Cyber Security (BCSE410L)

DIGITAL ASSIGNMENT PROJECT REPORT

Fall Semester 2024-2025

Submitted by

Name: SURIYA KUMAR M REGNUM: 21BCE0630

in partial fulfillment for the award of the degree of

B. Tech

in

Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Vellore-632014, Tamil Nadu, India

School of Computer Science and Engineering

November 3, 2024

INTRODUCTION OF THE TOPIC

As cloud computing becomes the backbone of modern business operations, organizations across industries are rapidly adopting cloud platforms to benefit from their inherent flexibility, scalability, and cost-effectiveness. Cloud platforms allow companies to scale resources up or down based on demand, optimize IT costs, and accelerate innovation by deploying applications and services faster than traditional, on-premise environments. However, this shift to cloud-based infrastructure also introduces unique security challenges that traditional cybersecurity measures fail to address fully. Cloud environments operate in a shared, often multi-tenant infrastructure, which creates an expanded attack surface and demands specialized security measures. One approach that has emerged to address these challenges is Cloud Security Posture Management (CSPM).

Cloud Security Posture Management (CSPM) is a rapidly growing field within cybersecurity, specifically designed to manage and improve the security posture of an organization's cloud infrastructure. As organizations increasingly rely on cloud-based resources, CSPM plays a crucial role in maintaining strong security practices. By continuously monitoring cloud environments for misconfigurations, compliance violations, and vulnerabilities, CSPM provides an automated, proactive approach to cloud security. Unlike traditional security tools, which are often geared towards endpoint or network protection, CSPM is purpose-built to address cloud-specific issues. CSPM tools offer organizations a framework to identify, track, and remediate security risks in real time, ensuring visibility and control over cloud resources.

One of the primary motivations behind CSPM is the high risk of misconfigurations in cloud environments. As cloud environments are highly dynamic and resources are frequently created, modified, or removed, misconfigurations are a common risk. Misconfigurations can occur when sensitive data, such as databases or storage buckets, is inadvertently exposed to the public internet or when access control policies are too permissive. These misconfigurations are among the most common causes of cloud data breaches, as they expose assets to potential unauthorized access and exploitation. CSPM solutions address these issues by continuously scanning cloud configurations, identifying misconfigurations, and enabling security teams to enforce security policies that align with industry best practices. This proactive detection and remediation of security misconfigurations significantly reduce the likelihood of

data breaches and unauthorized access incidents, helping organizations to secure their cloud resources effectively.

Another key driver for CSPM is the unique **shared responsibility model** that governs cloud security. In this model, cloud service providers (CSPs) are responsible for securing the infrastructure that underpins cloud platforms, while customers retain responsibility for securing their data, applications, and configurations. This model places an obligation on customers to actively manage the security of their own assets within the cloud environment. CSPM tools equip organizations with the necessary capabilities to uphold this responsibility by continuously monitoring cloud configurations, tracking compliance, and managing risks. This capability is essential in today's cloud environments, where responsibility for security cannot rest solely on the cloud provider. Instead, CSPM ensures that both providers and customers fulfill their security obligations in a balanced, cooperative manner, reinforcing the overall security of the cloud ecosystem.

CSPM solutions also play a critical role in regulatory compliance. Many organizations are subject to compliance requirements from frameworks like GDPR, HIPAA, and PCI DSS, which mandate specific security controls to protect sensitive data. Ensuring compliance in cloud environments can be complex, as organizations may operate across multiple regions and use diverse cloud services. CSPM solutions simplify this process by automatically checking cloud configurations against compliance standards and generating reports that detail areas of non-compliance. These automated compliance checks not only help organizations meet regulatory requirements but also reduce the risk of penalties and reputational damage associated with data breaches.

Our project, titled **"CSPM Risk Scorer Proof of Concept (PoC)"**, is a demonstration of how foundational CSPM capabilities can be implemented to provide meaningful insights into cloud security risks. This project aims to create a risk assessment tool that scores cloud assets based on their risk level, using a heuristic scoring approach combined with the MITRE ATT&CK framework. By leveraging ATT&CK, a globally recognized framework that categorizes attack techniques, the tool can map identified vulnerabilities to specific tactics and techniques commonly used by attackers. This alignment enables security teams to gain a deeper understanding of the threat landscape and assess how each vulnerability may be exploited. By mapping vulnerabilities to real-world attack techniques, our CSPM tool not only identifies potential risks but also

provides actionable intelligence, allowing security teams to prioritize mitigation efforts based on the severity, impact, and likelihood of exploitation.

