AUTOMATING THE RENDERNING PROCESS WITH SECURE REST APIS

S. Ganesh Kumar¹, Ishdeep Singh Chugh², Sarthak Khanduja³

¹Associate Professor, SRM Institute of Science and Technology, Kattankulathur, Chennai, India, ganeshk1@srmist.edu.in
²SRM Institute of Science and Technology, Kattankulathur, Chennai, India, is7316@srmist.edu.in
³SRM Institute of Science and Technology, Kattankulathur, Chennai, India, sm6497@srmist.edu.in

ABSTRACT

Sharing data is crucial in today’s generation. Every minute tons of data are generated. Every bit of data is essential for both the sender and the receiver. Hence, defining a standard set of protocols to protect that data is the need of the hour. Without the need of human interaction, web services can send and receive data acting as a gateway, thus increasing the rate at which data and other essential information is transported over the internet.

This project will be about building a system that securely stores user’s password, and thus, using REST to securely login and deliver the users password from the backend. It will also be automating password regeneration thus helping users always be up to date with their passwords and hence preserve them from losing their precious data.

Keywords: Password Manager, Rijndael algorithm, Security, Amazon Web Services, OpenID Connect, Django, Encryption

I. INTRODUCTION

Every single process in this universe, in today’s generation, is data-driven. From sending a space shuttle to the moon, to analysing and forecasting a product’s sale growth – everything requires data. In many such processes, there comes a need of sharing data from one party to another. At the first sight, it might seem pretty easy because all one needs to do is gather the data stored, and send it to the other party’s address. But what if the stored data, becomes inaccessible or unavailable? There are a ton of possibilities as to why this could happen, ranging from data leak to a third party attack on the storage system. This is where the concept of data security comes into the picture.

Every data stored in this world, needs to have some encryption and storage security in place, to avoid the above-mentioned mishaps. Over the years, there have been many attempts at refining the data security protocols, and this, is another such attempt.

In today’s world, every bit of data is stored under the wall of a passcode/password, only by the knowledge of which, one can access the given data. Hence, it all boils down to remembering the designated password for the given data source. Being such a vast domain, there are thousands of different data sources possible, and it’s unrealistic to expect that a human can remember them all.

This project aims to serve a password manager, with reliable layers of security protocols, enabling people to store their passwords at a safe location. Aiming to achieve maximum security, the algorithm used in developing the password manager is a widely symmetric block cipher algorithm used all around the world. Using the password manager, the user can store multiple passwords, generate and regenerate new passwords all under the scheme of cryptic encryption to avoid password leaks. Not just this, even the password manager admin will not be able to decipher any given user’s stored passwords.
II. LITERATURE SURVEY

A. The CIA triad

Information security can be defined as the safeguarding of data in three areas: confidentiality, integrity, and availability. Unauthorized parties are not permitted to intercept the information.

The term “integrity” refers to the fact that the data has not been compromised or altered during storage or transmission by unauthorized parties.

When attempting to access information, availability ensures that it is accessible to approved parties. The CIA triad is a common name for this model.

Even though the CIA triad does not capture the entire image when it comes to information security, the scientific community agrees that confidentiality, transparency, and availability are critical principles in maintaining information security.

B. Authentication and passwords

Authentication can be divided into four broad categories: anything the user knows, such as a password; anything the user has, such as a smart card; anything the user is, such as a static biometric such as a fingerprint sensor; and anything the user does, such as dynamic biometrics such as sound recognition.

While there are many advanced and reliable authentication methods, such as biometrics and smart cards, the most popular form of authentication is still text-based passwords (often combined with a username) that depend on what the user knows. The human factor is the biggest security problem with this form of authentication since the user is involved in developing and handling the passwords.

C. Managing multiple passwords

Regardless of how robust the information security program is, mismanaging passwords by reusing, writing down, and exchanging them is still a significant part of the daily routine. When users bypass security measures because they are too inconvenient, a rigid security policy runs the risk of backfiring.

In conclusion, multiple password environments face security threats in the following areas:

1. Weak passwords – Using weak passwords, such as commonly used words and phrases.
2. Reusing passwords (using the same or identical password on several systems).
3. Password recovery – lost productivity and the cost of maintaining a password-related service desk
4. Password reminders – all electronic and paper passwords are written down to help you remember them.
5. Password sharing – friends and/or co-workers may share passwords.

If at all possible, avoid using multiple passwords in an organizational environment. When users are required to remember more than four to five passwords, they are more likely to forget them, raising the likelihood of bad password practices.

When users are required to remember more than four to five passwords, they are more likely to forget them, raising the likelihood of poor password management

D. Password managers

One way to minimize the vulnerabilities that occur when multiple passwords are needed is to use password managers. With the support of a single master password that encrypts the entire password file, these software programs can store multiple passwords. These programs can also create secure passwords, reducing the likelihood of password reuse.
III. PROPOSED WORK

Websites around the globe want people to create a strong and unique password for them, but the ordinary people are not security experts. People try their level best to make up for this unrealistic requirement to keep track of passwords, find laptops, write it down on a piece of paper or simplify them so that we don’t forget them. We have become accustomed to this pattern.

