Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Enhancing Cloud Security Posture Management - A Comprehensive Analysis And Experimental Validation Of Cspm Strategies

Sreenivasa Yadav[1] , Dr. G Karthick[2] , Dr. C.H. Mukundha[3]

[1]Research Scholar Department of Computer Science and Engineering
Annamalai University Annamalainagar – 608 002
Email:sreenivas1803@gmail.com
[2]Assistant Professor Department of Computer Science and Engineering
Annamalai University Annamalainagar – 608 002
Email: karthick18588@gmail.com
[3]Associate Professor Department of Information Technology
Sreenidhi Institute of Science and Technology, Hyderabad
Telangana-501301. Email:reddykrishna143@gmail.com

This research examines the significance of Cloud Security Posture Management (CSPM) in optimizing and strengthening security postures in cloud environments. CSPM systems are critical for organizations to evaluate, track, and improve their security configurations through automatic risk repair, ongoing monitoring, and agreement enforcement. The study employs a mixed-method research design which consists of quantitative analysis and technical experiment. Quantitative analysis evaluates security posture metrics e.g., incident frequency, repair time, compliance scores before and after CSPM implementation. Technical experiment includes real-time assessments conducted with open-source CSPM tools such as Cloud Custodian and Security Monkey inside simulated multi-cloud environments. The experiments measure the efficacy of CSPM in real-time threat identification and automatic repair. The study's objective is to provide empirical evidences regarding the effectiveness of employing CSPM while offering acceptably actionable recommendations for organizations to enforcing or optimizing their CSPM strategies with emphasis on integration, automation, and constant improvement.

Keywords: CSPM, Cloud Security Posture Management, Cloud Custodian, Security Monkey.

1. INTRODUCTION

In current IT infrastructures, it is impossible to know how extensive cloud computing is. The major concern lies with the security starting from the mailbox to data protection [3]. This can provide immense amounts of flexibility, scalability, and efficiency that were at once out of reach. Nevertheless, this revolution towards cloud environments brings never-before-seen

security problems that will require the most appropriate security measures to protect vulnerable data and infrastructure assets. The protection of structures in cloud settings requires the nimblest of frameworks [11]. That is where Cloud Security Posture Management (CSPM) is so useful. Incorporating best-of-breed CSPM systems allow individuals to be ahead of the risks associated by unveiling, deliberating, and then treating the risks involved in any security consequence throughout cloud settings.

CSPM consists of a set of tools and practices established for compliance with security best practices, regulatory requirements, and organizational policies. These tools make it possible to oversee cloud assets, configuration settings, and user activity at all times. Consequently, they enable organizations to maintain robust security even within fast-changing cloud environments. Moreover, automating the remediation of risks helps keep vulnerabilities in check and tackle emerging threats [12-13].

1.1 CLOUD SECURITY POSTURE MANAGEMENT: AN OVERVIEW

The basic idea behind Cloud Security Posture Management (CSPM) is that it is a more proactive approach to cloud security than traditional "after the fact" security checks that happen when a cloud environment is audited or breached. CSPM is designed to catch security problems in cloud environments early and fix them fast—before they can be exploited by bad actors. CSPM doesn't just check for and correct problems, however; it also continuously checks to ensure that the security controls that are supposed to be in place are, in fact, still in place—hence the word "posture" in the term "cloud security posture management."

Managing Visibility and Inventory

Continuously discovering and cataloging cloud resources, CSPM tools provide complete visibility into cloud environments, including virtual machines, storage buckets, databases, network components, and more. By keeping an up-to-date inventory of cloud assets, their configurations, and their security postures, CSPM helps organizations understand what they have adopted and how [14].

Managing the Setup and Assurance

One of the primary tasks of CSPM is to validate and impose configuration regulations that are centered on greatest practices and obedience requirements. CSPM tools are used to compare these contemporary breaks from custom that might tip off aggressive and disorder safety modules by evaluating them against a defense strategy show predefined laws and ascertains. Making sure that installations are paying to practices such a the GDPR treaty, HIPAA ordinance, the PCI-DSS code assists with control government and should be demonstrated to shareholders and controller strategy-compliance gets control fractures from unchanged accidental barred data for company finance investigations. WideString and are closely attached to consistency modules and provide organizations with clear comprehension allowing them to comprehensive enforcement and wholes effectiveness of remediation procedures with a vast range of variety systems [15-17].

Determining Risk and Identifying Threats

CSPM tools assess risks and identify vulnerabilities to maintain continuous visibility. Using predefined rules and machine learning, these tools search for potential threats and security

weaknesses, such as open ports, over-permissive access settings, and outdated software versions. More advanced solutions also augment their capabilities with feeds of threat intelligence to detect emerging risks and attack vectors [16].

Automation

One of the most important abilities of CSPM is automated remediation, which ensures that identified problems are addressed rapidly. CSPM technology may self-correct misconfigurations, strip out unnecessary permissions, and deploy security updates, all without manual overrides; this cuts the time vulnerabilities linger and makes sure best practices are repeatedly upheld. Some CSPM solutions also bring bespoke automation scripts on the table, letting enterprises design remedies customized for their own circumstances [18-21].

Constant Tracking and Reporting

CSPM tools continuously monitor cloud environments, providing consistent application of security policies and a persistent, strong security posture. Continuous monitoring involves scanning configurations and activities in real-time, alerting security teams to any deviation from established policies. Additionally, CSPM tools generate detailed reports and dashboards, providing insight into security trends, compliance status and areas needing attention [22].

1.2 TECHNICAL COMPONENTS OF CSPM

Cloud Orchestration

Cloud security posture management (CSPM) solutions work with many different cloud service providers (CSPs). The CSPs that tend to be used the most are AWS, Azure, and Google Cloud. The CSPM solutions we'll discuss in this paper are designed to work with those providers. They do this by using the application programming interfaces (APIs) those providers make available to authorized users to (among other things) look at resources running in a CSP, the configurations of those resources (which may or may not be compliant with some set of standards or rules), and any security events or logs at the CSP [23-25].

