Journal of Applied Engineering and Technological Science
Vol 3(2) 2022 : 149-155

THE MODEL FOR STORING TOKENS IN LOCAL STORAGE (COOKIES) USING JSON WEB TOKEN (JWT) WITH HMAC (HASH-BASED MESSAGE AUTHENTICATION CODE) IN E-LEARNING SYSTEMS

Syabdan Dalimunthe[1*], Joeharsyah Reza[2], Asep Marzuki[3]
Department Of Computer Engineering [1,2,3], Politeknik Caltex Riau,Indonesia
syabdan20s2tk@mahasiswa.pcr.acid[1],joharsyah20s2tk@mahasiswa.pcr.ac.id[2],
asep20s2tk@mahasiswa.pcr.ac.id[3]

Received : 20 May 2022, Revised: 29 June 2022, Accepted : 30 June 2022
Corresponding Author

ABSTRACT
The problem with the e-learning system is about data integration. Where the system is slow to access because it has to check into the database. Web Service is a set of standards and programming methods for sharing data between different software applications, distributing services over the internet that supports system interoperability. Data is exchanged in JSON format, and JSON Web Tokens are used for authentication security (JWT). Using JWTs for token-based authentication of web services can help overcome interoperability issues. JWT is stateless and allows inclusion of data in token authorization. JWT includes a number of possible algorithms, including HMAC. Overall, HMAC adoption outperformed the criteria for token generation time, token size, and token transfer speed. Saving JSON Web Token to local storage in client browser using HMAC algorithm has been presented in this paper. The proposed work has shown that JWTs do not need to be stored on the server but can be stored on the client browser side using local storage.

Keywords: *Hash-Based Message Authentication Code, Json Web Token, Restful API, E-Learning*

1. Introduction

Learning Media in Indonesia is increasingly experiencing significant development. The dynamics of today's technology is achieving tremendous acceleration. The form of information technology development that can be used as a learning medium is e-learning. E-learning is an innovation that can be utilized in the learning process, not only in the delivery of learning materials but also changes in the abilities of various competencies of learners. Based on the Joint Decree of the Minister of Education and Culture, Minister of Religion, Minister of Health, and Minister of Home Affairs regarding guidelines for implementing learning during the coronavirus disease 2019 (covid-19) pandemic which encourages the implementation of limited face-to-face learning and distance learning. E-learning is an integrated system with several other systems to support online learning data needs, such as: student data, lecturer data, course data, class data, and several other data spread across various systems.

The data is integrated using a Web Service. Web Service is a set of standards and programming methods for sharing data between different software applications, distributing services over the internet that supports system interoperability (Alam et al., 2020; Rahmatulloh et al., 2019). Web service is the technology choice most often used to conduct information exchange and sharing (Adam et al., 2020; Ethelbert et al., 2017; Tihomirovs & Grabis, 2017).

Session-based authentication constantly causes the server to overload because it has to check the session all of the time. This affects server performance. Because no sessions are formed every time a user logs in, unlike token-based authentication, this strategy outperforms session-based techniques (Adam et al., 2020; “Insights of JSON Web Token,” 2020). Web services are developed on open protocol technologies such as TCP/IP and HTTP. Web service is a method for communicating between two or more electronic devices over a computer network. There are two protocols commonly used in web service development, namely SOAP and REST (Kumari et al., 2019). Many web service implementations have been carried out in several sectors. In previous research, used a web service with a REST API that equipped with an access token on the data management system Public. The research discusses about creation of a web-based system designed for manage community data with features for monitoring access to databases on the server, so that administrators can easily monitoring data traffic conditions (Perkasa & Setiawan, 2018). Web service platform used in monitoring Staffing daily attendance has been made in another study, able to display employee attendance data in json format using REST API concept. By using the REST API then can bridge various access devices information from those in need and convenience other devices to access information.

