IT Journal Research and Development (ITJRD)
Vol.8, No.1, August 2023, E-ISSN: 2528-4053 | P-ISSN: 2528-4061
DOI : 10.25299/itjrd.2023.12029

RESTFUL API SECURITY USING JSON WEB TOKEN (JWT) WITH HMAC-SHA512 ALGORITHM IN SESSION MANAGEMENT

Syabdan Dalimunthe, Emansa Hasri Putra, Muhammad Arif Fadhly Ridha
Departement of Computer Engineering, Politeknik Caltex Riau
syabdan20s2tk@mahasiswa.pcr.ac.id, emansa@pcr.ac.id, fadhly@pcr.ac.id

Article Info

Article history:

Received Jan 28, 2023
Revised Aug 08, 2023
Accepted Dec 05, 2023

Keyword:

API
Cookies Browser
JSON Web Token
HMAC-SHA512
Restfull
Web Service

ABSTRACT

Applications or information systems are technologies that can help work systematically. However, the existing systems or applications are not yet integrated with one another, making many processes have the same function on different systems, for example the authentication process is built using the web service concept. Integration or interoperability of information system software involving various components, which may create gaps that can disrupt system security. In this study, security has been implemented in web services using JSON Web Token (JWT) with the HMAC-SHA512 algorithm which is stored in browser cookies. From the research results, this concept is very suitable to be applied to applications or information systems on different platforms that use the same service, JWT tokens are also successfully stored in browser cookies. In addition, a comparison of the HMAC-SHA512 and HMAC-SHA256 algorithms was also carried out. In the final results of the serial test, it was found that the total time difference was 75 ms and the average time difference was 2.5 ms. It can be concluded that the HMAC-SHA256 algorithm is faster 0, 45% compared to the HMAC-SHA512 algorithm in serial trials. Meanwhile, in the final results of parallel testing, it was found that the total time difference was 185 ms and the average time difference was 6.16 ms. It can be concluded that the HMAC-SHA512 algorithm is 1.4% faster than the HMAC-SHA256 algorithm. The speed of the HMAC-SHA algorithm is also affected by the network and connection when accessing a web service endpoint.

Corresponding Author:

Syabdan Dalimunthe
Applied Master of Computer Engineering Program
Politeknik Caltex Riau
Jl. Umban Sari (Patin) No. 1 Rumbai, Pekanbaru, Riau
Email: syabdan20s2tk@mahasiswa.pcr.ac.id

1. INTRODUCTION

Applications or information systems are technologies used in various companies. Generally, every company has various kinds of systems or applications that are used to help employees do work systematically. However, the existing systems or applications are not yet integrated with one

another, making many processes have the same function on different systems, for example the authentication process is built using the web service concept. [1]–[3].

Integration or interoperability of information system software involving various components, which may create gaps that can disrupt system security. Various ways to reduce threats to security in web services have been carried out in previous studies, including JSON Web Token (JWT) for authentication in RESTful Web Service-based Architecture Interoperability in research [4]. The implementation of JSON Web Token authentication has also been carried out by adding the RSA-512 algorithm by research [5].

Other research [6] bout the authentication and load balancing scheme based on JSON Tokens for Multi-Agent Systems. However, from several previous studies that conducted research using JSON Web Token for authentication and data exchange, they still used the RSA-512 algorithm which tends to be slow in the RestFul API process, this research will use another algorithm, namely HMAC SHA-512 which uses a token storage mechanism. on search engine local storage (cookies).

Cookies are also known as HTTP Cookies, Web Cookies and also Browser Cookies. When viewed from its location and use, cookies can be interpreted as records used by website applications to send information in the form of status to visitors' web browsers which are used as web server reminders for their visitors. [7]. Because JWT is stateless, the server does not need to store data because data that is often needed by the system, such as user data and authorization, can all be stored in JWT. This concept is very suitable to be applied to systems that have more than one server. According to research conducted by [4][8] The concept of local storage (cookies) needs to be tested on how to store the tokens in these cookies.

2. RESEARCH METHOD

2.1. Related research

The literature study in this study is a literature study, where the literature referred to according to previous researchers is as follows:

Based on research conducted by [8], create a model for storing tokens into local storage (cookies) using JSON Web Token (JWT) with HMAC (Hash-based Message Authentication Code) on e-learning systems. From the results of this study, it is possible that JWT tokens can be stored in browser cookies and secured using the HMAC algorithm.