Humans are not good at memorising so many passwords, but putting them on a piece of paper or storing them is also very dangerous. For many people who keep track of the unique passwords of their personal accounts, writing down the passwords on paper seems to be a good solution because it solves the problem of using different passwords for each account. However, if the password list is not kept safely there is a high chance all the accounts are compromised and if bank account details are also on the list people can lose their hard earned income also.

Experts generally recommend using a password manager to keep track of all the passwords that a person establishes when creating accounts on various websites. A password manager is usually an application (whether it is a smartphone or a computer) or a website that can be used to securely store all the user's passwords in an encrypted database. After logging into the password manager, users can save and retrieve passwords used on other websites. Users don’t have to remember many different passwords, but only one password: the password to unlock their password manager. The password manager will then remember all other passwords for the user.

The main goal of the password manager is to device a website so that the end client can just open the website, find their desired site, copy its password and paste it on the respective site.

The advantage of having a website is that it is available on each platform rather than being restricted to a specific platform such as iOS or android. You just need a browser on any device and you can access all your passwords securely. You just need to remember one password to retrieve all the remaining passwords. The availability of drag and drop makes it very easy for the customer to login to any site without wasting any time. Also, all fields are protected so that a person standing behind your screen cannot see your passwords while you are trying to copy them. It also features a password generator which will help you generate a password of your desired length and characters let it be alphabets or numerical or special characters and then randomly assigns a password which is highly secure and hence difficult to crack.

IV. IMPLEMENTATION

The backend is implemented on the Django framework. This is the most suitable option as Django is an advanced Python web framework that encourages rapid development and concise and practical design. It is created by skilled developers, and it takes up most of the web development space, so you can focus on writing applications without spending a lot of effort. Also on the plus side it is free and open source. Major advantages of Django include:

- **Incredibly Fast:** Django is designed so as to help developers complete applications as quickly as possible.

- **Super Secure:** Django takes pride in taking security as a major concern and helps developers avoid common security mistakes.

- **Highly scalable:** Some of the busiest websites on the Internet take advantage of the power of Django.

Further to store the passwords in the database we will take advantage of the **Rijndael algorithm**.

The Advanced Encryption Standard (also known as the Rijndael algorithm) is a symmetric block encryption chosen by the US government to protect confidential information. AES is implemented in software and hardware worldwide to encrypt confidential data. This is important for government computer security, network security and electronic data protection.

AES-128 uses a 128-bit key to encrypt and decrypt message blocks, while AES-192 uses a 192-bit key pair to encrypt and decrypt messages, and AES-256 uses a 256-bit key to encrypt and decrypt messages. Use 128-bit, 192-bit and 256-bit encryption keys to decrypt the data in 128-bit blocks. A symmetric encryptor (also called a secret key) uses the same key for encryption and decryption. Therefore, the sender and receiver must know and use the same key. The government divides information into three categories: confidential, confidential or top secret. Keys
of any length can be used to protect confidentiality and secrecy levels. A 192-bit or 256-bit key is required to obtain top secret information. A 128-bit key has 10 rounds, a 192-bit key has 12 rounds, and a 256-bit key has 14 rounds. A round contains several processing steps, including replacement, transposition, and reorganization of the input plain text to convert it into the final ciphertext result.

![Fig. 1. AES Algorithm](image)

The following four steps take part in each and every step of the algorithm:

1. **Replacement of the bytes**: The first step of byte replacement is to replace the bytes which are there in the block text according to the rules specified by the defined S-box.

![Fig. 2. AES Algorithm: Replacement of the bytes](image)

2. **Shifting the rows**: This is a permutation sequence where all rows are shifted by one except the first one.

![Fig. 3. AES Algorithm: Shifting the rows](image)

3. **Mixing the columns**: In this step, the message is jumbled using the Hill Cipher technique rather than mixing the blocks columns.
4. **Attaching the round key**: This is the last step in which the message will be finally XORed along with the specified round key.

The major step is finding a safe way to securely login users on the website. OpenID Connect is the technology chosen to achieve this task. OpenID Connect is basically an identity layer on top of OAuth 2.0 protocol.

It uses REST API call to verify the identity of the end user by following a strict authentication which is performed on a separate authorization server to enhance security. It also provides basic profile details of the end user to the client.

OpenID Connect allows various clients let it be web based or even from a mobile to request and hence provide the required data about the sessions which are authenticated and the information about the end user. Session management is an added bonus to enhance its security.

OIDC extends OAuth authentication capabilities with components such as tokens, which are issued as JSON Web Tokens (JWT). Identity tokens are similar in concept to ID cards because they contain a series of statements about the user, such as their name and email address. It is different from an access token, which does not contain any identifiable information, but instead authorizes access to the resource server. Access tokens do that contain any user information.

```
{
    "access_token": "IEZKr6oP3txZ8Ed",
    "token_type": "bearer",
    "scope": "read:org",
    "expires_in": 3600
}
```

OIDC Authorization follows a code grant flow which incorporates various steps in order to achieve the task. The five steps that take place in the authorization are as follows:
i. The user is prompted to login or sign up via the organizations login page.

ii. Once the user clicks on “Login”, he is passed to the servers authorization API end point with various parameters included in the HTTPS request.

iii. The server on receiving the request will identify whether the information provided is correct and if it is successful in verifying it returns authorization code to the redirect URL.

iv. After receiving the code, the client automatically sends a query to the token endpoint to exchange the code for a token.

v. Now that the client has received the token, it has to make a final API request with access token received in the previous step as the Bearer credential in the HTTPS authorization header. On successful API call, the client receives the resources needed.