This research will describe the implementation of RESTful web services as a technology to realize interoperability that can bridge between different platforms, as well as carry out the testing process at the security stage of data exchange which is carried out using the Hash-based Message Authentication Code (HMAC) algorithm. Interoperability of information system software involves various components, which may create loopholes that can compromise system security. This study aims to explain the security feature model Json Web Token (JWT) on web service in mode system authentication. This research also discusses whether jwt can be stored on cookies or not. With application of the JWT feature model then is expected to solve security problems and information accuracy problems in web service technology which can be implemented in the future.

2. Literature Review

Various ways to reduce threats to security on web services have been carried out in previous studies, including: JSON Web Token (JWT) for authentication on Interoperability Architecture based on RESTful Web Service in research (Aldya et al., 2019). The application of JSON Web Token authentication has also been carried out by adding the RSA-512 algorithm in research (Aldya et al., 2019). Another research (Sabir et al., 2019) on authentication and load balancing schemes based on JSON Tokens for Multi-Agent Systems. However, from several previous studies that conducted research using JSON Web Tokens for authentication, they still used the RSA-512 algorithm which tends to be slow in the RestFul API process, in this study we will use another asymmetric algorithm, namely HMAC.

HMAC uses a cryptography key and a hash function to construct the message authentication code, which is then appended to the end of the message intended for the receiver. The packet will be confirmed if the message code recipient matches the message authentication code. By looking at the structure of HMAC, the researchers were able to create a new function dubbed X-HMAC. The dedicated cryptography key of each packet and the dedicated cryptography key of each packet generated from the main X-HMAC cryptography key are used in this function. It hashes message bits and HMAC using bit Swapp and rotation to the left in two steps. The findings reveal that the X-HMAC function can act as a strong barrier against data identification and HMAC attacks, preventing the attacker from simply identifying the blocks and exploiting HMAC flaws (Mousavi & Shakour, 2019).

The results of the research in (Rahmatulloh et al., 2019b) The comparison of token-based authentication performance utilizing JWT with many techniques has been explained. Overall, the adoption of HMAC outperforms the criteria of token generation time, token size, and token transfer speed.

The results of the research in (Sabir et al., 2019), JWT access control solutions for different applications developed on platforms, such as IoT and cloud systems, have been improved. Rapid production of new tokens on each client request with an O (1) time complexity on the server can prevent an attacker from identifying a client's signature.

3. Results and Discussions

E-learning, or online learning, is an educational system that provides educational services to students in a remote learning environment via the internet (Miguel et al., 2015). Some experts believe that this online learning system based on e-learning has fewer learning effects than offline learning. This is because the online learning system may create an atmosphere that allows unauthorised users to attend and take tests when the learner is engaged in something else (Lee & Han, n.d.)

REST is an architectural web service that developed from several network-based architectural styles that are often applied in web-based services (Ed-Douibi et al., 2016; Haupt et al., 2017; Neumann et al., 2021). REST architecture is typically implemented using HTTP (Hypertext Transfer Protocol), which entails reading an XML or JSON file from a specific web page. The server does not keep any request state because each request is independent. RESTful APIs are application programming interfaces (APIs) that follow the REST style. A Uniform Resource Identifier (URI) is used to represent resources in RESTful APIs. A URI link is used to identify each data source. The following are some of the REST methods: The GET method retrieves the resource, the POST method creates a new resource, and the PUT method updates the resource based on the resource. The DELETE method is used to remove a resource or a group of resources.

Json Web Token (JWT) is a lightweight means of exchanging data between two parties so as to ease authentication, authorization and security. Each JWT statement is kept as a json object, which can be used as plaintext for JSON Web Encryption or as a payload for JSON Web Signature (JWS), which enables claims to be digitally protected and authenticated using the Message Authentication Code (MAC) [6]. JWT is a token in the form of a string that is made up of three parts: header, payload, and signature. It is used for authentication and information transmission [18]. Tokens are of two types: bearer tokens and keyholder tokens. Meanwhile, based on the purpose there are two schemes: identity tokens and access tokens [19]. JWT functions similarly to a password. The server will provide a token stored in local storage or browser cookies after the user successfully signs in. The token is used to gain access to specified pages, and the user will return the token as proof of successful login.