Based on research conducted by [5], discusses research on Stateless Authentication with JSON Web Tokens Using the RSA-512 Algorithm, where this research examines user authentication using JSON Web Tokens using the RSA-512 algorithm. In this study, we apply this algorithm to JWT in SOAP and Restful. So it can be concluded that the Restful process speed is 24.69% better than SOAP, while for authentication speed using JWT RSA-512 Restful speed is 11.64% better than SOAP and for generated tokens the Restful process is 1.25% better than SOAP.

Based on research conducted by [9], conducting research integrating academic information systems with payment systems that overcome problems related to manual and repetitive data input. The implementation is carried out using a host-to-host web service so that data updates between systems can be carried out.

Then further research was carried out by [10] who conducted research comparing the HMAC, RSA, ECDSA algorithms in system authentication using JSON Web Token. Based on the results of research the HMAC algorithm is the best algorithm of all algorithms tested with an average token generation time of 21m3 seconds, a token size of 109 bytes and a data transfer speed of 91.2 seconds.

Further research conducted by [4], conducting research to address the problem of search and distribution of blood donors, thus requiring data integration. From these problems, integration was carried out with pre-existing systems using web services, but they were still constrained by REST, namely regarding data security and the authentication process. The REST architecture requires stateless authentication, one of which is using JSON Web Token Authentication in web services. The results of this study indicate that the use of JSON Web Token Authentication on web services and the Blood Donors Backend system can form a system that is highly scalable, secure, capable of multi-platform interaction and reliable.

Subsequent research refers to research conducted by [11], the study searched 4000 popular sites on alexa.com to identify exactly 500 websites claiming to provide REST web service APIs. From the analysis that has been done, the level of compliance with the REST architecture, for example, resource addressing capabilities and for compliance with API versioning. Compliance is only 0.8% of services that fully comply with REST principles. Thus, this research can benefit and contribute to the development of web services.

The update of this research with the research described earlier is found in the problem under study, namely securing the JSON Web Token in the authentication process and data exchange in session management using the HMAC SHA512 algorithm.

2.2. Web Service (WS)

Web service is a software that is not affected by platform, architecture, or programming language, which provides services or methods for exchanging data that can be accessed by the network [12]. In general, web services can be identified by using a URL (Uniform Resource Locator) like just the web in general (for example: https://pcr.ac.id). However, what distinguishes web service URLs from web URLs in general is the interaction provided by the web service. In contrast to web URLs in general, web service URLs only contain a collection of information, commands, configurations or syntax that are useful for building certain functions of the application. Web service can also be interpreted as a method of exchanging data, regardless of where a database is embedded, made in what language an application consumes data, and on what platform the data is consumed. Web service is able to support interoperability. So that the web service is able to become a bridge between the various existing systems.

2.3. Hypertext Transfer Protocol

Hypertext Transfer Protocol (HTTP) merupakan prokotol aplikasi yang umumnya digunakan pada World Wide Web (WWW) dan merupakan fondasi dari komunikasi data WWW [13]. Protokol HTTP merupakan aturan komunikasi yang harus dipenuhi pada saat menggunakan protokol HTTP. Penerapan protokol HTTP diterapkan pada arsitektur client-server yang menggunakan pola Representational State Transfer (REST). Dalam layer OSI protokol HTTP terdapat pada layer application dan terdapat 2 jenis operasi pada protokol HTTP yaitu HTTP request dan HTTP response yang dapat diidentifikasi melalui header HTTP tersebut. Hypertext Transfer Protocol Secure (HTTPS) merupakan ekstensi protokol HTTP di mana menggunakan protokol Transport Layer Security (TLS) untuk mengenkripsi data komunikasi.

2.4. Representational State Transfer (REST)

REST is a web service architecture developed from several network-based architectural styles that are often applied in web-based services [14]. REST architecture usually runs over HTTP (Hypertext Transfer Protocol), involving the process of reading certain web pages that contain

XML or JSON files. Each request is independent, the server does not save any state of the request. An Application Programming Interface (API) that follows the REST style is called a RESTful API. The RESTful API uses a Uniform Resource Identifier (URI) to represent resources. Each data source is identified using a URI link. The methods used in REST include: GET to get resources, POST is used to create new resources and PUT method is used to update resources based on resources. Meanwhile, the DELETE method is used to delete a resource or a collection of resources.