Policy and Its Management

The Cloud Security Posture Management (CSPM) tools have built-in and preconfigured security policies that are directly related to industry standards and the best security practices. These policies are just like armies of ants that get to work right after being customized to the specific requirements of a given organization. Policy management encompasses a lot of functions, but it mainly revolves around the idea of being able to ensure that the policies that are in the CSPM tool. And seeing how important these policies are is kind of the key [26].

Gathering Data and Analyzing It

Cloud security and posture management (CSPM) tools gather data 24/7 from cloud environments. This includes a multitude of types of data, like configuration settings, user account activity, and network traffic. It feeds those things to various analytical engines. Some CSPM tools use basic engines that run rules: "If you see this, do this" kinds of things. Many others make use of machine learning to enable some kind of smart prediction. And this is where the work of the CSPM engine should kick in to assist in solving the two problems, it

should enable smart prediction about when a security incident is likely to occur, and it should have good recommendations about what to do next. [27-29]

User Interface and Reporting

User Interface and Reporting are fundamental to the measurement of customer service and to the operation of any truly effective CSM system. The purpose of CSMUI and CSMR is to enable fast, effective, and easy-to-use tools with which service providers and key customer-facing personnel can access, use, and interact with the CSM database. The UI and Reporting solution must be designed for optimal performance and ease of use by a diverse set of users who will be using a number of different devices (e.g., desktop computers, laptops, thin clients, ruggedized laptops, tablets, and/or smartphones) to access it [30].

CSPM tools offer security administrators easily usable interfaces to interact with the system. Administrators can access the necessary information in real time via the dashboard to see the security situation, which is just one key strength of a CSPM tool. And the tool doesn't just provide any real-time information—it provides the real-time information necessary for sound decision-making. In addition, a CSPM tool can aid a security administrator in properly responding to and remediating a threat [31].

1.3 PROBLEM STATEMENT & MOTIVATION

Using the Cloud Security Posture Management (CSPM) platform can be tremendously helpful. However, many organizations find it difficult to implement these solutions properly. It is not always easy to integrate CSPM with existing legacy systems. Organizations must also face the "alert fatigue" problem. It can be tough to decide which security alerts to pay attention to when you have so many coming in. And the one you didn't pay attention to could be the opening that lets an even more significant security event happen.

To be effective, CSPM must provide comprehensive visibility into cloud infrastructure, identify misconfigurations and vulnerabilities, and enable timely remediation. CSPM features include automated policy enforcement, threat detection, incident response, and compliance auditing. This way, CSPM can be proactive, organizations become less liable to both data breaches and compliance violations [33].

In this regard, this research intends to carry out an extensive analysis and empirical verification of CSPM strategies for amplifying cloud security posture. By employing a multi-methods approach that blends quantitative analysis and technical experimentation, the main objective is to assess the association between CSPM and security posture metrics and valid its effectiveness in practical scenarios. To achieve these aims, by using open-source CSPM tools like Cloud Custodian and Security Monkey, and also commercial solutions, this research explains to offer an experimental background that shows the effectiveness of CSPM and proposes a roadmap to adopt or optimize CSPM's programs in other organizations [31-32].

1.4 CONTRIBUTIONS

Our proposed solution combines remediation scripts into the Continuous Improvement Loop to detect potential security problems in real time. With tailored, pre-built remediation scripts

for popular security issues, organizations can minimize response times and remediate risks immediately.

By taking a proactive stance, organizations will further strengthen the robustness of their security efforts and reduce the amount of manual work required to address security incidents.

We prioritize refining security policies iteratively using analysis results and feedback. Continuously updating security policies adapts security measures to mitigate new threats, comply with evolving regulations, and reflects feedback from diverse stakeholders. By iterating on policy, security remains in line with business objectives, minimizing risk in multi-cloud environments.

Using pre-built threat intelligence & machine learning algorithms to perform anomaly detection is an essential part of our proposed approach. Businesses are able to identify and mitigate risks ahead of time; by sending an early warning of detrimental anomalies and potential security issues.

Organization of the paper is as follows: Section II of this paper offers a comprehensive Literature Review that explores existing studies, frameworks, and methodologies in the realm of CSPM. Section III, presents the Proposed Methodology. By spotlighting the Continuous Improvement Loop algorithm, this portion of the paper reveals intricacies such as automated remediation, statistical analysis, policy refinement, and anomaly detection, which are the pivots of security posture improvement. Section IV focuses on the experimental results along with a detail regarding the experimental set up and demonstration of results obtained from real world affirmation of the proposed methodology. Section V concludes the paper.

2. LITERATURE SURVEY

Cloud Security Posture Management (CSPM) tools have been gaining popularity to automate, monitor, and visualize the security posture of multi-cloud environments. The basis of risk assessment is being able to first analyze every vulnerability and then to be able to assign a risk value to each one. But the National Vulnerability Database (NVD), where we can look up the number of vulnerabilities and check their risk value, is currently overwhelmed—more overwhelmed than ever, really. In late 2020, it housed over 144,000 vulnerabilities that must have risk values assigned to them for our new normal. At the moment, human-directed initiative is responsible for identifying most vulnerable spots in code. Moreover, open-source libraries do not always adhere to the vulnerability reporting conventions set in place by the CVE and NIST organizations. Instead, they often employ GitHub's issue feature for working through bugs and security problems. To overcome these difficulties, they present NL2Vul, a tool that uses natural language processing. NL2Vul uses deep neural networks that have been trained on NVD's descriptions of software vulnerabilities to determine the probability of a score for a vulnerability. To derive that probability score, NVD deep learning models use text obtained from vulnerability descriptions. The models have been trained by exposing them to a huge amount of text data. On top of NVD's text data, the deep learning models use also text data derived from the threat intelligence sector [1].

