Designing and Implementing a Scalable Data Loss Prevention System: A Full-Stack Approach

CS4991 Capstone Report, 2024

Natalie Yee
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
nvn5yn@viginia.edu

ABSTRACT
Capital One needed to address the challenge of safeguarding sensitive data across a large enterprise, specifically focusing on Data Loss Prevention (DLP). To protect sensitive company data, my team and I developed a scalable, serverless frontend and backend using Amazon Web Services (AWS) Lambda, S3, and DynamoDB. The solution involved the design and implementation of RESTful APIs for violation management, improving data retrieval and security through backend integration with DynamoDB and Single Sign-On (SSO) for secure access. The deployed system successfully improved user access and security for over 50,000 employees. Future work includes expanding the platform's capabilities to allow users to remediate violations and submit suppressions for sensitive data.

1. INTRODUCTION

Data Loss Prevention (DLP) is a critical concern for organizations like financial institutions that handle large amounts of sensitive information, such as financial institutions. With the growing threats of data breaches, corporations need to implement measures to safeguard sensitive data from unauthorized access, leakage, or loss. DLP solutions help companies ensure compliance with regulations as well as protect sensitive business and customer information.

Capital One, a major financial insitution, faced the challenge of securing large amounts of data distributed across its global enterprise. The company needed a scalable solution that was capable of monitoring, identifying, and preventing potential data loss. Existing solutions for protecting sensitive data needed to adapt to the growing digital environment.

To address these challenges, my team and I developed a scalable, serverless solution using Amazon Web Services (AWS). By leveraging AWS Lambda, S3, and DynamoDB, we created an end-to-end system to improve Data Loss Prevention (DLP). This system integrates with Single Sign-On (SSO) for secure access and enables efficient management of violation data.

2. RELATED WORKS

Data leakage is a critical threat for large enterprises. According to IBM's 2016 Cost of Data Breach Study, the average cost of a data breach was $4 million, with incidents such as the 2013 Target Corporation breach and the 2014 Yahoo breach resulting in millions of dollars in losses (Cheng et al., 2017). The need for Data Loss Prevention systems is evident with the rise in internal and external data leak incidents.

Traditional DLP systems use techniques like traffic inspection and the enforcement of data use policies to protect sensitive information

from being leaked. However, as the amount of data being collected continues to grow, these systems face challenges in effectively monitoring, identifying, and preventing data loss across an enterprise. For instance, communication channels such as cloud file sharing and instant messaging have increased the area for data leaks (Cheng et al., 2017).

Serverless computing allows organizations to reduce operational overhead by deploying functions without needing to manage servers. As detailed by Rajan (2018), serverless architectures like Function as a Service (FaaS) enable the efficient scaling of resources. In particular, AWS Lambda supports cost-effective scaling which is crucial for handling large amounts of data. Our system at Capital One utilizes AWS Lambda to minimize infrastructure management and optimize resource use, addressing some of the challenges DLP systems face as identified in previous research.

3. PROJECT DESIGN [or PROPOSAL DESIGN or appropriate section title]

The design and implementation process of the Data Loss Prevention system known as the Everglade Portal, was guided by the need to create a cost-effective, scalable, and maintainable system. Below are the key decisions my team made, and the solutions implemented to meet these goals.

3.1 Front End Design

Initially, we considered AWS Fargate for the front-end due to its ability to handle dynamic content. However, after further research, we switched to AWS S3 because it offers a simpler and more cost-effective solution for front-end hosting. The main difference between AWS Fargate and AWS S3 is that AWS Fargate is constantly running while AWS S3 only runs when it needs to. Since AWS Fargate is always running it costs a lot more to maintain than AWS S3. My team found that the Everglade Portal did not need to run constantly, it only needed to be running while a user was on the site.

3.2 Back-End Transition: From Faragte to Lambda

The back-end, initially hosted on AWS Fargate, was reconsidered for its cost and complexity. My team decided to transition the back-end services to AWS Lambda due to reduced costs and simplified management. Fargate is ideal for applications that need continuous availability. Since the API traffic was sporadic, with users primarily accessing the portal after receiving notifications, hosting the back-end on Fargate resulted in unnecessary resource usage and higher costs. My team wanted to use Lambda because it is serverless and event-driven, making it ideal for our API, which does not experience consistent traffic. This can lower operational costs by eliminating excess resource usage, while maintaining scalability and flexibility. Transitioning the back-end from Fargate to Lambda involved reconfiguring API endpoints and integrating AWS API Gateway to route incoming requests to the appropriate Lambda functions.