2.5. Javascript Object Notation (JSON)

Javascript Object Notation (JSON) is a format for exchanging data between machines that is easy for humans to read and write (JSON.org). JSON is based on part of the javascript programming and is standardized in the ECMA-263 document. JSON is a format that does not depend on any programming language because it uses a common language style, therefore JSON is ideal as a data exchange format. The JSON structure is declared as an object where the object members are in the form of key and value pairs or can be JSON objects which are separated by commas.

2.6. JSON Web Token (JWT)

JWT is a token in the form of a string consisting of three parts, namely: header, payload and signature which are used for authentication and information exchange. [15]. Tokens are of two types: bearer tokens and keyholder tokens. Meanwhile, based on the purpose, there are two schemes: identity tokens and access tokens [16]. The JSON Web Token structure can be seen in Figure 1 below.

Figure 1 Json Web Token Structure

The way JWT works is the same as a password, when the user successfully logs in, the server will provide a token that is stored in local storage or browser cookies. The token is used to access certain pages, the user will send the token back as proof that the user has successfully logged in.

2.7. Keyed-Hash Massage Authentication Code (HMAC)

HMAC is a message authentication technique by utilizing the hash function of the message and then encrypting the message using the private key. HMAC was created by Mihir Bellare, Ran Canetti, and Hugo Krawczyk in 1996. HMAC provides a way to check the integrity of information transmitted or stored in unreliable media which is a major requirement in the world of open computing and communications [17]. The mechanism that provides such integrity checks based on a secret key is usually called a message authentication code (MAC). Typically, authentication code messages are used between two parties that share a secret key to authenticate information sent

between these parties. This standard defines a MAC that uses a cryptographic hash function in conjunction with a secret key. This mechanism is called HMAC. HMAC will use an approved cryptographic hash function.

The HMAC equation is as follows [17]:

$$\mathrm{MAC(text)} = \mathrm{HMAC(K, text)} = \mathrm{H((K0 \oplus opad) || H((K0 \oplus ipad) || text))} \quad (1)$$

HMAC uses a secret key for MAC computation and verification. The HMAC algorithm can be described in ten steps, as shown in Figure 2.

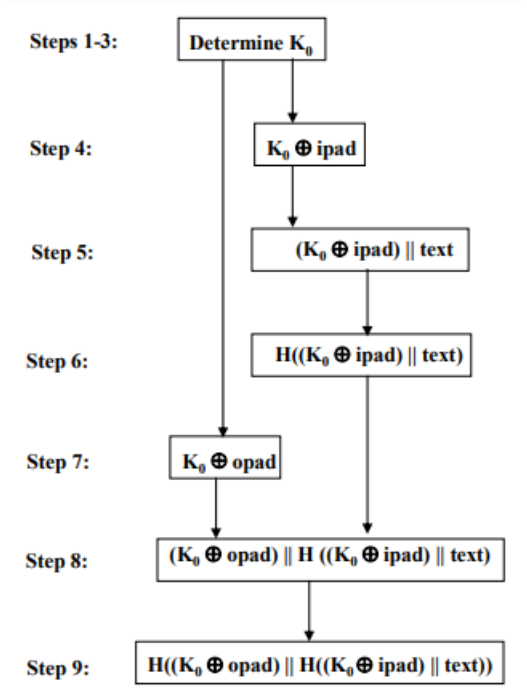

Figure 2 HMAC Diagram [17]

2.8. Secure Hash Algorithm (SHA)

SHA is a one-way hash function created by the National Institute of Standards and Technology (NIST) and used with the Digital Signature Standard (DSS) [18]. The equation of the hash function is as follows [19] which can be seen in the following equation 2.

$$h = H(M) \quad (2)$$

Hash properties are as follows [19]:

1. The H function can be used on data blocks that have any data capacity.
2. The H function produces a value (h) with a fixed-length output.
3. H(x) can be easily obtained from any given value of x.

4. Every result of the value of h is impossible to return the value of x so that H(x) = h. Therefore, the function H is called a one-way hash function (one-way hash function).
5. For each entered x value, you cannot search y ≠ x so that H(y) = H(x).
6. Cannot find pairs of x and y so that H(x) = H(y).