Authors in [2] designed a tool for Cloud Security Posture Management (CSPM). The tool promises to keep a close watch over cloud assets and to be especially diligent in its efforts when it comes to those assets residing in Amazon Web Services (AWS). The whole concept of CSPM is centered around the National Institute of Standards and Technology (NIST) Cybersecurity Framework version 1.1 or NIST CSF for short. CSPM is supposed to be an effective risk management tool in the cloud, and we will see that it does so in very beneficial ways. The CSPM tool can be modified to meet the way your organization does security. You can use Amazon Web Services (AWS) services to enable that modification. Their paper discusses the design of the CSPM tool and the parts of its architecture that make it work. It also looks at what the tool can do to enhance security and compliance in cloud computing. Since part of the tool involves AI and Big Data, it requires careful planning and implementation, so this paper touches on that as well. Their work aligned with [4] has a direct effect on cloud data management and ensures that we have access to our data regardless of where we are digitally connected, which in turn has a lot of effects on smart cities, one of their most vital being smart mobility [4].

Dantas et al. [5] claimed that the Security is one of the biggest concerns for cloud infrastructures, which are vulnerable to a variety of threats, including external and internal threats. In the absence of proper security mechanisms, these threats can violate the security properties of cloud-hosted services. In order to protect cloud infrastructures from threats it is important to conduct threat analysis in the early stages of system development (i.e. system architecture design phase). Threat Analysis and Risk Assessment (TARA) is one of the well-known approaches proposed by researchers and practitioners. TARA has several activities, for example, asset identification, threat scenarios, attack paths, and risk treatment decision. The decision-making process for treating risk is when security measures are selected to reduce threat scenarios. In the current practice, the TARA process is performed manually by engineers, which can be time consuming and contain errors. In their work they initiated a line of work aiming at automating TARA activities. As a first result we proposed a logic programming tool which enable automation of those TARA activities composed of the automation of the recommendation of the security control measures. In this paper we propose a last plug-in of Model-Based Systems Engineering (MBSE) to secure cloud architectures: a Security Pattern Synthesis. Their plugin has been implemented in Java. It uses our logic programming tool to reason on the security of the cloud architecture.

An et al. [7] presents CloudSafe, a new cloud security assessment and enforcement tool, which allows for automating security assessment and enforce best security control for the cloud based on integrating multiple existing security tools. To illustrate the practical utility and usability of CloudSafe, they deployed and conducted security assessment for CloudSafe in the Amazon AWS. Furthermore, they conducted an in-depth analysis on four potential countermeasures; Vulnerability Patching, Virtual Patching, Network Hardening and Moving Target Defence. For the purpose of the project, it was determined that deploying Virtual Patching, Network Hardening, and Moving Target Defence was a viable approach. They developed proof of concepts to show how effective each possible countermeasure option would be. Their findings suggest that CloudSafe is a useful and efficient instrument intended

for security administrators, which allows them to pick out the most suitable countermeasures to ensure security of their cloud environment by deeply assessing security.

Another study by Sibi Chakkaravarthy et al. [6][10][18], the researchers managed to address the many challenges the IT security industry has faced and the reasons for the urgent need for the containerized honeypot solution. Their study examines a range of container honeypot strategies, all of which share the same broad design idea: One or more executable containers are hosted on a computer and made to appear vulnerable, running services and listening on network ports that look like they could be exploited [8-9]. Optionally, containers can also simulate attracting traffic by, for example, hanging on after clients have connected, persisting with a kind of mirage of resolved content that makes the clients willing to do ID checks after the set up. Using a group of research honeypot sensors, they are able to figure out and study hacker behavior. They've taken that observational approach, along with a lot of data they've collected, and put together a pretty good representation of what sorts of things a hacker does and why [6].

The adoption of multi-cloud environments is on the rise, and with it, the manifold puzzles of managing and securing this newly complex digital infrastructure are taxing the industry. They are solving for the problem of how to ensure that all of an enterprise's digital presence—regardless of location—adheres to an optimal and high-security configuration. Secure access service edge (SASE) is back in the conversation because it offers a comprehensive solution that could overhaul businesses' long-established security framework. The Cloud Security Posture Management (CSPM) market, part of a wider cybersecurity sector of some $173 billion, was worth an estimated $675 million in revenues for 2019, driving in products from startups, specialist service providers, and enterprise software firms. But taken as a group, existing CSPM solutions have a credibility problem and fail on many fronts. Hence the current landscape demands a more robust, scalable, and efficient approach to managing cloud security posture management.

3. THE PROPOSED MIXED METHODOLOGY FOR CLOUD SECURITY POSTURE MANAGEMENT

The mixed-methods research strategy in this study combines two research approaches: analysis of numerical data and experimentation with technology to assess the effectiveness of security tools in "truly" cloud-based environments. To accomplish this, the study uses two related tools (meaning that their results are comparable): one open-source (in this case, Cloud Custodian) and one commercial (in this case, Security Monkey). These tools represent two different ways of achieving the same goal; they are designed to patrol cloud-based architectures at the construction and C2M stages and look for security issues. Figure 1 shows the architecture of the proposed approach.

To implement a mixed-methods approach for Cloud Security Posture Management (CSPM), an architecture needs to be built with various components to enable comprehensive data collection, analysis, and validation within a cloud environment. This architecture utilizes both quantitative and qualitative methods, emphasizing real-time monitoring, automated remediation, and compliance management using CSPM tools.

3.1 THE PROPOSED APPROACH WORKFLOW

During this stage, security policies are reviewed and improved on a regular basis. New regulatory rules, threat intelligence, and security audit feedback are all taken into account. Whenever new vulnerabilities are found and insights are gained, CSPM instruments are adjusted and optimized. This continuous cycle breeds a culture of readiness, allowing businesses to anticipate and counter evolving dangers while preserving solid compliance and control criteria.

Data collection layer

The first step of the approach is to establish end-to-end data collection from cloud environments including AWS, Azure and GCP. The objective here is to capture baseline security metrics before implementing CSPM tools. These metrics include incident count, time to remediate, compliance scores and configuration drift. Cloud-native monitoring services such as AWS CloudWatch, Azure Monitor and Google Cloud Operations Suite were used to collect data, complemented by logging capability CSPM tools have [34]. In an ongoing process and after deploying CSPM solutions, always-on data collection was performed continually assessing the security posture and detecting real-time changes and progresses across cloud infrastructure or in the universe of cloud accounts.

