# Information Disclosure Control Data Prevention

**Boobalan S[1], Sakthivel N[2]**

[1]*Student, Department of Master of Computer Applications, Adhiyamaan College of Engineering (ACE), Hosur, Tamil Nadu, India.*

[2]*Assistant Professor, Department of Master of Computer Applications, Adhiyamaan College of Engineering (ACE), Hosur, Tamil Nadu, India.*

Corresponding Author: *boobalanbalu2000@gmail.com*

**Abstract: -** The project was created with the help of python, django, flask, js, HTML, and CSS. Data breaches can have an immediate impact on hundreds of millions or possibly billions of individuals in the data-driven world of today. Data breaches have grown in scope along with the digital transformation as attackers take advantage of our everyday reliance on data. Various measures are taken by many companies to control data breaches in order to prevent them. A business or firm may employ techniques like data encryption, human error, data backup and recovery, and data-security software. In the case of middle-level and low-level organizations suffering the most cyber-attacks, many businesses are successful in defending their data from intruders, typically large multinational corporations which engage a specialist and safeguard their own data. Therefore, we can employ some of the standard data branches prevention strategies in our project to ensure that the domain can receive its user data without any consequences. Our project proposes to improve data-security where user data must reach to their domain without any disruption.

**Key Words:** ***Data transfer, Encryption, Decryption.***

## I. Introduction

Due to the upward moving rising trend in data breaches, founding and operating a profitable business has never been more challenging. In today's data-driven society, data breaches can have an immediate impact on hundreds of millions or perhaps billions of people. As a result of attackers taking advantage of our daily reliance on data, data breaches have expanded in scope along with the digital transformation. Many businesses are successful in defending their data from intruders, typically large multinational corporations which engage a specialist and safeguard their own data. In the case of middle-level and low-level organizations suffering the most cyber-attacks. In order to win the users' trust, we present a suitable proposed system in this project that specifies how we may prevent the data from beginning to end point. The administrator will next create a virtual box for domain users to keep information that is only visible to the IDCP team for those initial domain registration purposes. In order to prevent malware attacks, users initially registered their purposes along with URLs and files. The URL's legitimacy will subsequently be determined by a group of experts. If the file is sound, the next step is to encrypt the data so that only trustworthy people who input the right access ID that admin gave can view the data, Finally, the domain receives the encrypted data, and the domain user also receives the access key to decrypt the data. This will increase security and increase the trustworthiness.

## II. Methodology

- To begin with, you need to implement a user login system that allows users to register and log in to the application.
- When users register, you need to ensure that their passwords are stored securely.
- Admin has provided a verified user to allow a work in data exchanging process in application.
- Once users have logged in, you can create authorization roles that define what actions they are allowed to perform within the application. For example, you might have a role for users who can encrypt data, and another role for users who can decrypt data.

Manuscript revised March 29, 2023; accepted March 30, 2023. Date of publication March 31, 2023.

This paper available online at www.ijprse.com

ISSN (Online): 2582-7898; SJIF: 5.59

- With the user login and authorization system in place, you can now implement the encryption and decryption logic for your application. When a user attempts to encrypt or decrypt data, your application should first check whether they are authorized to perform the action. If they are, the data can be encrypted or decrypted using a secure algorithm like AES or RSA.
- Finally, you need to ensure that the encryption and decryption keys are stored securely. You might store them in a separate database or use a key management service like AWS KMS or HashiCorp Vault.

