Token Based Messenger

Shivam S. Patel[1], Heril N. Patel[2], Keval N. Desai[3], Harshil D. Rathod[4], Narendra V. Jagtap[5]

Assistant Professor5, Student1,2,3,4
Department of Computer Science & Engineering1,2,3,4,5
R.N.G. Patel Institute of Technology1,2,3,4,5
Surat, Gujarat, India

---***---

Abstract - Information security is a major challenge. Old session-based authentication has issues related to network tampering and cookie stealing over the network which can give attackers access to read/modify data and control smart devices in the network. So, we propose a method to authenticate users using a Token based authentication method with the purpose to secure data and validate it. Tokens are being generated by both server and client to validate each other. Tokens are unique and random make it tough to guess and regenerate. It validates the client's integrity, login credentials and server. Our Solution is a JWT alternative in which we are going to develop token based messenger for Web application and android platform. Basically, messages are only encrypted in Package so it is possible to Hack/Capture that packages with different Tools (Specially WireShark) and Techniques. We are going to merge message and authorization token in only one token so each time when a user sends a message to anyone, every time authenticity of the user and user's message will be checked. Because of this token the risk of the data unauthentic will be reduced and it will help to increase the security of the Internet.

Key Words: Token, Authentication, JWT(JSON Web Token), Security

1. INTRODUCTION

In recent years IIoT (Industrial Internet of Things) grown up very fast. These smart factories are modular by design and it is a cyber-physical system which can monitor all physical process going on. As its digital era, there are challenges in implementation like Data Security, Reliability and Stability important for machine to machine (M2M) communication, maintain integrity of production processes, etc.

In any digital system security is a major concern. There are lot of chances that attacker or hackers try to breach sensitive files and industrial nodes. Industry rivals can try to get sensitive data. There are many techniques to attackers uses to breach into system. Attackers can attach into network of industry and can-do man-in-the-middle attack to capture data transferring into network. Attackers can physically inject malware, spyware or viruses into company's system or network to access data, credentials or destroy whole system.

2. Body of Paper

2.1 LITERATURE SURVEY

In 2017 Shih-Hsiung Lee [1] propose article on TBAS framework that the device can have secure access to the cloud services through the token. Token is generating by the third-party authentication center so that the sensitive data are not easily leak. And the article proposes an authentication service for IoT scenario.

In 2017 Yjvesa Balaj [2] propose a survey on comparison of Token-Based and Session-Based Authentication. These both techniques have its own advantages and disadvantages.

This method was old technique which is almost every site used it. By using cookies, we can exchange information between the remote server and user. one disadvantage of using session-Based authentication method. Cookies have very low limit in the data. Cookies can be set or read both side client and user.

Where other side we have token-based technique. Token-based is a newly in IT field, the usage of this technique has increased dramatically. The token-based technique is completely Stateless, it doesn't store any information about the user. Token is signed not encrypted; it carries user's information who wants to authenticate. Cookies and Tokens have same usage, but tokens don't need to be stored in the server in order to work, with tokens the server only needs to verify that the token is valid before authorizing a request. [2]

In 2006 B. Lin, Y. Chen, X. Chen and Y. Yu [3] propose comparison between XML and JSON (Java Script Object Notation) for Data loads, JSON provides high level of efficiency and flexibility for light-weight data interchanging. Results show that JSON is more suitable ass a data-loading tool.

In 2009 Aloul, Fadi, Syed Zahidi, and Wassim El-Hajj [4] propose a method for two factor authentication using mobile phones, where a mobile phone is used as software token for password generation which is valid only for a short user-defined period and is unique to both user and mobile every time. And SMS-based mechanism is used for retrieving password and as a mean of synchronization. Initial results showed the success of proposed method.

In 2020, Apple recently paid $100,000 to researcher for finding vulnerability in their JWT mechanism in which attacker can change the email in token and create new account with victim email address in third party applications.[20].

2.2 PROBLEM STATEMENT

When session is opened attacker may intercept network traffic. Sessions are maintained by the persistent network and state information shared between client and server.

Session based authentication is still used by lot of website which do not hold important data for the users instead others use token-based authentication for a better security. While session-based technique is stateful, it stores every client data on server and this cause overload the server. And its management across other server is quite a big challenge. [19]

2.3 ABOUT EXISTING SYSTEM

➢ Passwords:

The simplest and most convenient way of authenticating a user is Using a Login ID and Password(s). Many Applications uses this method to Authenticate the user. In traditional password authentication, each user has an ID and Password. User has to submit their ID and Passwords as their Login Credentials which are stored and maintained by remote server in a table. If the Login Credentials submitted by user matches with the corresponding pair on server, they are authenticated. However, if an intruder breaks into server, they can impersonate a user by Stealing their credentials from table which is called stolen-verifier attack. Attacker intercept network traffic and easily capture session data or steal cookies between user and remote server by performing man-in-middle attack. To prevent the password from stealing by attackers, passwords are usually hashed or encrypted inside the computer [5, 6]. Cookies and session data are still vulnerable.