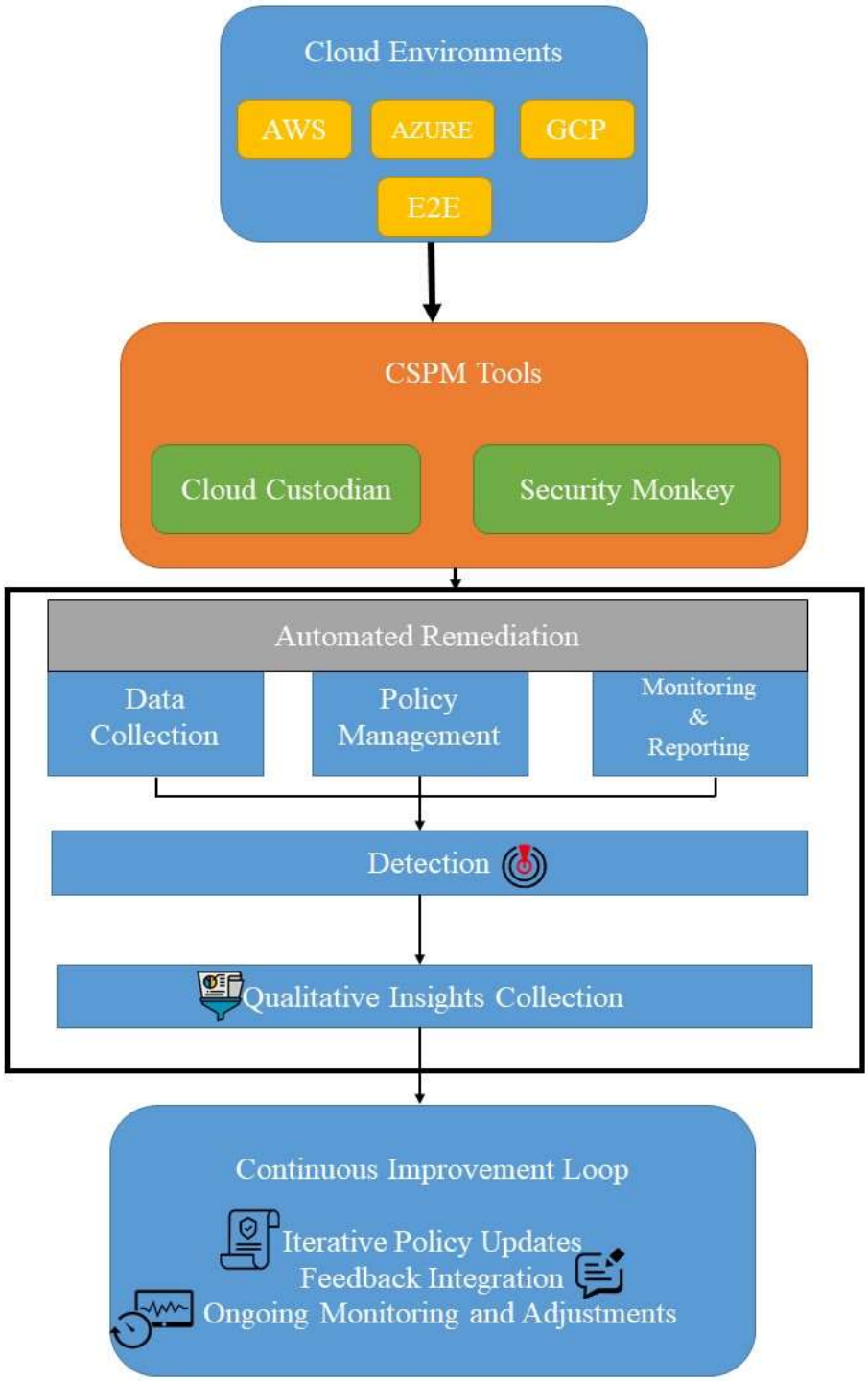
Cloud Environments
AWS
AZURE
GCP
E2E
CSPM Tools
Cloud Custodian
Security Monkey
Automated Remediation
Data Collection
Policy Management
Monitoring & Reporting
Detection
Qualitative Insights Collection
Continuous Improvement Loop
Iterative Policy Updates
Feedback Integration
Ongoing Monitoring and Adjustments

Figure 1. Architectural Representation of the proposed approach

Policy Definition and Management

During the definition and management of a cloud security policy, it is outlined by using best practices and by following specific guidelines like GDPR, PCI-DSS, HIPAA. The cloud security policies cover various subjects in cloud security like Access control, data encryption, network security and compliance checks. CSPM tools such as Security Monkey, Cloud Custodian map these policies into set of automated scripts and rules. CSPM tools continuously enforce these cloud security policies by comparing real-time configuration states with these defined security baselines, identify deviations and initiate automated scripts for remediation when and wherever required. Following which, cloud infrastructure remains compliant to predefined SLA and regulations, further reducing the risk of security breaches.

Monitoring and Detection

The continuous monitoring and detection phase utilizes real-time features of CSPM tools to oversee cloud resources, configurations, and activities. This includes enabling CSPM tools' continuous monitoring capabilities and integrating threat intelligence feeds to keep abreast of evolving threats and vulnerabilities. AI-based machine learning algorithms are utilized to improve threat detection, uncovering irregular behaviors and prospective security incidents. The monitoring system captures comprehensive logs of configuration changes, access patterns, and network traffic to send alerts when the documented security policies are breached. This real-time surveillance is essential for the early spotting and efficient response to security adversities.

Automated Remediation

The automation-based fixing of CSPM errors is an essential part of the CSPM strategy. This involves writing the fix-it scripts and workflows for the commonly found CSPM issues such as minimum required control configurations, open ports or misconfigurations. Let’s take examples of scripts I mentioned, we can have simple scripts that can add tags, start noncompliant instances, enforce field level encryption and fix access controls. When these errors gets fixed by the scripts automatically, we reduce the security window or increase the part of your cloud into the compliant system. In addition, we need to continuously validate the actions over time, to guarantee if the auto remedies have been effective and appropriate.

