CS-405 Secure Coding – Case Study

Eric Slutz

Southern New Hampshire University

Case Study

Introduction

The Aadhaar data breach (https://www.csoonline.com/article/534628/the-biggest-data-breaches-of-the-21st-century.html), occurred in January 2018. This breach made the news both due to the size of the breach and the amount of the personally identifiable information (pii) and personally identifiable financial information (pifi) that was taken.

Breach description

The breach of Aadhaar contained the information of more that 1.1 billion Indian citizens from a database created in 2009 by the Unique Identification Authority of India (UIDAI). This information included names, addresses, photos, phone numbers, emails, biometric data such as fingerprints and iris scans, and bank account information. The Unique Identification Authority of India originally denied that any financial information was stored in the database (Swinhoe, 2023). This data was likely targeted due to the amount of extremely personal information that was stored in it.

Threat identification

With the type of information exposed about these Indian citizens there is a wide range of serious threats they could immediately face. This includes identity theft, financial loss due to the exposing banking information, phishing attacks, extortion or ransom with a promise to not expose the data if paid, stalking of harassment, social engineering attacks, and access to systems or records that authenticate with the user's biometric data. If the vulnerability goes unresolved, the potential threat is that this data breach continues and exposes the information of more people.

What could a developer have done to prevent this breach?

The developers could have implemented literally anything to secure the API that resulted in the breach, and it would have been an improvement. The API didn't have any type of access control or policy in place, providing the data to anyone who accessed this open API.

Explain the role of best practices, Triple A and DiD in preventing future attacks

By following best practices and using Triple A (authentication, authorization, and accounting) and defense in depth, the chances of a successful future attack like this can be reduced. The first step in this case is that the API should be asking who is attempting to access it (authentication). Once it's known who's trying to access the API, it should then be checked if they are allowed to access it (authorization). The information of who is accessing the API and if the access is granted or not, should be retained for potential future review (accounting). For authentication and authorization, a basic username or password could work, but what is more common with API access for this purpose is the use of API tokens. Accounting can be accomplished by logging all access attempts. Adding Triple A is a good first step, but it's important to have layer of security (defense in depth). Another layer that should be added is to encrypt all data. That way if the data is exposed, it would be unreadable due to the encryption.

References

Swinhoe, M. H. a. D. (2023). The 15 biggest data breaches of the 21st century. *CSO Online*. https://www.csoonline.com/article/534628/the-biggest-data-breaches-of-the-21st-century.html