➢ **2FA (Two Factor Authentication):**

This is extension to password method to increase security. It is extra layer of security provided by company or trusted third-party vendors. Along with normal authentication after validating password one temporary code is sent to the user. User can enter that OTP (One Time Password) during login and authenticate himself. This method is secure as only authenticate user can get OTP through predefined sources like Emails, SMS on validated Mobile Number. OTP generate by servers and are random in nature so it is hard to guess it and regenerate it. However, it is still one-time process so after login also same cookies and session id are generated and sent to client. So, Attackers can capture it or steal from physical computer and get access to account. Some Vulnerable nodes can also give access to attackers. [4]

➢ **Tokens:**

Token are replacement for session-cookie methods. Instead of session details, server generates one authenticated token which sent to the client. Client then use that token with original request packet. Token are generally not store in cookies so not much issues. Some attacks like CSRF and XSS generally not work for this method. Token validation are depending on server side so token can be invalidating easily. Token are disposable and not store inside server so no storage problem on server because of tokens. [17]

JSON Web Token (JWT) is an open standard that defines a compact way for securely transmitting information between parties as a JSON. It is better than XML in performance [3] The information can be verified and trusted because it is digitally signed using a secret (with the HMAC algorithm) or a public/private key pair using RSA [11, 14]. Token-based authentication uses JWT to store the data and send it to server in authorization header starting with Bearer keyword. The JWT has three segments: header, payload and signature. [18]

Referring to RFC there are different types of token [9, 10]:

- perishable token: is used to validate a single action
- session token: is valid for one specific session and can be used several times within this session.
- access token: can be used multiple times but cannot be renewed
- refresh token: can be used only once (must be invalidated after its use).

2.4 PROPOSED SOLUTION

Here we propose our solution with Token Based Authentication Algorithm. We will use RSA (Rivest-Shamir-Adleman) Public Key Cryptography algorithm to generate keypair of public key and private key [11]. This algorithm is relatively slower to compute on smaller microprocessors so It is optional to use in system depending on need of extra security. Tokens are in format contains id, data, hash and validity encoded in compact JSON [8] Serialization format using Base64-URL [7]. We are using JSON here in place of XML for data because its compact [3].

Our Propose algorithm is a Token-Based authentication algorithm contains following 4 major parts:

I. Token validation.
II. Send/Receive Data in Token.
III. Keys Exchange with RSA Security.
IV. Send/Receive Data with RSA Security.

➢ Format:

Packet Data:

```
{
    “id” : “#id number”,
    “Data” : “#Token data”,
    “Hash” : {
                “algo” : “#hashing algorithm”,
                “hash” : “#hash of token data”

             }
    “Validity” : [Time in milliseconds]
}
```

#id number
ID is predefined number to show which type of token is that

#Token Data:

```
{
    “userid” : “#userid of user”
    “password” : “#password“
    “details” : “#other details“
    …
}
```

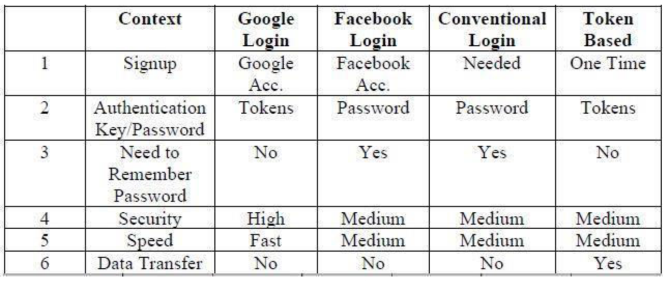
Token Data = f(Base64Encoded) (f (Encryption) (#Plain Token data))

#type of hash algorithm
Algorithm field specifies type of hash algorithm that will be use like MD5 [12], SHA1 [13] etc.

#Hash of Token Data

Hash is HMAC [14] of defined algorithm type of #Token Data and secret is use to sign it.

At First, Login Token is generated by client and send to the server. Server validates it and generate response for it then send it to client. Client receive response ack token and validate it then store it for future usage. If server get error to validate login token then it sends error token for client. Server won't store any token data it just validates it. Whenever client request from server it sends token stored into Authorization header with packet data. So that server can receive and validate it every time.