Data Analysis and Reporting

The phase of the project that scrutinizes the effectiveness of CSPM tools concerns itself primarily with paying close attention to the relationship between a set of security metrics that the solution can collect and the overall security posture of the cloud environment. This means employing classic and elegant methods like comparing the pre- and post-implementation scenarios by using a t-test or analysis of variance (ANOVA) to see if the CSPM solution—perceived as a parameter in our statistical analysis—can be associated with some sort of improved security outcome. For example, incidents that are classified as being security-related, happen with less frequency. Obviously, increased remediation times for incidents (i.e., the time it takes to fully roll back the effects of a particular incident on the environment) would also be a great improvement.

Qualitative Insights Collection

Apart from the quantitative analyses, qualitative insights are gathered to perform a holistic assessment of the CSPM tools. Input is obtained from information security experts and IT practitioners by means of a survey and a technical evaluation, the survey and evaluation tally the feedback of the practitioners on usability of CSPM, integration challenges of CSPM with existing tools and overall effectiveness of CSPM in real world environment. The experts provide input to the practicality aspect of CSPM tools like - ease of configuration, UI experience, and impact on existing processes.

Continuous Improvement Loop

To continually refine and optimize CSPM strategies, the Continuous Improvement Loop leverages findings from both quantitative and qualitative analysis. They further rely on the latest threat intelligence, expert feedback, and on-going monitoring insights to keep security policies up-to-date. Furthermore, to improve their efficiency and usability and keep them aligned with constantly changing security requirements and industry standards, changes are made to CSPM tool configurations. The Continuous Improvement Loop will ensure the CSPM strategies and approaches keep up with evolving threats, stay CIS guidelines compliant and have an optimized security posture to provide a resilient cloud security framework that is adaptive to the rapid changes in the cloud security landscape.

4. EXPERIMENTAL RESULTS: A STEPPED APPROACH

At the start, the cloud computing infrastructure is established on diverse platforms, like AWS, Azure, and GCP. The necessary components must be constructed on each one. For AWS, a Virtual Private Cloud (VPC) must be made; this consists of a small network of secure and logically separated computing resources. That network must be further divided into multiple subnets. Within that VPC, at least one EC2 instance must be created, along with other necessary services like S3 buckets, RDS instances, and Lambda functions.

In Azure, you create a Virtual Network (VNet) with Azure VMs, Blob Storage, SQL Database, and Azure Functions. Likewise, in GCP, you create a Virtual Private Cloud (VPC) with Compute Engine instances, Cloud Storage buckets, Cloud SQL instances, and Cloud Functions.

After that, CSPM tools such as Cloud Custodian and Security Monkey (and commercially available tools like Prisma or Dome9 for comparison) are installed. Cloud Custodian is set up for serverless execution as an AWS Lambda function or a Docker container for portability. Security Monkey gets installed on an Amazon EC2 instance or equivalent virtual machine in Azure and GCP regions. These tools monitor and can automatically report on security configurations.

Data Ingestion & Collection

To start, gather baseline security metrics before implementing cloud security and posture management (CSPM) tools. Use cloud-native monitoring tools like AWS CloudWatch, Azure Monitor, and Google Cloud Operations to collect data about the state of security. Some of the main metrics to track are the number of incidents that happen over a given period (incident frequency) and how long it takes to solve those incidents (remediation time).

Policy Definition and Configuration

In this security phase, industry standards and organizational requirements are used to define and configure security policies. A security policy framework is implemented, ensuring that many critical areas of security are adequately addressed. This framework covers critical areas such as access control, data encryption, network security, resource configuration, and compliance checks.

Cloud Custodian is a policy-as-code (PAC) tool that serves as the workhorse for policy definition and enforcement. It is capable of enforcing a wide array of policies expressed in a clear, concise, and expressive language (i.e., YAML). Once defined, Cloud Custodian and Security Monkey (a continuous monitoring tool) form the basis for policy enforcement and reporting, all while the actual infrastructure is in place and operating. Algorithm1 explains the function to define initial security policies based on industry standards and organizational requirements

Algorithm 1: Initial security policies

```
def define_security_policies():
    policies = {
        "access_control": "enforce_least_privilege",
        "data_encryption": "enable_encryption",
        "network_security": "restrict_open_ports",
        "resource_configuration": "use_approved_instance_types",
        "compliance_checks": "automate_compliance_scans"
    }
    return policies
```

Real-Time Monitoring and Detection

To detect security breaches and noncompliance events, we don't rely on manual checks but install continuous real-time monitoring. CSPM tools are not just running on Nessus-like cloud environment scanners checking the data they find against an archaic DB of "best practices" to come up with compliance and security reports. We've performed more than manual scanning performed by security pros and instead have turned to using machine learning. Installed at two data centers for the moment: one runs more than 10 high-performance computing nodes with GPUs, and the other is the off-site backup center. Algorithm 2 explains the function to deploy and configure monitoring tools.

Algorithm 2: Deploy and configure monitoring tools

```
def deploy_monitoring_tools():
    setup_aws_cloudwatch()
    setup_azure_monitor()
    setup_gcp_operations()
    integrate_threat_intelligence_feeds()
```

Security Operation & Automation

Fixing security issues without pause is a crucial element. It eliminates all chances of manual nature and directs the quick correction of faults that identity has uncovered. Indeed, the Automation and Orchestration (A&O) set-up is primarily responsible for this function's execution, predominantly visible in the way it automatically runs through the Security Monkey (SM) pipeline to push notified security incidents into a certain state. Scenarios to validate occurrences are also automatically executed (though not at the same beat, and not also in the cases where a system level change may risk an occurrence of a "false positive"). Algorithm 3 explains the functions such as anomaly detection, current security metrics, statistical analysis on collected metrics, analyze metrics and identify trends, CSPM tool configurations.

