Vulnerability Management Hub

Group 6: Julia Nam, Lawrence Snow, Hari Kishan Reddy Abbasani, Dwireph Parmar, Chris Min

Problem Statement

Organizations often use multiple security tools like SentinelOne for endpoint protection, Nessus for vulnerability scanning, and Osquery for system monitoring. While each tool has its unique purpose, vulnerability data ends up spread out in multiple places and formats. Without the ability to analyze all the data in one place, it's difficult to understand the full risks and their contextual risk severity. This fragmentation makes it difficult to mitigate risks effectively, leading to incompliant assets and countless vulnerabilities in the system.

Initial Proposal/Design

A centralized dashboard platform can consolidate all data into a single, easy-to-understand interface, allowing for quicker identification of vulnerabilities and improving security efforts. By leveraging automated data ingestion from S3 bucket (data from security tools) to RDS using AWS Lambda, new vulnerability data is processed in real time via Quicksight, reducing manual effort and human error. This accelerates the availability of critical security data and enhances the overall efficiency and responsiveness of security operations. We will leverage AI to generate mock data for each security platform to provide us all the details of the found vulnerabilities.

Issues Encountered

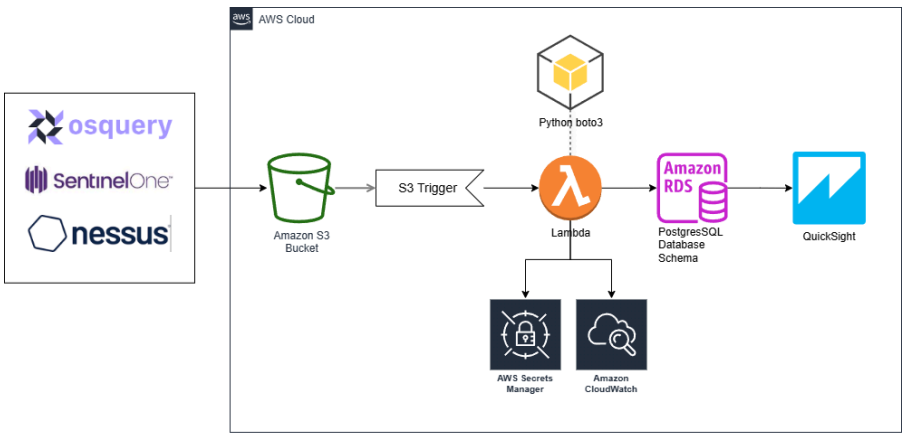
Event Bridge - The EventBridge did not trigger the Lambda function as expected. We learned we had to capture S3 events through AWS CloudTrail and set up EventBridge target/rule [1]. However, we found a better native solution that S3 has a built-in trigger for Lambda functions. This built-in trigger was easier to implement and was also a more intuitive solution [2]. By switching to the S3 trigger, we were able to make sure that new vulnerability data is processed in real time in a reliable manner.

Lambda - Even though Lambda and Secret Manager are native AWS tools, Lambda in a private subnet required a VPC interface endpoint to reach out to Secret Manager [3]. Another solution is that if Lambda is deployed in a private subnet, the NATgateway should be deployed in a public subnet and have the private subnet route to it for outbound traffic as it requires external access.