The scoring methodology of our CSPM Risk Scorer considers multiple factors, including the type of vulnerability, its exposure level, and its alignment with known attacker tactics from the ATT&CK framework. This enables a tailored approach to risk assessment, where vulnerabilities are categorized as high, medium, or low-risk, providing a clear hierarchy of risks for security teams to address. High-risk vulnerabilities, such as those involving exposed databases or unencrypted sensitive data, are prioritized for immediate remediation, while lower-risk issues can be monitored or addressed as part of routine maintenance. This risk-based approach ensures that resources are allocated effectively, focusing efforts on vulnerabilities that pose the greatest threat to the security posture of the cloud environment.

Furthermore, the integration of ATT&CK into our CSPM PoC project provides a comprehensive view of each asset's security posture. ATT&CK serves as a powerful knowledge base of tactics, techniques, and procedures (TTPs) used by cyber adversaries, allowing security practitioners to understand the motivations, methods, and patterns of attackers. By aligning our scoring methodology with ATT&CK, our CSPM tool enables organizations to take a proactive stance against potential threats, as they can anticipate likely attack vectors and take preemptive measures to secure vulnerable assets.

In conclusion, as organizations continue to migrate to cloud environments, the need for specialized security measures like CSPM becomes increasingly critical. CSPM not only helps manage the complexity of cloud security but also provides an essential layer of protection against the growing number of threats targeting cloud resources. By offering continuous monitoring, risk assessment, and automated remediation capabilities, CSPM empowers organizations to maintain a strong security posture and mitigate risks associated with cloud infrastructure. Our CSPM Risk Scorer PoC serves as a practical demonstration of how CSPM principles can be applied to evaluate cloud security risks, providing insights that are crucial for strengthening defenses and protecting valuable cloud-based assets.

DETAILED DESCRIPTION OF THE TOPIC

CSPM is a relatively new but rapidly evolving field within cybersecurity, addressing the specific needs of cloud security. Unlike traditional security tools, which may focus on network or endpoint security, CSPM is designed to address cloud-specific risks and challenges, such as configuration drift, compliance management, and visibility into multi-cloud environments.

One of the key functions of CSPM is automated misconfiguration detection. Cloud environments, by nature, are highly dynamic; new resources are frequently added, and existing configurations are modified. This dynamism makes it difficult for security teams to manually keep track of all changes. CSPM tools automate the process of detecting misconfigurations, providing real-time alerts whenever configurations deviate from established policies. For instance, if an organization’s policy mandates that all storage buckets must be private, a CSPM tool would immediately alert the team if a bucket is accidentally exposed to the public.

CSPM tools also play a critical role in compliance management. With many organizations subject to regulatory standards like GDPR, HIPAA, and PCI DSS, maintaining compliance can be a complex and resource-intensive process. CSPM solutions simplify this by automatically checking cloud configurations against compliance requirements and generating reports that detail areas of non-compliance. This functionality helps organizations avoid costly penalties and ensures that they remain compliant with industry standards and regulatory requirements.

Our project implements a heuristic risk scoring system within a CSPM framework. The scoring model assesses each cloud asset's security posture based on factors such as exposure to known vulnerabilities, access configurations, and encryption status. A critical component of our scoring model is its integration with the MITRE ATT&CK framework. MITRE ATT&CK provides a structured representation of the tactics, techniques, and procedures (TTPs) used by adversaries, making it easier to categorize risks based on real-world threat intelligence.

For example, a misconfigured API endpoint may be mapped to an "Initial Access" tactic in ATT&CK, indicating that it could be exploited by attackers to gain unauthorized access to the environment. By tagging cloud assets based on ATT&CK techniques, our tool offers a deeper level of insight into each

vulnerability's potential impact and provides actionable recommendations for remediation. This approach not only enhances the detection capabilities of CSPM but also enables more strategic, intelligence-driven risk management.

The risk scoring methodology we developed for this project categorizes vulnerabilities into high, medium, and low-risk tags, based on a combination of heuristic analysis and ATT&CK mappings. This categorization allows security teams to focus their efforts on the most critical vulnerabilities, rather than being overwhelmed by an extensive list of issues with varying levels of severity. The integration of ATT&CK also facilitates a better understanding of how each vulnerability aligns with attacker tactics, enhancing the organization's ability to anticipate and defend against potential threats.

