APPLIED RESEARCH PLAN

Teachers: Kiavash Bahreini, Márcio Paixão Dantas

Student Name: Omar Abou Dehn

Student Number: 3560813

Project Git Repository: https://git.fhict.nl/I407846/safar_travelapp

Document Version History:

Version	Date	Changes
0.1	07/03/2022	Document setup
0.2	09/03/2022	Added problem definition, main research question and sub questions
0.3	12/03/2022	Defined Dot framework, added methods of research
1.0	29/05/2022	Answered the research questions

CONTENTS

PROBLEM DEFINITION:

The customers of Safar travel application would like to use the website securely without exposing sensitive data, one way to accomplish is by using **JSON web token**, which is the topic of this research.

MAIN QUESTION:

How can JSON web token be used to ensure security in a web application?

SUB QUESTIONS:

1. What is JSON web token?
2. What is JWT used for?
3. How to implement JWT properly?

DOT FRAMEWORK:

The Development Oriented Triangulation (DOT) framework will be used to conduct this research.

“The DOT framework can help you to structure your research and to communicate about it. The Development Oriented Triangulation (DOT) framework consists of three levels:

1. The "What" of your research (the domains)
2. The "Why" of your research (the trade-offs)
3. The "How" of your research (the strategies and methods)”

(The DOT Framework, 2021)

METHODS:

The research strategy that will be used for this research is Library, specifically:

1. Literature study.
2. Best good and bad practice.
3. Community research.
4. Prototyping.

RESULTS:

WHAT IS JSON WEB TOKEN?

JSON Web Token is an open industry standard that is used to exchange data between two entities, typically a client and a server.

They contain JSON objects that have the information that must be shared. Each JWT consists of three parts separated by a dot, those parts are:

1. **Header**
   The header consists of two parts:
   - The signing algorithm being used.
   - The type of token, which is in this case mostly “JWT”.
2. **Payload**
   The payload contains the claims (user attributes) as well as additional data such as issuer and expiration time.
3. **Signature**
   The signature is a hash of the JWT payload and header sections. The signature is created using the same algorithm that is described in the JWT's header section. The signature ensures that the JWT token was not tampered with or altered while in transit. It can also be used to verify who is sending the message.

Furthermore, JWT can also be encrypted to restrict the visibility of the data it contains for added security.

“JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the HMAC algorithm) or a public/private key pair using RSA or ECDSA.”

(JSON Web Token, n.d.)

WHAT IS JWT USED FOR?

JWT is frequently used in authorization. They can be signed with either a private key or a public/private key pair. After a user authenticate, the server will require the JWT in each subsequent request, check its validity, and (if the token is not tampered with or expired) allow the user to access routes, services, and data that have been permitted with that token.

Because the token is saved on the client side, the server and client have a stateless connection, where the server does not need to store any information about the logged-in user. As long as they share the same secret key, we can grant the client access to several operational servers or multiple instances of the same server.

HOW TO IMPLEMENT JWT PROPERLY?

For the implementation, the focus will be on signed JWT only and not the encrypted.

As mentioned before, the client sends a log-in/registration request to server which contains the necessary data to authenticate the user.
If the authentication is successful, the server uses the chosen algorithm which utilizes the secret key, defined on the server, to generate the token.

The token is then sent back to the client within a response body which in turn extracts the token and stores it within the browser, most commonly in local storage.

After that the client is expected to send the token with each following request usually within the authorization header.

Then the server will use the secret key to validate the integrity of the token, make sure that it is not expired, extract the claims included in the token and check if the user is authorized to get the requested services based on their role.

If all was successful, the client is granted access to the resource.

Best Practices:

- The secret should be stored safely on the server side, to prevent other parties from generating tokens that will give access to the application.
- An expiration date should be included in the token, to prevent it from being misused and to suspend access to deleted/blocked accounts.

A prototype was created within my individual project to demonstrate the above-described process.

REFERENCES

JSON Web Token. (n.d.). Retrieved from JSON Web Token: https://jwt.io/introduction

The DOT Framework. (2021, July). Retrieved from ICT research methods: https://ictresearchmethods.nl/The_DOT_Framework