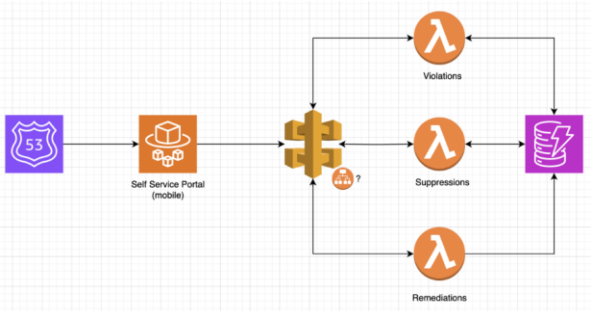

Fig. 1 Architecture Design

3.3 API Design and Endpoints

Our API manages data related to violations, remediations, and suppressions. To ensure flexibility and maintainability, we created separate Lambda functions for each category of data, allowing easier updates and future scalability.

Endpoint Overview

Violations: GET/, GET/:id, POST/:id/action/remediate

Remediations: GET/:id, GET/user/:user, POST/

Suppressions: GET/, GET/user/:eid, POST/:id

This structure ensures that each function remains easy to modify and since they are separated, multiple teams can be working on different Lambdas at the same time.

3.4 Technology Stack

Our team chose to write the codebase in Typescript for consistency and collaboration. The existing application that the Everglade Portal was based on was written in Typescript, making it easier to collaborate with team members who had worked on the original system.

4. ANTICIPATED RESULTS

My team's design choices have several anticipated benefits. One of the main benefits is cost efficiency. Transitioning the front-end hosting to AWS S3 and the back-end services to AWS Lambda significantly reduces operational costs. S3 is highly cost-effective for serving static content, and the costs associated with Lambdas only scale with usage.

The architecture is simplified as a result of the design choices made by my team. The decision to use serverless technologies, such as AWS S3 and Lambda simplifies the system's architecture. These serverless technologies simplify the system's architecture by eliminating the need to manage underlying infrastructure, such as servers and load balancers. Additionally, using serverless technologies helped make the architecture more flexible by allowing for easier updates, faster deployment cycles, and the ability to scale components independently.

5. CONCLUSION

The serverless, scalable Data Loss Prevention (DLP) system, my team and I developed at Capital One, effectively addressed the need for robust data protection across a large enterprise. By leveraging AWS components such as Lambda, S3, and DynamoDB, we ensured that violation management was efficient, data retrieval was streamlined, and security was enhanced. Our integration of Single Sign-On (SSO) provided secure user access and ensured compliance with security standards. This system enabled employees to easily access and view their data violations on mobile devices, significantly improving user experience and operational efficiency. The knowledge gained from this project included insights into optimizing serverless architectures, cost management, and balancing security with usability. This project's meaningful impact included simplifying data protection efforts and reducing potential data exposure.

6. FUTURE WORK

There are still opportunities for future work to expand the platform's functionality and address additional business needs. One area for future work involves extending the system's capabilities to include remediation workflows. Allowing users to take corrective actions within the platform, such as remediating violations and submitting data suppression requests, would improve efficiency of violation management.

Additionally, future efforts could focus on integrating the DLP platform with other enterprise data management tools to provide a more comprehensive security ecosystem. This would allow for a seamless flow of information across different business units and promote better collaboration while maintaining data protection.

7. ACKNOWLEDGMENTS

I would like to thank my team members: Rebecca Peng, Sophia Hubscher, and Lei Hanna. As well as my manager Logan Miles.

REFERENCES

Cheng, L., Liu, F., & Yao, D. (Daphne). (2017). Enterprise data breach: Causes, challenges, prevention, and future directions. *WIREs Data Mining and Knowledge Discovery*, *7*(5), e1211. https://doi.org/10.1002/widm.1211

Rajan, R. A. P. (2018). Serverless architecture—A revolution in cloud computing. *2018 Tenth International Conference on Advanced Computing (ICoAC)*, 88–93. https://doi.org/10.1109/ICoAC44903.2018.8939081