DESCRIPTION OF THE TOOLS USED IN THE PROJECT

Our CSPM Risk Scorer PoC project was developed using Python, leveraging its robust libraries for data processing, and implemented in Visual Studio Code (VS Code). The choice of Python was influenced by its simplicity, versatility, and extensive support for libraries that facilitate tasks such as data handling, automation, and integration with external frameworks. This made Python a natural fit for developing a risk scoring mechanism that aligns with CSPM objectives.

VS Code served as the primary Integrated Development Environment (IDE) for this project. Known for its lightweight and highly customizable interface, VS Code provides various features that enhance productivity, including syntax highlighting, debugging capabilities, and integrated terminal support. Additionally, the IDE's built-in Git integration allowed us to manage version control efficiently, enabling collaboration, tracking changes, and rolling back versions when necessary.

The Python libraries used in this project played a critical role in implementing CSPM functionalities. For instance, JSON parsing libraries were essential in handling the MITRE ATT&CK mappings, which are represented as structured data files. By loading and parsing these files, we were able to map cloud assets to relevant ATT&CK techniques and tactics, creating a risk profile for each asset that aligns with real-world attacker behaviours.

To simulate cloud assets and configurations, we generated synthetic data through custom scripts. This allowed us to create a realistic dataset that represents various cloud assets with different configurations and vulnerabilities. By processing this data through our CSPM tool, we were able to assess the risk associated with each asset based on predefined heuristic criteria and ATT&CK mappings. This approach demonstrated how the tool could identify, categorize, and prioritize vulnerabilities even in the absence of actual cloud infrastructure.

Although CSPM tools are typically integrated with platforms like AWS or Azure for direct scanning of cloud environments, our project was designed as a standalone solution, implemented entirely within VS Code. This simplified setup focused on the core functionalities of CSPM, such as misconfiguration detection, heuristic scoring, and ATT&CK-based vulnerability mapping, without relying on cloud-specific APIs or services. This flexibility ensures that our tool can be adapted to various cloud environments, making it a versatile solution for organizations that require a customizable CSPM tool.

The MITRE ATT&CK framework was instrumental in developing our heuristic risk scoring model. ATT&CK is a well-established taxonomy that categorizes attack techniques based on real-world adversarial behaviours, allowing security teams to understand and anticipate attacker strategies. By mapping each vulnerability to ATT&CK tactics, our tool provides a contextual risk score for each asset, highlighting vulnerabilities that align with known attack vectors. This integration with ATT&CK enhances the accuracy and relevance of our scoring model, enabling more strategic decision-making in cloud security.

STEP BY STEP IMPLEMENTATION

Our code is designed to analyze vulnerabilities for cloud assets, calculating a comprehensive risk score for each asset based on their environment, security, compliance requirements, and other factors. It uses data from two external JSON files, which contain details about various risk types and asset records. It has been implemented in the following manner:

1. Data Loading and Preparation: The code reads two datasets, one containing descriptions of risk categories and their scores, and another with information

about individual assets. The risk descriptions include each risk type's name, score, and associated comments.

2. Risk Score Mapping: The script creates a mapping of risk names to their respective scores and comments. This mapping allows the program to quickly retrieve the risk score and relevant details for each risk associated with an asset.
3. Likelihood Calculation: We define a method to assess how likely each risk is to affect an asset. It evaluates the severity of the risks and sums up their values to generate a "likelihood score" for each asset, factoring in higher scores for more critical risks. If an asset has multiple high-risk factors, the score is adjusted to reflect a higher probability of issues.

4. Impact Calculation: Next, the program evaluates the potential impact of a risk on each asset based on factors like the asset's environment (e.g., development or production), regulatory compliance (e.g., GDPR, HIPAA), data classification, and security level. Each of these factors contributes to an "impact score." Assets with more sensitive data or higher security requirements receive a higher score.

5. Summary Generation: For each asset, the code generates a summary that explains its risk profile. This summary includes the calculated likelihood and impact scores and references key factors like environment, compliance, and data sensitivity, along with specific reasons tied to the risks involved.

6. Risk Scoring and Record Updating: The main section of the code iterates through each asset, applying the likelihood and impact calculations. It then combines these scores into an "overall risk score," which gives a broad sense of each asset's risk level. Each asset record is updated with these scores and the generated summary.

7. Top Asset Identification: After scoring all assets, we identify the top 10 and top 50 highest-risk assets. These subsets are then analyzed separately, providing insight into the riskiest assets within the dataset.