	Context	Google Login	Facebook Login	Conventional Login	Token Based
1	Signup	Google Acc.	Facebook Acc.	Needed	One Time
2	Authentication Key/Password	Tokens	Password	Password	Tokens
3	Need to Remember Password	No	Yes	Yes	No
4	Security	High	Medium	Medium	Medium
5	Speed	Fast	Medium	Medium	Medium
6	Data Transfer	No	No	No	Yes

Table 1: Comparison of various platforms

3. COMPARISON ANALYSIS

We compared the features and functionalities of various existing platforms (Google, Facebook, and Conventional login). In the existing systems like Google and Facebook, we need to login through their own login mechanism while on a token based system, users need to login for once and then their saved token will be used afterwards for login. Also, Passwords are stored in a conventional login system while in a token based system, the token will be stored and the user has not to remember the password. Moreover, Data Transfer rate is also good in compare to other applications.

4. CONCLUSIONS

A Token based authentication method proposed in this paper. The paper leverages on the stateless and compact feature of Tokens for authentication and access for IoT devices and Computer applications. We give introduction comparison of Session-based Authentication and Token-based Authentication. There was discussion related to problems with sessions and tokens. A proposed solution is there to show how can we transfer data with tokens securely and an optional and more secure RSA Cryptography [6] based Implementation also shown.

ACKNOWLEDGEMENT

The acknowledgement is just a drop of sense of gratitude within our hearts for the people who helped us out of the most embarrassing part of life when we are standing on the last & most difficult step towards our life.

The entire session of our phase I completion was a great experience providing us with the insight & invocation into learning various software engineering concepts & benefits of team work. Likewise, every member has burnt fuel, day & night for completing project. We would like to take this opportunity to express our sincere thanks to all those people without whose support and co-operation, it would have been difficult to complete this project.

Primarily, we are very much thankful to our project guide MR. NARENDRA V. JAGTAP for his leading guidance and sincere efforts throughout project work. He took deep interest in simplifying the difficulties. Also, he has been consistent source of inspiration for us.

We also express thanks to Prof. H. J. Shah project coordinator for his personal involvement, constructive suggestion and thoughtful idea for betterment the project.

We are grateful to our H.O.D. Dr. Madhavi B. Desai and our beloved Principal Dr. Latesh B. Chaudhari for providing us deep knowledge and all necessary resources.

We are also thankful to our Friends and Non-teaching Staff for their valuable time & help for completion project.

Once again, we are grateful to all those without whom this work would not have been successful.

REFERENCES

[1] International Journal of Distributed Sensor Networks, vol. 13, 7, First Published July 14, 2017.

[2] Yjvesa Balaj "Token-Based vs Session-Based Authentication: A Survey" originally published in 2017

[3] B. Lin, Y. Chen, X. Chen and Y. Yu, "Comparison between JSON and XML in Applications Based on AJAX," 2012

[4] Aloul, Fadi, Syed Zahidi, and Wassim El-Hajj. "Two factor authentication using mobile phones." 2009 IEEE/ACS International Conference on Computer Systems and Applications. IEEE, 2009.

[5] A. Evans Jr., W. Kantrowitz, E. Weiss, A user authentication scheme not requiring secrecy in the computer,Commun. ACM 17 (1974) 437– 442.

[6] G.B. Purdy, A high security log-in procedure, Commun. ACM 17 (1974) 442–445.

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

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

[9] 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.

[10] 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

[11] R. L. Rivest, A. Shamir, and L. Adleman. 1978. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21, 2 (February 1978), 120-126.

[12] Rivest, Ronald. The MD5 message-digest algorithm. No. RFC 1321. 1992.

[13] Eastlake 3rd, D., and Paul Jones. US secure hash algorithm 1 (SHA1). No. RFC 3174. 2001.

[14] Krawczyk, Hugo, Mihir Bellare, and Ran Canetti. HMAC: Keyed-hashing for message authentication. No. RFC 2104. 1997.

[15] Josefsson, Simon. The base16, base32, and base64 data encodings. No. RFC 4648. 2006.

[16] Bray, Tim. The JavaScript object notation (json) data interchange format. No. RFC 8259. 2017.

[17] https://scotch.io/tutorials/the-ins-and-outs-of-token-basedauthentication

[18] https://auth0.com/learn/token-based-authentication-made-easy/

[19] https://swoopnow.com/token-based-authentication/

[20] https://www.welivesecurity.com/2020/06/01/bug-sign-in-apple-account-hijacking/