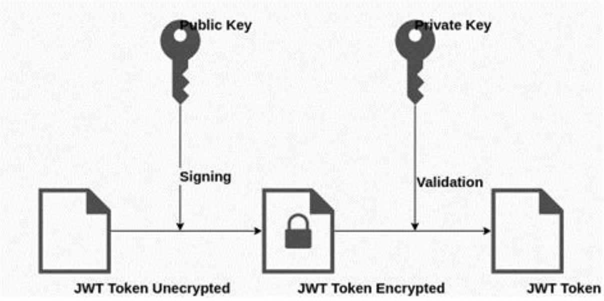

Fig. 1. Flow JSON Web Token

Figure 1 is an example of JWT flow starting from signing which is the public key, after that the JWT token is encrypted, then in the validation process the private key is matched with the public key and a JWT token is generated.

A cryptographic hash function, denoted by H, and a secret key, denoted by K, are required for the definition of HMAC. H is a cryptographic hash function that hashes data by iterating a basic compression function on blocks of data. The byte-length of such blocks is denoted by B (B=64 for all the hash methods listed above), and the byte-length of hash outputs is denoted by L (L=16 for MD5, L=20 for SHA-1). The authentication key K can have any length up to B, the hash function block length [x].

In general, this HMAC algorithm could explained by the equation below:

$$\mathrm{HMACk}\,(\mathrm{M}) \;=\; \mathrm{h}(\,\mathrm{k}||\mathrm{pad1}\;||\;\mathrm{h}(\,\mathrm{k}||\mathrm{pad2}\;||\mathrm{M}\,)) \qquad (1)$$

Where K is the private key known to both sender and receiver, h is the hash function used, m is the message to be authenticated, pad is 0x5c5c5c...5c and ipad is 0x363636...36 with length you same. The first step that must be done in this HMAC algorithm is to normalize the required key length given, so that length is equal to B. Key size K must be greater than/equal to L/2, where L is the output size of the hash function. Larger keys do not necessarily increase the security of the hash function, this is because the key will be hashed first to get more size from/same with L/2. But highly recommended for use a long key if the randomness of the key is quite weak.

If the key length is equal to B, then the value of K0 (the key that will be used for the Hash function and has been normalized) is filled with K. Meanwhile, if the key length is less than B, then the K key is appended with 0 so that the length is equal to B. For the case when the length of K exceeds the length of B, the key is hashed first. Then this hash result will be appended with 0 so that length is the same as B, the same as if the key length was shorter than B.

The next step is to do the XOR between K0 with the ipad. The result of this operation is also called the inner key. The sixth step is to calculate the hash value of the concatenation that has just been done. The next step is to calculate the value of the outer key, which is obtained from the XOR operation between K0 and the opad. Then the concatenation operation is performed again between the outer key with hash value calculated in the sixth step. The ninth step is to calculate the hash value of the concatenation result.

Then the last step is to take the leftmost t bytes of the hash value from the ninth step. This last step itself is actually optional. This is because the result of the hash function in the ninth step can already be taken as the MAC value of the message, with length L.

This tenth step can be done if: user want an authentication tag that more. But from the HMAC specification, it is suggested that it is short, taking only t bytes. This t value cannot be less than half L or not less than 80 for certain cases. If the value of t is small, then this HMAC is likely to be unsafe.