2.9. Postman

Postman is an application that functions as a REST CLIENT for testing REST APIs. Postman is commonly used by API developers as a tool to test the API they have created [20]. Postman itself has features that can be used individually or in groups (teams), can also be used for free or paid. Postman can also be used to collect APIs that can be made into a complete documentation for a particular project. If the API documentation is made complete by utilizing Postman it will make the project development process easier, because each developer can have clear references for using each API.

3. RESULTS AND ANALYSIS

3.1. Model Design

The functional requirements of this system depart from the background of research problems, namely authenticating with the web service concept using JSON Web Token with the HMAC-SHA512 algorithm which is stored in browser cookies. This research applies the authentication process to a system. The authentication process will be carried out on several different systems. The employee authenticates only by logging in using a username and password. Username and password will be provided by default for users to reduce error level function for each user. If authentication is possible and successful, the JSON Web Token is assigned by the REST server and then validates the token. If the token provided is valid, the client will be given encrypted JSON Web Token data from the REST server, so that it can access information and use the application.

a. Restfull API architecture

In handling the authentication and authorization process on the RESTful API, it is necessary to have the role of the backend system. This system functions as a JWT provider and regulator. JWT is public key encryption with backend secret key. The scheme for implementing JSON Web Token in applications based on Restful API can be seen in figure 3 below.

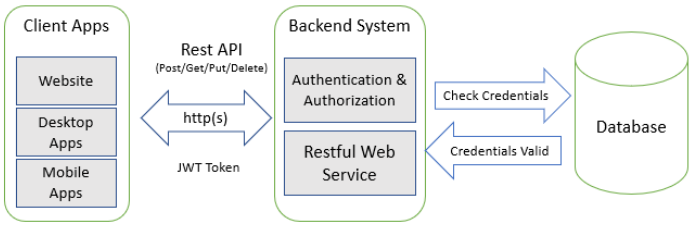

Figure 3 JWT Implementation Scheme

b. JSON Web Token Verification

The way JSON Web Token works in verifying incoming data starts with receiving the token. Then the token is validated and detected whether the token is valid. If yes, then the signature is validated, if not, then the token is invalid and an error message is given. After

that, the validation time is corrected whether it is still valid or expired. If yes, it will check the time before it expires, if not, it will be labeled as an invalid token and an error message will be displayed. If the time does not exceed the expiration or expired then a session is created. If not, the token has expired and an error message appears. After the session is created, services will be provided to the information system according to previously verified data. The JSON Web Token verification workflow can be seen in figure 4 below.

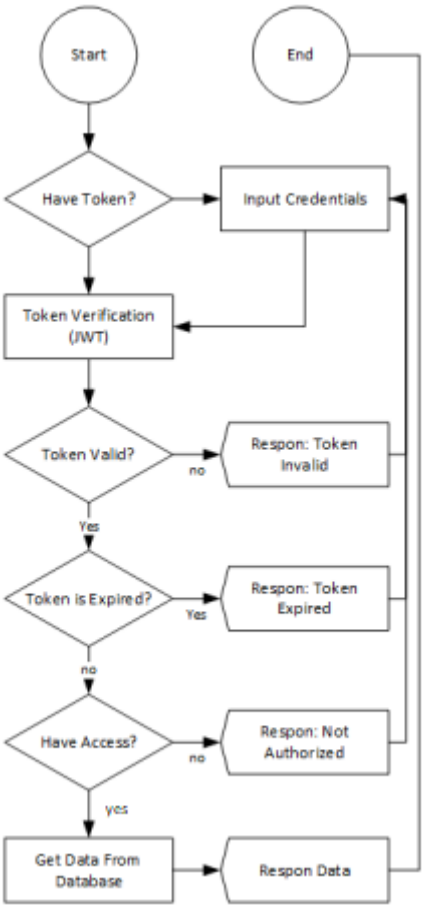

Figure 4 JWT Verification Workflow

c. Storing Token

In authentication or authorization, we generally use a session, which is when a user logs into a web, the server will store the user's data. The stored session data will be used to verify authentication, to make sure the user is logged in or not. The application can be seen in figure 5 below.

