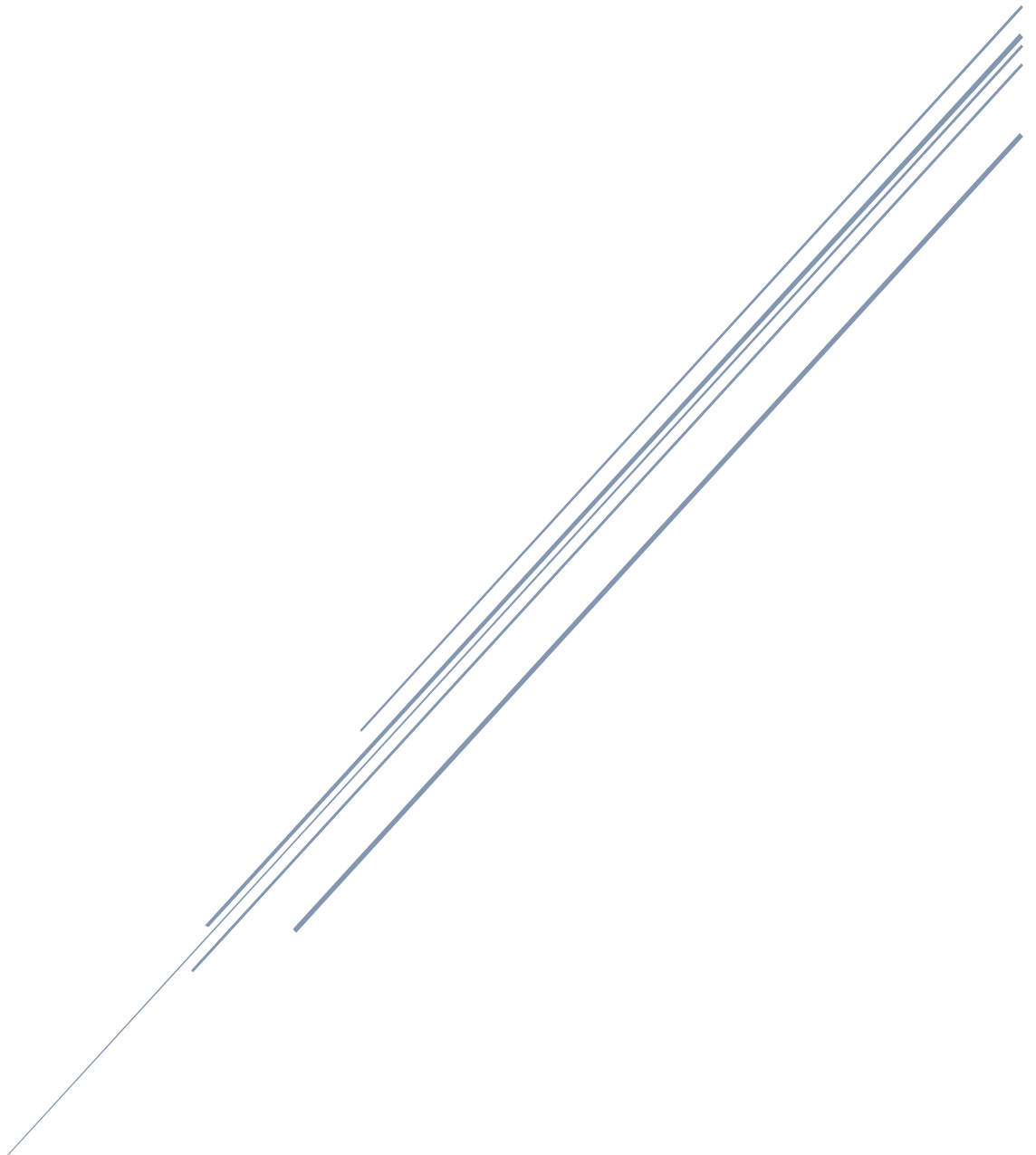


# RESEARCH DOCUMENT

## Sensitive Data Exposure(SDE)

*Joshua dos Santos Oliveira Mota*



Fontys Hogeschool ICT  
B HBO-ICT – Demand based

## Table of Contents

Sensitive Data Exposure .....	2
What is SDE?.....	2
The cause of SDE.....	2
How have I prevented SDE in my project? .....	2
Conclusion.....	2

# Sensitive Data Exposure

## What is SDE?

Instead of stealing crypto from sites, SDE attackers steal secret keys like passwords, execute man-in-the-middle attacks, or steal clear text at off the server. These attackers are required to do their attacks manually generally.

## The cause of SDE

The main cause of sensitive data exposure is simply not encrypting sensitive data. I am talking about data like passwords or addresses or when crypto is employed, a weak key generation and management. Its not only a threat if the data is not encrypted, because SDE is also a threat when data is encrypted weakly like a weak password hash.

## How can you prevent SDE?

You can do the following things to prevent SDE:

- Classify the data that is processed, stored or transmitted by the application and identify which data is sensitive according to privacy laws.
- Don't store sensitive data unnecessarily. Discard it or encrypt it.
- Ensure up-to-date and strong standard algorithms, protocols, and keys are in place.
- Disable caching for response that contain sensitive data.
- Store passwords using strong adaptive and salted hashing functions.
- Verify independently the effectiveness of configuration and settings.

Like you see there are some ways to prevent Sensitive Data Exposure.

## How have I prevented SDE in my project?

First, I have used MD5 hashing for my password hashing. MD5 is a commonly used hash function that was used for cryptographic data. I started with just generating a random string but found out soon enough that this wasn't the solution. I asked my fellow students on what to use and they said to use MD5, because it was easy to implement.

I also am using JSON web tokens to get to use a user's data but not knowing what is inside of them. JWT's are used to create data with optional signature and/or optional encryption which holds several claims. Usually, JWT's are signed with a private key. I have used these to hold the user that is logged in and set the token in the session storage. With this token I can get the data from the user that is not encrypted, else I am getting the encrypted data.

## Conclusion

My users don't have to enter a lot of sensitive data but the data that they've entered is fully secured to prevent SDE from happening. But during this research I've found out that SDE is still happening a lot on the internet without anyone knowing.