The above algorithm can be the following diagram in figure 2:

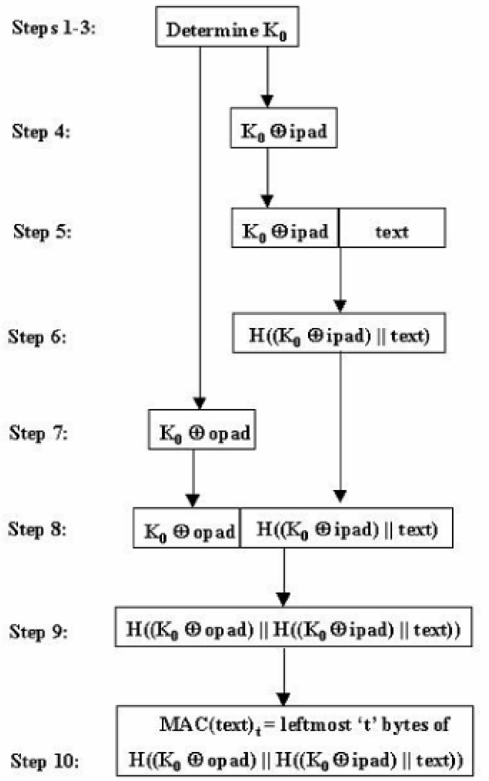
Steps 1-3:
Determine K_0
Step 4:
$K_0 \oplus ipad$
Step 5:
$K_0 \oplus ipad$
text
Step 6:
$H((K_0 \oplus ipad) \| text)$
Step 7:
$K_0 \oplus opad$
Step 8:
$K_0 \oplus opad$
$H((K_0 \oplus ipad) \| text)$
Step 9:
$H((K_0 \oplus opad) \| H((K_0 \oplus ipad) \| text))$
Step 10:
$MAC(text)_t$ = leftmost 't' bytes of
$H((K_0 \oplus opad) \| H((K_0 \oplus ipad) \| text))$

Fig. 2. Diagram Hash-based Message Authentication Code

Table 1 - The Variable List HMAC

Variable	Description
B	Block size(in bytes) of input to Hash function
H	The hash function used
opad	inner pad(0x3636...36)

Variable	Description
K	A private key is known to both sender and receiver
K_o	Keys that have been processed as necessary so that they are long B
L	Block size(in bytes) of input to Hash function
opad	outer pad(0x5c5c...5c)
t	The number of bytes of the desired MAC
text	Message/information to be manipulated

The proposed model facilitates user access to the e-learning system through a secure JWT access token using the Hash-based Message Authentication Code (HMAC) method. The design of the access token model can be seen in the following figure:

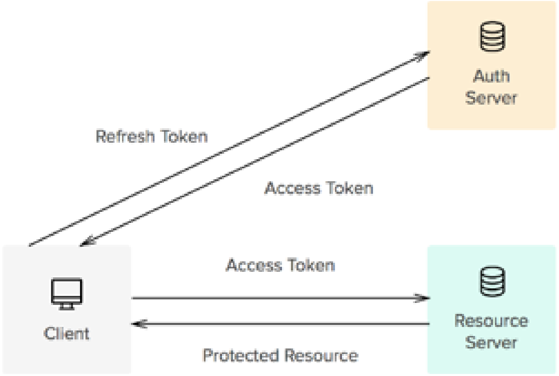

Fig. 3. Client Access Token

A JWT must be saved in a secure location within the user's browser. If you save it in localStorage, any script on your website will be able to access it. An XSS attack could provide an external attacker access to the token, which is as awful as it sounds.To be clear, you should never keep a JWT in local storage (or session storage). If one of the third-party scripts you utilize on your page gets hacked, it will have access to all of your users' tokens.

JWTs should always be stored in a httpOnly cookie to keep them safe. This is a unique cookie that is only transmitted to the server in HTTP requests. JavaScript running in the browser is never able to access it (for reading or writing).JWTs can be used as an authentication method that doesn't require the use of a database. Because the data stored in the JWT given to the client is secure, the server can avoid utilizing a database. After we know that the JWT mechanism does not require a database, in this case we simulate the storage method in local storage, namely the browser, which can be seen in figure 4.