Algorithm 3: SOAR CSPM Functions

Algorithm 3.1: Anomaly Detection

```
def enable_anomaly_detection():
    setup_behavioral_analytics()
    deploy_ml_algorithms_for_anomaly_detection()
```

Algorithm 3.2: Automated Remediation Scripts

```
def develop_remediation_scripts():
    remediation_scripts = {
        "s3_public_access_removal": "remove_public_access_script",
        "enable_vm_backup": "enable_backup_script",
        "apply_lifecycle_policies": "apply_lifecycle_policy_script"
    }
    return remediation_scripts
```

Algorithm 3.3: CSPM configuration script for automated remediation

```
def configure_cspm_remediation(remediation_scripts):
    configure_cloud_custodian(remediation_scripts)
    configure_security_monkey_notifications()
```

Algorithm 3.4: Collection of current security metrics

```
def collect_current_metrics():
    current_metrics = collect_metrics()
    return current_metrics
```

Algorithm 3.5: Statistical analysis over collected metrics

```
def perform_statistical_analysis(baseline_metrics, current_metrics):
    analysis_results = compare_metrics(baseline_metrics, current_metrics)
    return analysis_results
```

Algorithm 3.6: Analyze metrics and identify trends

```
def analyze_metrics(current_metrics):
    trends = identify_trends(current_metrics)
    improvements_needed = detect_areas_for_improvement(current_metrics)
    return trends, improvements_needed
```

Algorithm 3.7: Security Policy updates

```
def update_security_policies(trends):
    updated_policies = refine_policies(trends)
    return updated_policies
```

Algorithm 3.8: Reconfiguring CSPM tool configurations

```
def adjust_cspm_configurations(policies):
    reconfigure_cloud_custodian(policies)
    reconfigure_security_monkey(policies)
```

Algorithm 3.9: Feedback

```
def collect_qualitative_feedback():
    feedback = gather_expert_feedback()
    return feedback
```

Algorithm 3.10: Policy redefinition

```
def refine_policies_based_on_feedback(feedback):
    refined_policies = refine_policies(feedback)
    return refined_policies
```

Algorithms 3.1 to 3.10 present the proposed steps to optimize the Cloud Security Posture Management (CSPM) system using anomaly detection, automated remediation, configuration scripting, metric collection, statistical analysis, trend identification, policy updates, tool reconfiguration, feedback collection, and policy refinement holistically. Algorithm 3.1 exploits machine learning techniques for anomaly detection so that CSPM can distinguish abnormal behaviors and identify security threats. Algorithm 3.2 develops and configures automated remediation scripts. Algorithm 3.3 configures CSPM tools to perform automated remediation processes described in Algorithms 2.1 and 2.2. Algorithm 3.4 focuses on gathering fresh security metrics to perform experiments. Algorithm 3.5 employs statistical analysis on collected metrics to expose trends. Algorithm 3.6 consists of analysis about metrics which pinpoint areas needing improvement. Algorithm 3.7 selects security policies to update based on the analysis results. Algorithm 3.8 performs CSPM tool reconfiguration based on policies updated in the Algorithm 3.7. Algorithm 3.9 draws qualitative feedback from experts. Algorithm 3.10 exploits this feedback to refine policies, assuring the CSPM system continually improves to effectively protect cloud environments.

Reporting and Visualization

Visualizing the security posture improvements are achieved through detailed reports and dashboards. By analyzing the collected data, we can summarize security metric changes, compliance status as well as incident response effectiveness. Grafana or native cloud dashboards allow the creation of visualizations to stakeholders that can effectively communicate the findings.

Algorithm 4: Reporting

```
def generate_reports_and_dashboards(current_metrics):
    create_reports(current_metrics)
    create_visual_dashboards(current_metrics)
```

Algorithm 4.1: Creating reports

```
def create_reports(current_metrics):
    report = {}
    report['summary'] = create_summary(current_metrics)
    report['incident_trends'] = analyze_incident_trends(current_metrics)
    report['compliance_status'] = check_compliance_status(current_metrics)
    report['remediation_effectiveness'] = evaluate_remediation_effectiveness(current_metrics)
    save_report_to_file(report)
    send_report_to_stakeholders(report)
```

Algorithm 4.2: Creating summary

```
def create_summary(current_metrics):
    summary = {
        'total_incidents': len(current_metrics['incidents']),
        'resolved_incidents': len([i for i in current_metrics['incidents'] if i['status'] == 'resolved']),
        'compliance_score': current_metrics['compliance_score']
    }
    return summary
```

Algorithm 4.3: Analyze Incident Trends

```
def analyze_incident_trends(current_metrics):
    trends = {
        'incident_types': categorize_incidents(current_metrics['incidents']),
        'incident_frequency': calculate_incident_frequency(current_metrics['incidents']),
        'time_to_remediate': calculate_remediation_times(current_metrics['incidents'])
    }
    return trends
```

The creation of the report starts with Algorithm 4.1, which aggregates the metrics in a series of sections, the summary of major findings, the scrutiny of incident trends, the verification of compliance status, and the assessment of remediation effectiveness. The purpose of Algorithm 4.2 is to create a concise summation of key measures, including total incidents, resolved incidents, and compliance scores. Next, Algorithm 4.3 goes through the incident history over time. It helps to reveal the pattern or change of security incidents in order to provide a general idea about the update of security threat.

Algorithm 4.4: Verify Compliance Status

```
def check_compliance_status(current_metrics):
    compliance = {
        'policies_violated': list_violated_policies(current_metrics['compliance_checks']),
        'non_compliant_resources': identify_non_compliant_resources(current_metrics['compliance_checks'])
    }
    return compliance
```

Algorithm 4.4 verifies the compliance state of the cloud against a fixed set of security policies that describe concisely and correctly relevant compliance requirements, express cloud security compliance and violations in a platform-independent manner, and initiate rectification strategies for violations or corrective actions.

Algorithm 4.5: Evaluate Remediation Effectiveness

```
def evaluate_remediation_effectiveness(current_metrics):
    effectiveness = {
        'remediation_success_rate': calculate_remediation_success_rate(current_metrics['remediation_actions']),
        'average_remediation_time': calculate_average_remediation_time(current_metrics['remediation_actions'])
    }
    return effectiveness
```

Algorithm 4.5: Save Report to File & Send it to the stakeholders

```
def save_send_report_to_file(report):
    with open('security_report.json', 'w') as file:
json.dump(report, file)
email_report_to_stakeholders(report)
```

Algorithm 4.5 is used to measure the effectiveness of remediation actions to address security issues. It measures the success rates and the average time to complete the remediation. Lastly, Algorithm 4.6 stores the compiled report onto a data file and delivers it to necessary recipients, providing timely release of significant inferences and suggestions regarding the progress of cloud security.