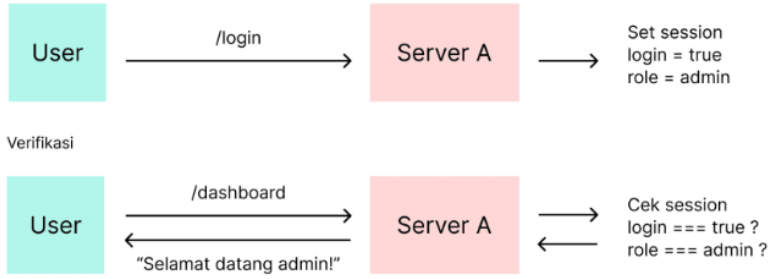

Figure 5 Single Server Application Authentication

So far there is no problem with the authentication method in figure 5 if the application architecture is not integrated with several applications or servers. However, if you need integration or more application sharing and want to be able to share authentication without having to create multiple authentications, of course the method in figure 5 is not possible to apply, because user sessions are only stored by the website the user used to log in previously. This can be seen in figure 6 below.

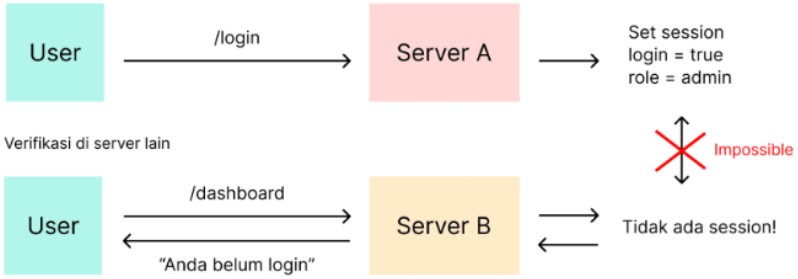

Figure 6 Imposible Session Implementation

From the story above, the role of JWT is present as a solution to the problem. Since the validity of a JWT can be independently verified using the signature, it is possible for other applications to use the token provided they have the same secret key. Then the token is used to access modules in applications that require authentication. This can be seen in figure 7 below.

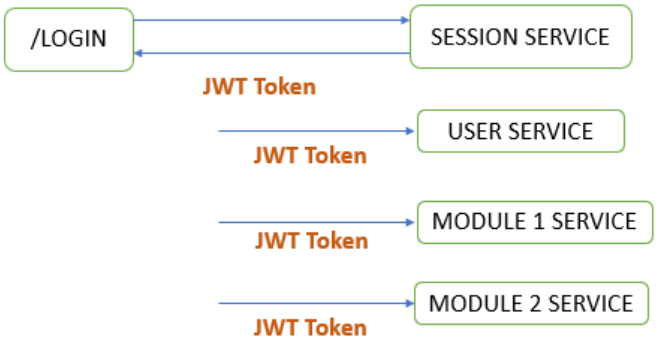

Figure 7 Access JWT Various Applications

3.2. Testing and Analysis

a. System Functional Testing

System functional testing can be carried out to ensure that the REST API system functionality that is built can run without encountering error problems. This test is carried out by testing that focuses on the functionality side where each URL resource from the REST API will be tested, especially the input and output of the application. Black box testing or also known as Behavioral Testing is a test that is carried out to observe the input and output results of the software without knowing the code structure of the software. This test is carried out at the end of making the software to find out whether the software can function properly.

System functional testing with blackbox testing has been successfully carried out by testing the login and logout processes on 2 different systems with the same REST API. In the black box testing test, no errors were found and it went according to the expectations of the experiment, so it can be concluded that the functional system can run well without finding problems.

b. Authentication Time Testing

Authentication time testing is a test carried out to find out and compare the time used when authenticating using the HMAC-SHA512 and HMAC-SHA256 algorithms.

The procedure for testing the authentication time is carried out by first authenticating the user at the /api/login POST URL resource. Then authenticate 30 times for each algorithm, namely HMAC-SHA512 and HMAC-SHA256 to find out how much time each trial takes. This test was carried out as many as 30 trials on each algorithm with the same scenario. According to [21] when the number of samples is 30 or more there is a normality value in the sample, the normality value will affect the results of the test. At the end of the experiment, the overall value of the experiment is stored and the average time taken then the results will be displayed. In the experimental algorithm, each authentication process runs sequentially, meaning the authentication process runs one by one.

Table 1. Table of Authentication Speed Comparison Experiment

Trial to	HMAC-SHA512 (ms)	HMAC-SHA256 (ms)
1	578	517
2	442	444
3	454	394
4	421	417
5	418	459
6	377	600
7	454	502
8	428	441
9	430	407
10	491	468
11	472	510

12	451	437
13	431	411
14	446	405
15	382	412
16	389	393
17	411	508
18	447	496
19	462	415
20	419	435
21	443	411
22	403	411
23	441	391
24	416	423
25	418	438
26	425	578
27	479	398
28	526	386
29	376	420
30	432	420
Total	13162	13347
Average	438.7333333	444.9

Authentication trials can also be seen in graphical form in Figure 3 below.

