentication and the security of the users or clients and makes their data more secure and confidential.

REFERENCES

[1] Session Management Vulnerabilities in Today's Web, Corrado AaronVisaggio https://www.researchgate.net/publication/220496661_Session_Management_Vulnerabilities_in_ Today's_Web

[2] Token-Based vs Session-Based Authentication A survey, Yjvesa Balaj https://www.researchgate.net/publication/320068250_Token-Based_vs_Session- Based_Authentication_A_survey

[3] Kubovy J., Huber C., Jäger M., Küng J. (2016) "A Secure Token-Based Communication for Authentication and Authorization Servers". In: Dang T., Wagner R., Küng J., Thoai N.

[4] Hardt, D.: "The OAuth 2.0 Authorization Framework." RFC 6749, RFC Editor, October 2012.

[5] Richard E. Smith "Authentication: From passwords to public keys", only first chapter called "The authentication landscape", originally published on October 01, 2001.

[6] Rene Enriquez, Andres Salazar C "RESTful Java Web Services Security" originally published on July 28, 2014.

[7] Sungchul Lee, Ju-Yeon Yo,Yoohwan Kim "Authentication System for stateless RESTful WebService", originally published in November 2016 in Computer Science, vol 10018. Springer, Cham.

[8] Xiang-Wen Huang, Chin-Yun Hsieh, Cheng Hao Wu and Yu Chin Cheng, "A Token-Based User Authentication Mechanism for Data Exchange in RESTful API" , vol. 00, no. , pp. 601-606, 2015, doi:10.1109/NBiS.2015.89.

[9] Brachmann E., Dittmann G., Schubert KD. (2012) "Simplified Authentication and Authorization for RESTful Services in Trusted Environments." In: De Paoli F., Pimentel E., Zavattaro G. (eds) Service-Oriented and Cloud Computing. ESOCC 2012. Lecture Notes in Computer Science, vol 7592. Springer, Berlin, Heidelberg.

[10] Kurt Gutzmann. "Access control and session management in the HTTP environment." IEEE Internet Computing, 5:26–35, 2001.

[11] Jones, M.B., Hardt, D.:"The OAuth 2.0 Authorization Framework: Bearer Token Usage." RFC 6750, RFC Editor, October 2012.

[12] Sandro Wefel and Paul Molit "Raising User Acceptance Of Token-Based Authentication by single sign on" originally published in 2012.