8. Risk Comparison by Environment: The code then structures the data to enable a comparison of risks across different environments (e.g., production vs. testing), summarizing the average risk score for each environment and capability group. This information is organized in a way that reveals which combinations of environment and asset type carry the highest risk.

9. Risk Visualization: Finally, the program creates a visual representation, focusing on the top 50 riskiest assets. This visualization highlights patterns in risk across different environments, showing where the greatest risks lie.

10. Interactive Dashboard: The code concludes by displaying an interactive summary of the top risks. This setup enables users to view the riskiest assets, explore detailed risk summaries, and examine the risk distribution by environment.

OVERVIEW OF THE TOOLS AND LIBRARIES USED

1. JSON: This library is essential for loading and saving data in JSON (JavaScript Object Notation) format. JSON is commonly used for data interchange, particularly in web applications and APIs, because of its readability and compatibility with various programming environments. Here, it loads risk and asset data from JSON files and saves updated data back to a new JSON file after analysis.

2. Hashlib: The hashlib library provides a set of hash functions, enabling secure, unique hash generation. In this script, it's used to create a unique hash suffix for the output file name by applying the MD5 hash function to random data, ensuring that each generated filename is distinct. This is especially useful when running the script multiple times to avoid overwriting files.

3. OS: The os module provides functions for interacting with the operating system, such as file handling and generating random bytes. Here, it generates random bytes for the filename hash suffix, enhancing file organization and security by adding randomness to filenames.

4. Pandas: Pandas is a data analysis library widely used for handling structured data. In this code, it transforms the JSON data into a DataFrame, which makes it easy to manipulate, filter, and analyze. Pandas also enables efficient data operations like sorting, grouping, and pivoting, which are used here to identify and analyze the top 10 and top 50 riskiest assets and to create summary statistics for visualization.

5. Plotly: Plotly is a powerful library for creating interactive visualizations. Unlike static plots, interactive visualizations allow users to engage with the data by hovering, zooming, and panning. In this code, Plotly's graph objects module is used to create an interactive heatmap that represents risk scores across different environments, making it easy to explore detailed information dynamically.

6. Streamlit: Streamlit is an innovative open-source Python library designed for creating interactive web applications focused on data science and machine learning projects. It enables data scientists and developers to turn their data scripts into fully functional, shareable web apps with minimal effort and without the need for web development experience. With its straightforward API, Streamlit allows for the rapid creation of interactive dashboards and visualization tools by offering built-in widgets like sliders, buttons, and select boxes to capture user inputs and update visualizations dynamically. Additionally, Streamlit reruns the script from top to bottom upon any input change, ensuring real-time data exploration and analysis. This makes it highly suitable for building dashboards to demonstrate model predictions, visualize data insights, or even monitor live data streams. The simplicity and versatility of Streamlit have made it a popular choice for quickly prototyping data applications and sharing insights across teams or with non-technical stakeholders. Here, it creates an interactive risk analysis dashboard, displaying tables and visualizations that allow users to navigate the results. This setup makes complex data more accessible and digestible for stakeholders who may not be familiar with coding, enhancing the code's usability.

INDIVIDUAL CONTRIBUTIONS

Aritra's Contributions: Data Processing and Scoring Logic

Aritra took charge of building the backend logic for data handling and risk assessment. This included:

• Data Loading and Preparation: Aritra handled loading and structuring the JSON data, defining the mappings for risk scores and setting up the foundation for calculating likelihood and impact scores.

• Risk Score Calculations: Aritra designed the core scoring functions—calculate_likelihood and calculate_impact—that assess each asset's likelihood and impact based on various attributes. Aritra carefully defined the mappings for risk levels, compliance requirements, and security levels, ensuring accurate calculations.

• Summary Generation: Aritra created the generate_summary function to automate the generation of detailed explanations for each asset's risk score, allowing for clear interpretations of the scoring process.

• File Management: Aritra also implemented the unique filename generation for saving the processed data, ensuring no files were accidentally overwritten in multiple runs.

Suriya's Contributions: Visualization and Dashboard Design

Suriya's focus was on transforming the processed data into insightful visualizations and developing an interactive interface for end-users. This included:

• Data Visualization: Suriya used the processed data to identify the top 10 and top 50 riskiest assets, creating lists to highlight critical assets. Suriya also built a heatmap to visualize the distribution of risk scores across different environments, using color gradients to make risk levels easily distinguishable.