Algorithm 5. Sample pseudocode for Continuous improvement loop

```
# Continuous Improvement Loop
def continuous_improvement_loop():
    policies, baseline_metrics = initialize_security_policies()
    setup_continuous_monitoring()
    setup_automated_remediation()

    while True:
        current_metrics = collect_current_metrics()
        perform_statistical_analysis(baseline_metrics, current_metrics)
        trends, improvements_needed = analyze_metrics(current_metrics)

        if trends indicate issues or improvements:
            policies = update_security_policies(trends)
            adjust_cspm_configurations(policies)

        qualitative_feedback = collect_qualitative_feedback()
        if qualitative_feedback:
            policies = refine_policies_based_on_feedback(qualitative_feedback)
            adjust_cspm_configurations(policies)

        generate_reports_and_dashboards(current_metrics)
        baseline_metrics = current_metrics  # Update baseline for next iteration

# Execute the continuous improvement loop
continuous_improvement_loop()
```

Continuous Improvement Loop

The continuous improvement loop integrates the outcomes of the experiment and uses them to sharpen and optimize the CSPM's strategies. From experiment to experiment, the policies and also the tools of the CSPM are updated and adjusted. Mostly, this is a behind-the-scenes activity, invisible to the bulk of cloud's users. But it's necessary to maintain the kind of adaptive, evolving intelligence capability that keeps an organization (and its cloud) safe and secure. Algorithm 5 details each step needed in a Continuous Improvement Loop including initializing security policies and baseline metrics, deploying monitoring tools, configure automated remediation, collect current security metrics, perform statistical analysis, analyze metrics for trends, update security policies, collect qualitative feedback, refine policies, generate reports and dashboards, and update baseline metrics for the next iteration.

Real time results

A real-time test environment with the cloud infrastructure equipped with the ELK (Elasticsearch, Logstash, Kibana) stack including Suricata for intrusion detection is deployed in a AWS and E2E cloud. The ELK stack is configured with Elasticsearch for data storage and indexing, Logstash for data processing and enrichment, and Kibana for data visualization and analysis.

Utilizing Suricata, a corresponding detection system, allows for constant monitoring of malware traffic and alerts the user on threats. With modern technology today, an additional ELK stack integration with Suricata gives the capability of ingestion and traffic analysis. As Suricata grabs these events, it can instantly forward it to logstash to process and enrich them before pushing the events for indexing in Elasticsearch.

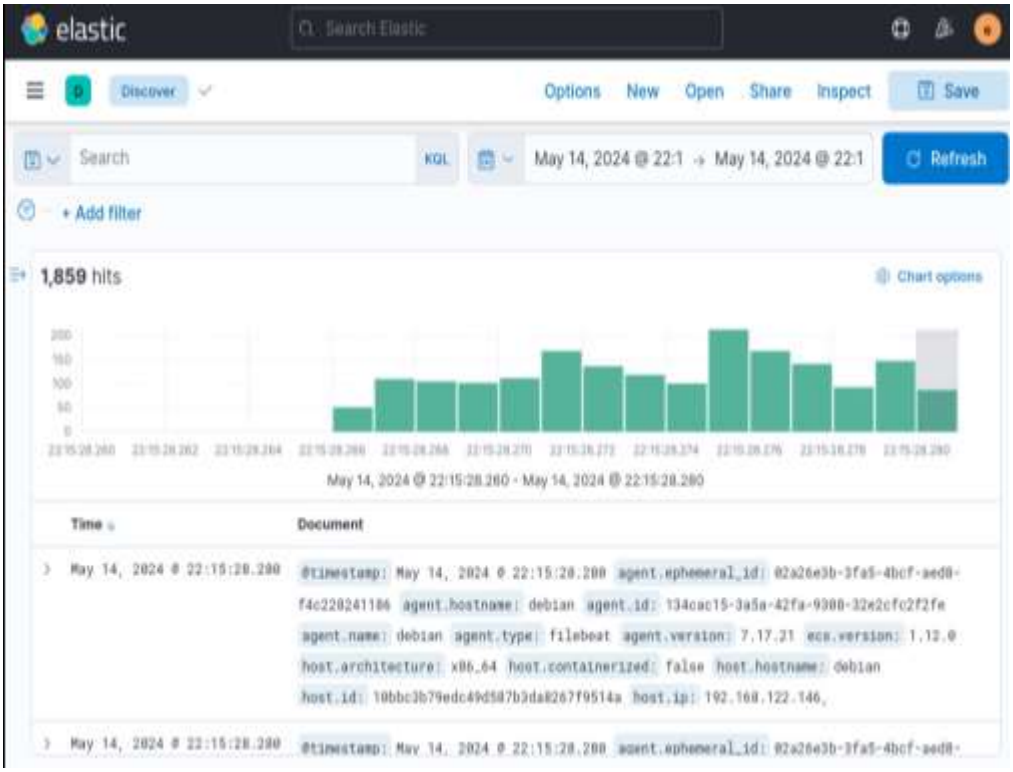

Figure 2. Real time events collected (logs).

By enabling the visualisation and analysis of security events in real-time, Kibana helps to facilitate proactive measures in incident response and remedies. Using the ELK stack, baseline security metrics are gathered so as to give a basis for the evaluation of deviations and anomalies associated with security incidences.

Next, the Continuous Improvement Loop is carried out, during which network traffic and system logs are under continuous monitoring; security events are under real-time analysis and correlation with historical data and predetermined security policies; feedback from stakeholders is sought for insights into the effectiveness of the Continuous Improvement Loop, findings from the experimentation are documented comprehensively.

The findings are aggregated and synthesized in a comprehensive document, spotlighting the competence and efficacy of the Continuous Improvement Loop in heightening safety state within the organization's cloud space, accompanied with suggestions for further polishing and future study emphases.