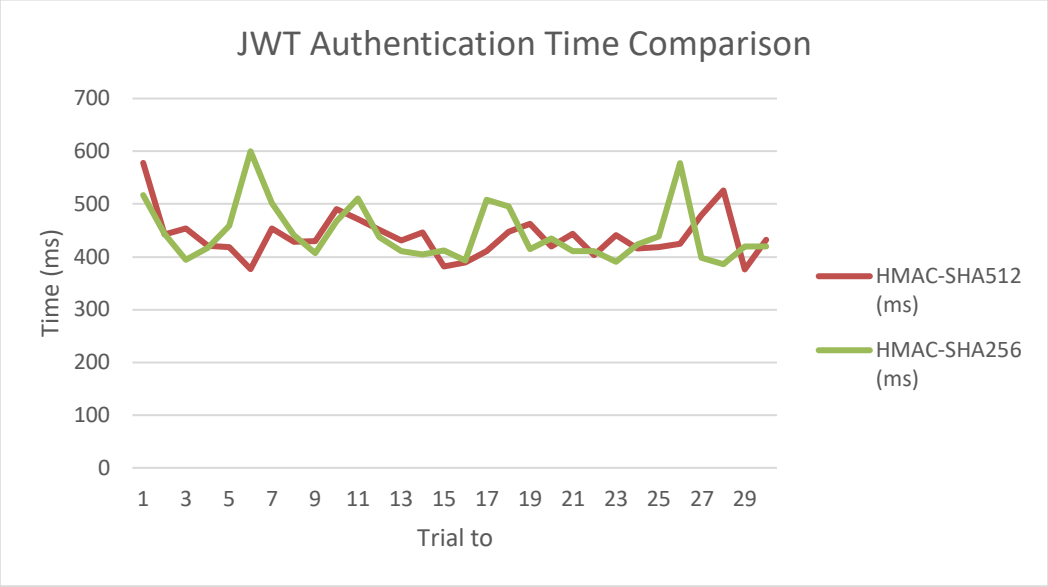

Figure 8 JWT Authentication Time Comparison

Authentication time testing has been carried out on each algorithm, both the HMAC-SHA512 and the HMAC-SHA256 algorithms. In the final results, it was found that the total time difference was 185 ms and the average time difference was 6.17 ms. It can be concluded that the HMAC-SHA512 algorithm is 0.9861% faster than the HMAC-SHA256 algorithm.

c. Security Testing

In this scenario, we will try using two browsers, namely Google Chrome and Microsoft Edge. The first attempt to login in the chrome browser. After the user has successfully logged in, the system will be directed to the /api/user page, then it checks whether the token in the browser cookie is stored or not. The explanation can be seen that the JWT token has been stored in the chrome browser cookies in the image.

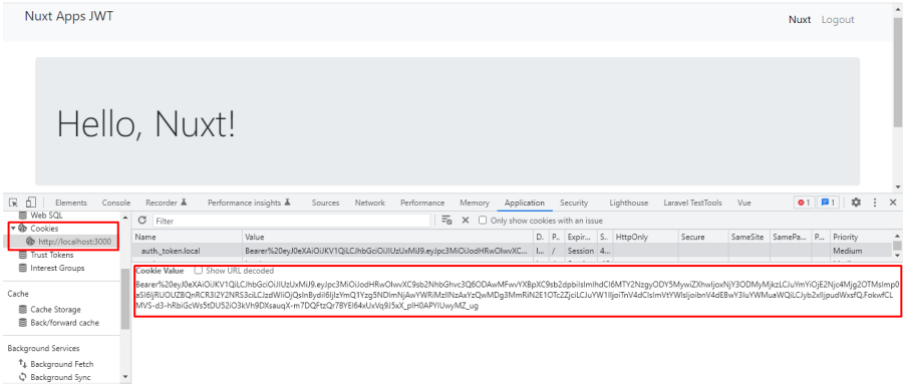

Figure 9 JWT in Cookies Browser

The next scenario is to experiment with changing the JWT token that is copied from the browser cookies and modified using the jwt.io editor as shown in figure 9 below.