• Interactive Dashboard: Suriya designed the interactive dashboard using Streamlit, creating a user-friendly interface where stakeholders can explore tables of the top riskiest assets and the heatmap. Suriya ensured the dashboard's layout was clear and the data was easily navigable.

• Tooltip and Hover Text for Visualization: Suriya added hover text to the heatmap, allowing users to get detailed descriptions of each risk factor when they hover over cells, making the data both informative and interactive.

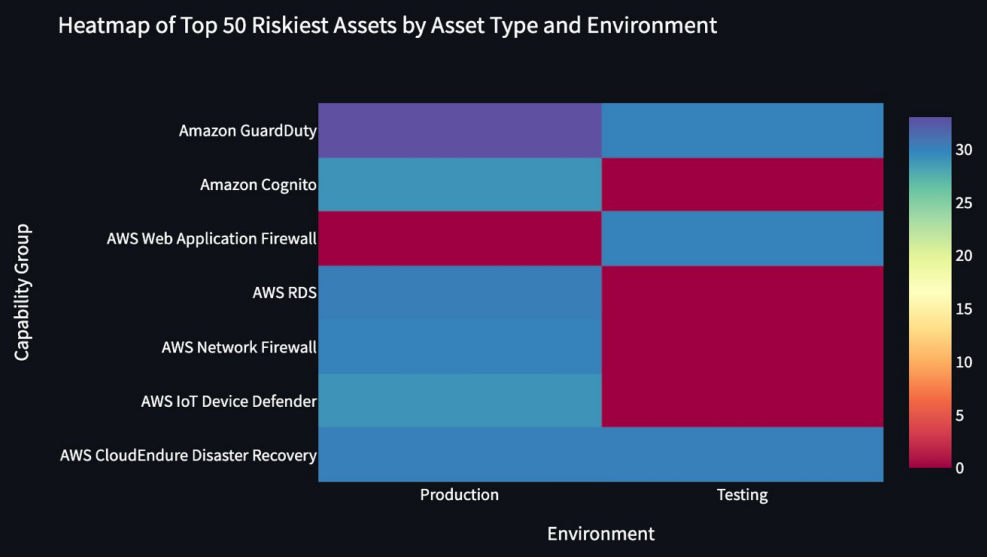
SCREENSHOTS OF THE RESULT

Risk Analysis Dashboard

Top 10 Riskiest Assets

		Environment	Owner	Confiden
12787	Defacement, External Defacement, Data Encrypted for Impact	Production	Julia Moss	High
85169	ɔulation, Data Encrypted for Impact, Inhibit System Recovery, Dis	Production	Patricia Reed	Low
11248	ɨd Data Manipulation, Disk Structure Wipe, Man-in-the-Middle, Dis	Production	Matthew Smith	High
16924	ɛfacement, Disk Content Wipe, Defacement, Data Encrypted for Ir	Testing	Eric Alexander	Medium
19730	ɒred Data Manipulation, Internal Defacement, Data Encrypted for	Production	Mrs. Donna Chung	High
25492	a Encrypted for Impact, Disk Wipe, Internal Defacement	Testing	Cassandra Walters	Low
36926	ɪt, Inhibit System Recovery, Internal Defacement, Disk Wipe	Production	Brian Larson	Low
45606	ta Manipulation, Disk Wipe, Inhibit System Recovery, Disk Conten	Production	Alyssa Miller	Medium
51695	tem Recovery, External Defacement, Data Encrypted for Impact, D	Testing	Ronald Dickerson	High
54830	ɑcement, Defacement, Disk Wipe, External Defacement	Production	Kevin Floyd	Medium

Heatmap of Top 50 Riskiest Assets by Asset Type and Environment



Details of Top 50 Riskiest Assets

	asset_type	capability_group	risks
12787	Amazon GuardDuty	Amazon GuardDuty	Web Protocols, Internal Deracem
85169	AWS RDS	AWS RDS	Disk Content Wipe, Stored Data I
11248	AWS RDS	AWS RDS	Stored Data Manipulation, Trans
16924	AWS CloudEndure Disaster Recovery	AWS CloudEndure Disaster Recovery	Stored Data Manipulation, Exter
19730	AWS CloudEndure Disaster Recovery	AWS CloudEndure Disaster Recovery	Defacement, External Defaceme
25492	AWS CloudEndure Disaster Recovery	AWS CloudEndure Disaster Recovery	Defacement, Disk Structure Wipe
36926	AWS CloudEndure Disaster Recovery	AWS CloudEndure Disaster Recovery	Stored Data Manipulation, Defac
45606	AWS CloudEndure Disaster Recovery	AWS CloudEndure Disaster Recovery	Data Encrypted for Impact, Store
51695	AWS CloudEndure Disaster Recovery	AWS CloudEndure Disaster Recovery	Stored Data Manipulation, Inhib
54830	AWS CloudEndure Disaster Recovery	AWS CloudEndure Disaster Recovery	Inhibit System Recovery, Interna
54967	AWS CloudEndure Disaster Recovery	AWS CloudEndure Disaster Recovery	Internal Defacement, External D