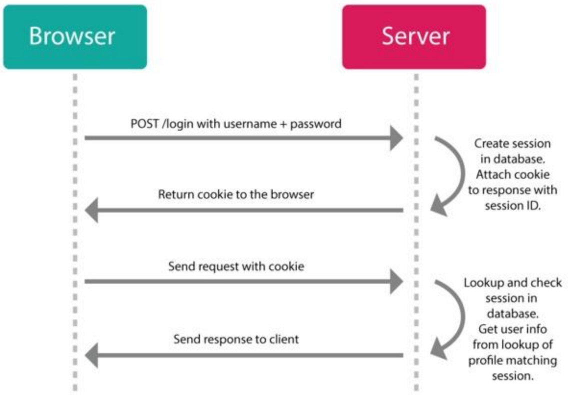
Browser
Server
POST /login with username + password
Create session in database. Attach cookie to response with session ID.
Return cookie to the browser
Send request with cookie
Lookup and check session in database. Get user info from lookup of profile matching session.
Send response to client

Fig. 4. Flow Cookie in Browser

Before storing tokens in the browser, authentication checks occur using the HMAC algorithm. The HMAC algorithm will check whether the tokens are valid or invalid and whether the token has expired or not, which can be seen in figure 5.

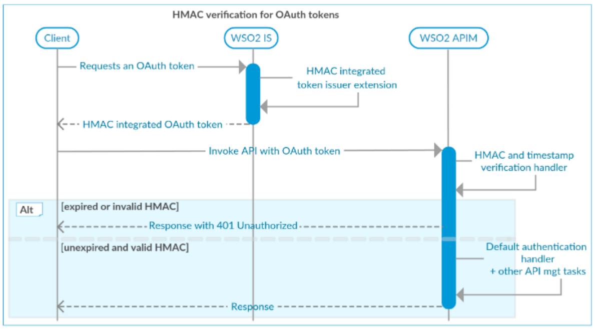
HMAC verification for OAuth tokens
Client
WSO2 IS
WSO2 APIM
Requests an OAuth token
HMAC integrated token issuer extension
HMAC integrated OAuth token
Invoke API with OAuth token
HMAC and timestamp verification handler
Alt
[expired or invalid HMAC]
Response with 401 Unauthorized
[unexpired and valid HMAC]
Default authentication handler + other API mgt tasks
Response

Fig. 5. HMAC algorithm Process

The model offered by implementing a Json Web Token which is stored in cookies and later secured with the HMAC algorithm makes the authentication process more secure and speeds up the authentication process because the client does not ask the server continuously. Then this model is very suitable to be applied by systems that use integration.

4. Conclusion

Storing JSON Web Token into local storage in the client browser using the HMAC algorithm has been presented in this paper. The proposed work has shown that JWTs do not need to be stored on the server but can be stored on the client browser side using local storage. JWT eliminates the requirement for the server to store data because it is stateless. Data such as scope, authorisation, and a user can be stored in JWT. When applied to a centralized authentication system that uses single sign-on, this notion is excellent. It will be tested in the future to determine if the mechanism of storing tokens on local storage (cookies) is secure and how the right token can be preserved (Gunawan & Rahmatulloh, 2018). Therefore, what can be concluded from this paper is that the proposed work has been carried out properly so that it can be applied according to its needs.

References

Adam, S. I., Moedjahedy, J. H., & Maramis, J. (2020, October 27). RESTful Web Service Implementation on Unklab Information System Using JSON Web Token (JWT). *2020 2nd International Conference on Cybernetics and Intelligent System, ICORIS 2020*. https://doi.org/10.1109/ICORIS50180.2020.9320801

Alam, M. S., Atmojo, U. D., Blech, J. O., & Lastra, J. L. M. (2020). A REST and HTTP-based Service Architecture for Industrial Facilities. *Proceedings - 2020 IEEE Conference on Industrial Cyberphysical Systems, ICPS 2020*, 398–401. https://doi.org/10.1109/ICPS48405.2020.9274792

Aldya, A. P., Rahmatulloh, A., & Arifin, M. N. (2019). Stateless Authentication with JSON Web Tokens using RSA-512 Algorithm. *JURNAL INFOTEL*, *11*(2), 36. https://doi.org/10.20895/infotel.v11i2.427