.

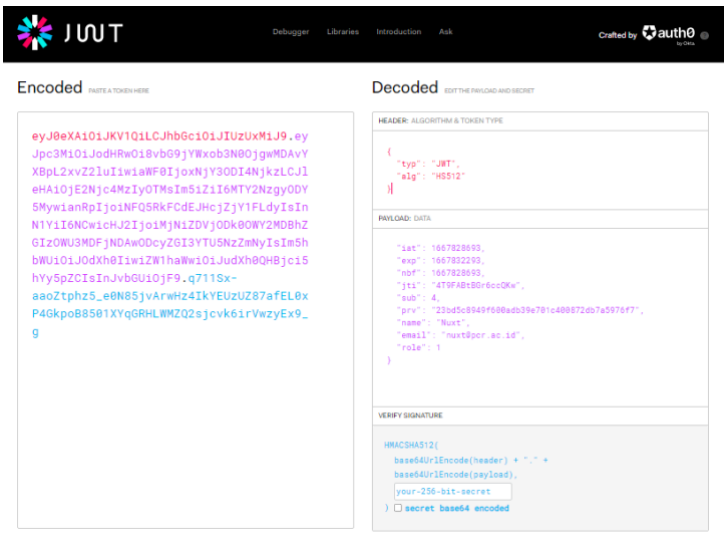

Figure 10 Decode Token in jwt.io

In the figure you can see the payload data that has been decoded from the encoded JWT token. In this session the role_id data will be changed to 2 so that the JWT token in the encode editor changes. Next, the new JWT token is copied and pasted into the Postman tools. After the JWT token has been copied to the postman tools, then click the send button to run the REST API GET /api/user command and the results can be seen in the body as shown in figure 11 below.

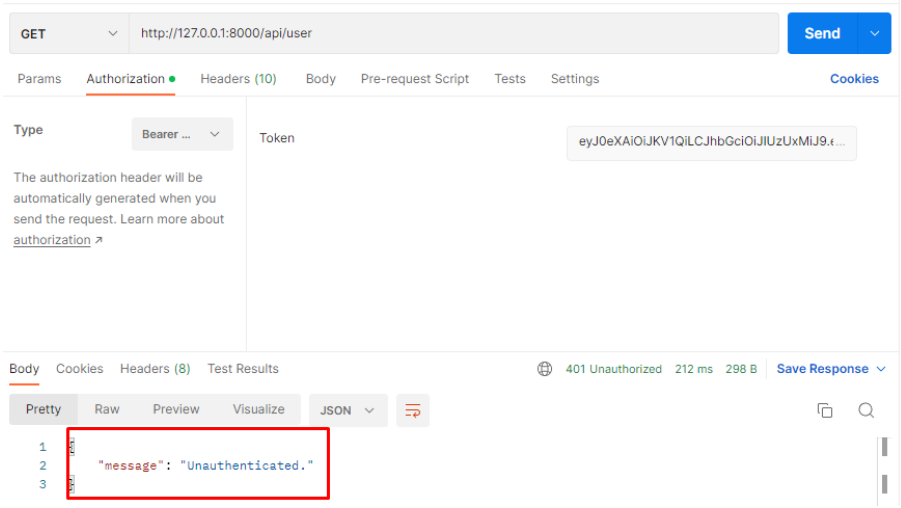

Figure 11 User Unauthenticated API

The response from the REST API GET /api/user generates code 410 Unauthorized which means the token is not valid because previously we changed the contents of the payload data.

4. CONCLUSION

Based on the results of the research, testing and analysis that has been carried out, several conclusions can be drawn regarding securing the restfull API using the JSON Web Token with the HMAC-SHA512 algorithm as follows:

1. The implementation of the HMAC-SHA512 algorithm on JSON Web Token was successfully carried out on a web service API with two different frontend frameworks.

2. In comparison, the HMAC-SHA512 algorithm is faster than the HMAC-SHA256 algorithm in the authentication tests carried out.
3. Token storage has been successfully stored in browser cookies and secured with JSON Web Token using the HMAC-SHA512 algorithm.

REFERENCES

[1] Y. Yu, J. Lu, J. Fernandez-Ramil, and P. Yuan, "Comparing Web Services with other Software Components," in *IEEE International Conference on Web Services (ICWS 2007)*, 2007, pp. 388–397. doi: 10.1109/ICWS.2007.64.