RECENT RESEARCH

Recent advancements in CSPM research have focused on several key areas, including machine learning integration, DevSecOps practices, and improved incident response capabilities. Researchers are increasingly exploring how CSPM tools can adapt to the dynamic nature of cloud environments, where resources and configurations can change rapidly, and the attack surface is constantly evolving.

Machine learning (ML) has emerged as a promising tool in enhancing CSPM capabilities. By applying ML algorithms to analyze patterns in cloud activity, researchers are developing CSPM tools that can detect anomalies in real time, providing proactive protection against potential security incidents. For example, ML models can be trained to recognize unusual access patterns or configuration changes, alerting security teams to potential risks even before they manifest as actual threats. This predictive approach enhances the responsiveness of CSPM tools and allows organizations to address issues proactively.

The integration of CSPM with DevSecOps pipelines is another area of research that has gained traction. DevSecOps aims to incorporate security practices into every stage of the development lifecycle, promoting a "shift-left" approach

where security is prioritized from the outset. By embedding CSPM functionalities into DevSecOps workflows, organizations can enforce security policies and compliance checks during the development and deployment phases, rather than retrofitting security after the fact. This approach not only reduces the likelihood of vulnerabilities but also minimizes delays and friction between development, operations, and security teams.

Incident response automation is also an important focus within CSPM research. Traditional incident response can be a time-consuming process, especially in large cloud environments where multiple assets and configurations must be assessed. CSPM tools that incorporate automated response capabilities can streamline this process, enabling faster containment and remediation of security incidents. For example, a CSPM tool with response automation might automatically revoke access to a misconfigured asset or apply corrective configurations based on predefined policies. This functionality reduces the burden on security teams and ensures a faster, more consistent response to security incidents.

GITHUB LINK: https://github.com/SK-DRAGO7/Risc_Score

BIBLIOGRAPHY

1) Lutkevich, B., & Rouse, M. (2023). *Cloud Security Posture Management (CSPM)*. TechTarget. Retrieved from https://www.techtarget.com

2) **Zhao, M., & Kim, S.** (2021). "A Survey of Cloud Security Posture Management." *Journal of Cloud Computing: Advances, Systems, and Applications*, 10(3), pp. 15–30. DOI: 10.1186/s13677-021-00230-4.

3) **Whitesource Software**. (2022). *The Ultimate Guide to Cloud Security Posture Management*. Retrieved from https://www.whitesourcesoftware.com

4) **Hu, V. C., Kuhn, D. R., & Yaga, D.** (2019). "Cloud Misconfiguration: Analysis, Mitigations, and Tools." *National Institute of Standards and Technology (NIST) Cybersecurity White Paper*. DOI: 10.6028/NIST.CSWP.04282019.

5) **Jin, X., & Martin, A.** (2020). "Understanding Cloud Security Posture Management." *IEEE Security & Privacy*, 18(4), pp. 45-52. DOI: 10.1109/MSP.2020.2981349.

6) **Amazon Web Services**. (2023). *Shared Responsibility Model*. Retrieved from https://aws.amazon.com

7) **Microsoft Azure**. (2022). *Best Practices for Cloud Security Posture Management*. Retrieved from https://docs.microsoft.com/en-us/azure/security

8) **Santos, J., & Martin, L.** (2021). "Mitigating Cloud Vulnerabilities through Continuous Security Monitoring." *International Journal of Cloud Computing*, 9(2), pp. 95-108. DOI: 10.1504/IJCC.2021.114587.

9) **Center for Internet Security**. (2023). *CIS Controls for Cloud Security Posture Management*. Retrieved from https://www.cisecurity.org

10) **MITRE ATT&CK**. (2023). *Understanding and Using the ATT&CK Framework for Cloud Security*. MITRE Corporation. Retrieved from https://attack.mitre.org