Ed-Douibi, H., Izquierdo, J. L. C., Gómez, A., Tisi, M., & Cabot, J. (2016). EMF-REST: Generation of RESTful APIs from models. *Proceedings of the ACM Symposium on Applied Computing*, *04-08-April-2016*, 1446–1453. https://doi.org/10.1145/2851613.2851782

Ethelbert, O., Moghaddam, F. F., Wieder, P., & Yahyapour, R. (2017). A JSON token-based authentication and access management schema for cloud SaaS applications. *Proceedings - 2017 IEEE 5th International Conference on Future Internet of Things and Cloud, FiCloud 2017*, *2017-January*, 47–53. https://doi.org/10.1109/FiCloud.2017.29

Gunawan, R., & Rahmatulloh, A. (2018). *Optimasi Sistem Informasi Akademik View project Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) Studi Kasus STIKes BTH Tasikmalaya View project*. https://www.researchgate.net/publication/332278532

Haupt, F., Leymann, F., Scherer, A., & Vukojevic-Haupt, K. (2017). A Framework for the Structural Analysis of REST APIs. *Proceedings - 2017 IEEE International Conference on Software Architecture, ICSA 2017*, 55–58. https://doi.org/10.1109/ICSA.2017.40

Insights of JSON Web Token. (2020). *International Journal of Recent Technology and Engineering*, *8*(6), 1707–1710. https://doi.org/10.35940/ijrte.f7689.038620

Kumari, A., Yahya Abbasi, M., Kumar, V., & Khan, A. A. (2019). A secure user authentication protocol using elliptic curve cryptography. *Journal of Discrete Mathematical Sciences and Cryptography*, *22*(4), 521–530. https://doi.org/10.1080/09720529.2019.1637155

Lee, A., & Han, J.-Y. (n.d.). Effective User Authentication System in an E-Learning Platform. In *International Journal of Innovation, Creativity and Change. www.ijicc.net* (Vol. 13). www.ijicc.net

Miguel, J., Caballé, S., Xhafa, F., & Prieto, J. (2015). Security in online web learning assessment: Providing an effective trustworthiness approach to support e-learning teams. *World Wide Web*, *18*(6), 1655–1676. https://doi.org/10.1007/s11280-014-0320-2

Mousavi, S. M., & Shakour, Dr. M. H. (2019). Increasing Cryptography Security using Hash-based Message Authentication Code. *International Journal of Engineering and Technology*, *11*(4), 1046–1056. https://doi.org/10.21817/ijet/2019/v11i4/191104086

Neumann, A., Laranjeiro, N., & Bernardino, J. (2021). An Analysis of Public REST Web Service APIs. *IEEE Transactions on Services Computing*, *14*(4), 957–970. https://doi.org/10.1109/TSC.2018.2847344

Perkasa, M. I., & Setiawan, E. B. (2018). Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token. *ULTIMA Computing*, *X*(1).

Rahmatulloh, A., Gunawan, R., & Nursuwars, F. M. S. (2019a). Performance comparison of signed algorithms on JSON Web Token. *IOP Conference Series: Materials Science and Engineering*, *550*(1). https://doi.org/10.1088/1757-899X/550/1/012023

Rahmatulloh, A., Gunawan, R., & Nursuwars, F. M. S. (2019b). Performance comparison of signed algorithms on JSON Web Token. *IOP Conference Series: Materials Science and Engineering*, *550*(1). https://doi.org/10.1088/1757-899X/550/1/012023

Sabir, B. E., Youssfi, M., Bouattane, O., & Allali, H. (2019). Authentication and load balancing scheme based on JSON Token for Multi-Agent Systems. *Procedia Computer Science*, *148*, 562–570. https://doi.org/10.1016/j.procs.2019.01.029

Tihomirovs, J., & Grabis, J. (2017). Comparison of SOAP and REST Based Web Services Using Software Evaluation Metrics. *Information Technology and Management Science*, *19*(1). https://doi.org/10.1515/itms-2016-0017