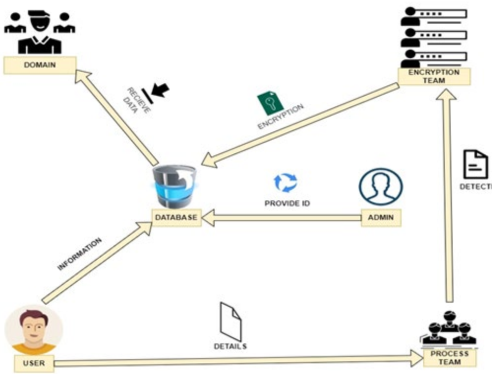

Fig.1. System Architecture

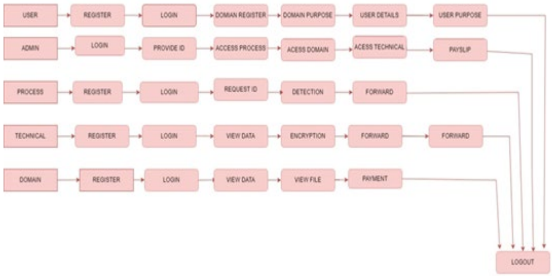

Fig.2. Data Flow Diagram

## III. Recommendations

Identify sensitive information: Determine what information in your project is sensitive and needs to be protected. This could include personal information, and financial data. Use access controls such as authentication, authorization, and role-based access control to limit access to sensitive information to only those who need it. Encrypt sensitive information both in transit and at rest using strong encryption algorithms such as AES or RSA. implement secure coding practices such as input validation, output encoding, and error handling to prevent information leakage through software vulnerabilities. Keep track of who is accessing sensitive information and when, and monitor for unusual activity or unauthorized access. Educate employees on the importance of information security, and provide training on how to handle sensitive information securely. Implement data loss prevention (DLP) solutions: Use DLP tools to monitor and prevent unauthorized access or transmission of sensitive data.

## IV. System Testing And Implementation

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

*System testing*: Before implementing information disclosure control measures, it's essential to thoroughly test the system to ensure that it meets the desired requirements and specifications. This testing may include unit testing, integration testing, and system testing, as well as user acceptance testing.

*Implementation*: Once the system has been tested and validated, it can be implemented in the production.

*Environment*. This involves configuring the system to meet the desired requirements, setting up access controls and authentication mechanisms, and ensuring that all data is properly secured.

*Training and user adoption*: It's important to provide training and support to users to ensure that they can effectively use the system and understand the importance of information disclosure control. This training may include best practices for data handling and security, as well as instructions for using the system itself.

*Ongoing monitoring and maintenance:* Once the system is in place, it's critical to monitor it regularly to ensure that it continues to meet the desired requirements and to identify and address any issues that may arise. This may involve conducting regular security audits, reviewing access logs and usage patterns, and addressing any user-reported issues or concerns.

*Conditional testing*: In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generated on particular condition is traced to uncover any possible errors.

*Data flow testing:* This type of testing selects the path of the program, according to the location of the definition and use of variables. This kind of testing was used only when some local variables were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

## V. Conclusion

In this project, a general study encrypting and how to prevent data breaches also predict the url whether good or bad is conducted how. There are many different ways to prevent the data breaches which some of them are access antimalware agency prevent social engineering, software update regularly, which were ineffective in most cases for so we implement the random encryption methods and predict the url good or bad before access it and provide distribution key for perspective users will enhance the security purpose and prevent the data breaches.

## References

[1]. Ross Anderson (1993): Ross Anderson is a computer scientist at the University of Cambridge, and his research focuses on security engineering and the economics of information security.

[2]. Bruce Schneier: Bruce Schneier (1995) is a security technologist and author of several books on information security, including "Applied Cryptography" and "Secrets and Lies.

[3]. Whitfield Diffie: Whitfield Diffie is a mathematician and cryptographer who co-invented public-key cryptography (1976).

[4]. Ron Rivest, Adi Shamir, and Leonard Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." 1978.

[5]. Gene Spafford - "The Internet Worm Program: An Analysis" (1988), "Computer Viruses as Artificial Life" (1991). Gene Spafford is a computer scientist at Purdue University and a leading expert on computer security and information disclosure control.

[6]. Michael Howard - Writing Secure Code (2002). Michael Howard is a British software security expert who works for Microsoft.

[7]. Kevin D. Mitnick - The Art of Deception: Controlling the Human Element of Security (2002). Kevin Mitnick is a former computer hacker who now works as a security consultant.

[8]. Gary McGraw - Software Security: Building Security In (2006). Gary McGraw is an American computer scientist and author who specializes in software security.

[9]. Harold F. Tipton - Information Security Management Handbook, Sixth Edition, Volume 2 (2007).

[10]. M. L. McCallister and E. J. Immerman. "A methodology for preserving confidentiality in data base management systems.

[11]. A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. "l-diversity: privacy beyond k-anonymity." Proceedings of the 22nd International Conference on Data Engineering, 2006.

[12]. D. Naor and M. Yung. "Public-key cryptosystems provably secure against chosen ciphertext attacks." Proceedings of the 22nd Annual Symposium on Theory of Computing, 1990.