The last step in the implementation is deploying the website on the server. Amongst the various options available on demand over the internet, AWS EC2 is the most competitive. Amazon Elastic Compute Cloud (EC2) is part of amazon's cloud computing platform Amazon Web Services (AWS), which enables organizations to develop and deploy applications faster, without needing to invest in hardware upfront. EC2 facilitates scalable application delivery by providing a web service that allows users to run Amazon Machine Image (AMI) to set up virtual machines (called Amazon "instances") containing the required software. Users can start virtual servers, set up security and network, and manage cookies through an intuitive dashboard. With EC2, users can also control the geographic location of an instance to optimize latency and thus provides a high level of redundancy.

V. COMPARISON

HTTPS or Hypertext Transfer Protocol Secure is the used for secure communication over a network. It has become the basic need in today’s world and is a primary protocol for sending information between a browser and a website. HTTPS is encrypted as it increases the security of the information being transferred. In password manager it is an essential requirement since the data being transferred is highly sensitive. As compared to HTTP is uses a TLS/SSL encryption over it.

Basic auth is quick and effective for most APIs, but if you want more protection, you should consider OAuth. OAuth is an open access authorization standard that allows websites or applications to access information on other websites without requiring a password from the user. In response to the direct authentication mode, OAuth was developed.

HTTP simple authentication, which requires the user to enter a username and password, is the most well-known example of this mode. The original form of API authentication for server applications is still used today: basic authentication. Instead of sending username and password to the server with each request, the user sends the API key ID and password.

Before you can use OAuth, you must first fill out a form with your username and password, as well as how you want to access the data (e.g. Gmail account). The password security mode is what it's called. A federated identity for single sign-on was developed in order to construct a better structure for the Web (SSO).
In this case, the end user communicates with their identity provider, who produces a cryptographically signed token and sends it to the application to verify the user's identity. The program signs you in because it trusts the identity provider.

In addition, the password manager built on top of OAuth uses OpenID Connect to improve security. OpenID Connect is a simple authentication layer built on top of OAuth 2.0 that allows the computing client to verify the end user's identity based on the authorization server's authentication and retrieve basic information about the end user through a REST API call. The data format used by OpenID Connect is JSON, which is linked with a RESTful HTTP API.

The password manager uses AES Algorithm for storing the passwords in the database. As compared to RSA, DES, 3DES it has the maximum number of key rounds. Also the key length is highest in comparison to the rest as Key length can be 128-bits, 192-bits, and 256-bits. The block size is also the highest being 128-bits. It uses symmetric block cipher and is faster compared to the rest of the algorithms. In today's encryption technology, AES is widely used and supported by most hardware and software. So far, no practical cryptanalysis attacks against AES have been discovered. In addition, AES has built-in key length flexibility, thus being future-proof as it has the capability to shine against the advancement in machine performing exhaustive key searches.

VI. CONCLUSION

This paper proposed a password managing tool based on Rijndael algorithm. Using Django framework, a successful implementation of the tool within the latest security metrics with enhanced password encryption and regeneration, the tool has significantly reduced the risk of password/sensitive data leaks for the end-user.

In the future, the same algorithm can be enhanced as the key length used in the algorithm is flexible in nature, hence enhancing the security layer even further.

REFERENCES

2. Gerard Wawrzyniak ; Imed El Fray (2020) “New XML Signature Scheme That is Resistant to Some Attacks”
4. Li Li ; Wu Chou ; Wei Zhou ; Min Luo (2016) “Design Patterns and Extensibility of REST API for Networking Applications”
5. Alexander Senf and the EGA Team (2019) “End-to-end Security for Local and Remote Human Genetic Data Applications at the EGA”
10. Shuang Liang ; Yue Zhang ; Bo Li ; Xiaojie Guo ; Chunfu Jia 2Zhel Liu (2018) “Secureweb: Protecting sensitive information through the web browser extension with a security token”
12. Wu-Jeng Li ; Chaming Yen ; You-Sheng Lin ; Shu-Chu Tung ; ShihMiao Huang (2018) “JustIoT Internet of Things based on the Firebase real-time database”
13. Ignacio Bermudez ; Stefano Traverso ; Maurizio Munafo ; Marco Melia (2014) “A Distributed Architecture for the Monitoring of Clouds and CDNs: Applications to Amazon AWS”
15. Xin Xue ; Chunhui Ren ; Yusheng Fu ; Jie Xu ; Jinhong Guo (2019) “SQL injection detection for web applications based on Elastic-Pooling CNN”
20. Daniel Tse ; Kecong Huang ; Bin Cai ; Kaicheng Liang (2018) “Robust Password-keeping System Using Block-chain Technology”