[2] I. Indu and P. M. R. Anand, "Identity and access management for cloud web services," in *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, 2015, pp. 406–410. doi: 10.1109/RAICS.2015.7488450.

[3] A. Neumann, N. Laranjeiro, and J. Bernardino, "An Analysis of Public REST Web Service APIs," *IEEE Trans Serv Comput*, vol. 14, no. 4, pp. 957–970, Jul. 2021, doi: 10.1109/TSC.2018.2847344.

[4] R. Gunawan and A. Rahmatulloh, "JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service," *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 5, no. 1, p. 74, Apr. 2019, doi: 10.26418/jp.v5i1.27232.

[5] A. P. Aldya, A. Rahmatulloh, and M. N. Arifin, "Stateless Authentication with JSON Web Tokens using RSA-512 Algorithm," *JURNAL INFOTEL*, vol. 11, no. 2, p. 36, Jun. 2019, doi: 10.20895/infotel.v11i2.427.

[6] B. E. Sabir, M. Youssfi, O. Bouattane, and H. Allali, "Authentication and load balancing scheme based on JSON Token for Multi-Agent Systems," in *Procedia Computer Science*, 2019, vol. 148, pp. 562–570. doi: 10.1016/j.procs.2019.01.029.

[7] "Support Microsoft," *https://support.microsoft.com/en-us/topic/description-of-cookies-ad01aa7e-66c9-8ab2-7898-6652c100999d*.

[8] S. Dalimunthe, J. Reza, and A. Marzuki, "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," *Journal of Applied Engineering and Technological Science (JAETS)*, vol. 3, no. 2, pp. 149–155, 2022.

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

[10] A. Rahmatulloh, R. Gunawan, and F. M. S. Nursuwars, "Performance comparison of signed algorithms on JSON Web Token," in *IOP Conference Series: Materials Science and Engineering*, Aug. 2019, vol. 550, no. 1. doi: 10.1088/1757-899X/550/1/012023.

[11] A. Neumann, N. Laranjeiro, and J. Bernardino, "An Analysis of Public REST Web Service APIs," *IEEE Trans Serv Comput*, vol. 14, no. 4, pp. 957–970, Jul. 2021, doi: 10.1109/TSC.2018.2847344.

[12] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, "Web Services," in *Web Services: Concepts, Architectures and Applications*, G. Alonso, F. Casati, H. Kuno, and V. Machiraju, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 123–149. doi: 10.1007/978-3-662-10876-5_5.

[13] J. G. J. M. H. F. L. M. P. L. T. B.-L. Roy Fielding, "Hypertext Transfer Protocol--HTTP/1.1," *RFC Editor*, Jun. 1999.

[14] R. T. Fielding, D. Software, and R. N. Taylor, "Principled Design of the Modern Web Architecture," 2000.

[15] M. Jones, "Internet Engineering Task Force (IETF)," 2015, [Online]. Available: http://www.rfc-editor.org/info/rfc7519.

[16] K. Zheng and W. Jiang, "A Token Authentication Solution for Hadoop Based on Kerberos Pre-Authentication," 2014.

[17] C. M. Gutierrez and J. M. Turner, "FIPS PUB 198-1 The Keyed-Hash Message Authentication Code (HMAC) CATEGORY: COMPUTER SECURITY SUBCATEGORY: CRYPTOGRAPHY," 2008. doi: https://doi.org/10.37385/jaets.v3i2.662.

[18] E. Conrad, S. Misenar, and J. Feldman, "Chapter 6 - Domain 5: Cryptography," in *CISSP Study Guide (Second Edition)*, E. Conrad, S. Misenar, and J. Feldman, Eds. Boston: Syngress, 2012, pp. 213–255. doi: https://doi.org/10.1016/B978-1-59749-961-3.00006-6.

[19] F. Piper and S. Murphy, *Cryptography: A Very Short Introduction*. OUP Oxford, 2002. [Online]. Available: https://books.google.co.id/books?id=UR43gHmlI1YC

[20] Antares, "Postman," 2022. https://antares.id/id/postman.html (accessed Dec. 03, 2022).

[21] Joseph F. Hair, William C. Black, Barry J. Babin, and Rolph E. Anderson, *Multivariate Data Analysis*. Cornell University: Prentice Hall, 2010.