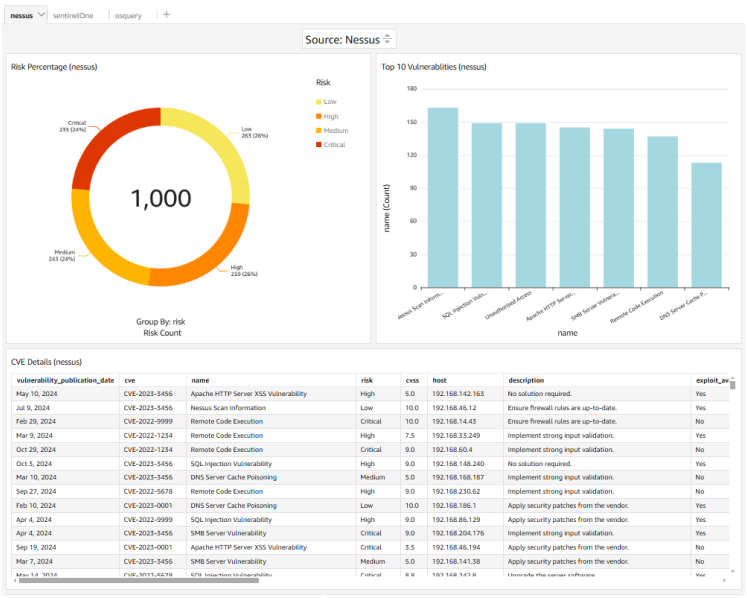
RDS Database Tables - When lambda is run to insert data into the RDS PostgreSQL database, we encountered an issue of the field type not matching with the payload. The type of python defined value should match the field type in the table schema. We had converted a few table fields to match the values we wanted to represent in Quicksight.

Final Design

Our final design solution to create a centralized vulnerability hub consists of ingesting the data via S3 bucket. With a S3 event trigger for when a file is uploaded to the bucket, a



lambda function is executed to read the csv file and insert each data into our RDS PostgreSQL database. Each table in the database consists of all the fields the tool provides as artifacts. Quickisght is connected to the database using VPC endpoint and directly queries our tables [4]. Cloudtrail logs exist for all the services for any troubleshooting and error deep dives. And for this project, our mock data was generated by OpenAI's ChatGPT.



We provide a detailed visual summary of vulnerabilities that exist in the environment from the perspective of three different threat detection engines which have been centralized in QuickSight for convenient access. The dashboards provide a simple and uniform view of severity levels, top vulnerabilities, and affected asset details. The Nessus dashboard provides information about vulnerability CVEs affecting assets and remediation information. The SentinelOne dashboard provides information about assets with threats and affected files. The Osquery dashboard provides information about risky commands and queries being run. With our detailed and centralized view, we can easily determine what assets are most critical and immediately prioritize remediation efforts from the displayed findings.

Lessons Learned

Teamwork played a crucial role in our success. We learned how to collaborate effectively, which allowed us to leverage each team member's strengths, leading to innovative solutions and a more robust final product. Regular communication and clear division of responsibilities ensured that everyone was aligned and working towards common goals. We also learned the value of flexibility and adaptability, as we had to pivot our strategies and tools based on real-time feedback and challenges. Security was another critical focus; implementing IAM roles and policies with the principle of least privilege reinforced our system's security posture and taught us the value of zero trust architecture in security. Overall, these experiences taught us the value of flexibility, automation, robust security practices, and strong teamwork in developing effective and resilient security solutions.

Future Work

If we were to approach the project differently, we would utilize Terraform for infrastructure as code, allowing us to easily take down and redeploy the project while ensuring our work is saved and configurations remain consistent. Additionally, we would create comprehensive documentation for each component and process, ensuring clarity and ease of maintenance for future developers. And instead of using S3 bucket to initially store the raw data, we want to call each platform's API endpoints and retrieve the data directly.

References

[1] AWS Solutions, "Using EventBridge - Amazon Simple Storage Service," 2024, last accessed December 17, 2024. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/EventBridge.html

[2] AWS Solutions, "Using AWS Lambda with Amazon S3," 2024, last accessed December 17, 2024. [Online]. Available: https://docs.aws.amazon.com/lambda/latest/dg/with-s3-example.html

[3] AWS Solutions, "VPC Endpoint Overview - AWS Secrets Manager," 2024, last accessed December 17, 2024. [Online]. Available: https://docs.aws.amazon.com/secretsmanager/latest/userguide/vpc-endpoint-overview.html

[4] AWS Solutions, "Connecting to a VPC with Amazon QuickSight," 2024, last accessed December 17, 2024. [Online]. Available: https://docs.aws.amazon.com/quicksight/latest/user/working-with-aws-vpc.html