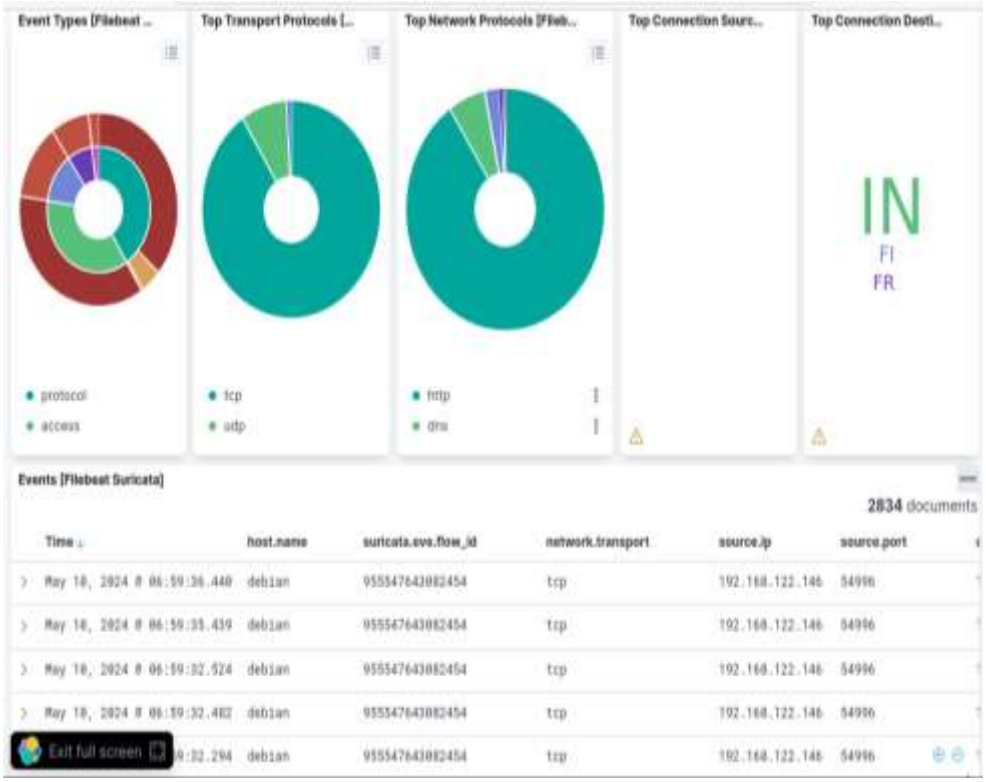

Figure 3. Results of classified events flagged by Suricata.

Figure 4. Suricata Event Alerts

The outcomes of classified events noticed by the Suricata intrusion detection system (IDS) are shown in Figure 3. Figure 3 demonstrates that how Suricata is able to thoroughly analyze the content of network traffic and system logs. Figure 3 also emphasizes how Suricata categorizes the content being analyzed. Figure 4 presents the Suricata event alerts in the Kibana dashboard in the ELK stack. The alerts are produced by Suricata in real-time to signal security incidents detected. An alert has many fields with rich information, for example: the threat type, the impacted entity, the source and destination IPs, timestamp in milliseconds, etc. Moreover, these alerts are separated by color to differentiate the severity level, hence, security personnel can catch the urgent problem as quickly as they can. Through this view, we can monitor our traffic, and estimate the situation, from which we ensure our safety status. When a specific event is in process, our incident response team can take swift actions to minimize the related damage. On another hand, the Suricata with the ELK stack has added the security level into this organization's horizons for 24/7 security services, which means that our system steady with the Suricata, keep warning if necessary, check the alerts, and analyze the relevant log data. Through the logs, we can recognize our lacking points, then improving our security solution better and better, keeping the situation is in the best range.

5. CONCLUSION

This paper investigates a novel technique for enabling a more effective and efficient Cloud Security Posture Management (CSPM). We present a novel technique by incorporating rule-based remediation, statistical detection, policy evolution, and machine learning-based anomaly detection within a Continuous Improvement Loop to compositions to conquer the limitations of existing methods for generating an effective CSPM when in action. We develop our approach based on the ELK stack for real-time data collection and visualization, Suricata for IDS, and create a novel framework for bespoken and adaptive CPSM to automate process of CSPM with a vision to analyzing, refining and remediating the cloud infrastructure posture against a pervasive, advancing and evolving cloud threat scape.

Our methodology has been shown to be feasible by the experimental setup. Suricata, through a high-fidelity malware detector, was integrated with the ELK stack and drug the connection to monitor and evaluate security incidents on-the-fly, delivering seamless integration with the ELK stack and Suricata. The experimental results well supported the success of our mechanism for detection and classification of threats. The mechanism consequently eases the task of response to remediation and maintenance of security policies with prompt detection of the new features, monitoring of data sets with different dimensions, and observation of expert feedback.

Our methodology proposal takes a big step forward in detecting and preventing security incidents and ensures that security measures stay up with constantly emerging threats and needs. The iterative refinement of policies and continuous feedback loop mean that organizations can stay ahead of the cybersecurity curve in a proactive way, protecting the cloud infrastructure against constant barrage of threats.

Potential future research avenues may involve enhancing anomaly recognition machine learning algorithms, advancing the scale of automated remediation, and generalizing

the effectiveness of proposed methodology across multiple cloud platforms and configurations. A continuous improvement system design is demonstrated throughout this research, which provides a strong basis for pushing forward CSPM practices and ultimately intensifying cloud security and resilience.

REFERENCES

1. Bulut, M. F., & Hwang, J. (2021, September). Nl2vul: Natural language to standard vulnerability score for cloud security posture management. In 2021 IEEE 14th International Conference on Cloud Computing (CLOUD) (pp. 566-571). IEEE.
2. G. Coppola, A. S. Varde and J. Shang, "Enhancing Cloud Security Posture for Ubiquitous Data Access with a Cybersecurity Framework Based Management Tool," 2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2023, pp. 0590-0594, doi: 10.1109/UEMCON59035.2023.10316003.
3. Sibi Chakkaravarthy Sethuraman, Devi Priya VS, Tarun Reddi, Mulka Sai Tharun Reddy, Muhammad Khurram Khan, "A Comprehensive Examination of Email Spoofing: Issues and Prospects for Email Security", Computers & Security, Elsevier, vol. 137, 103600, 2023.
4. D. S. Wijenayake, S. Henna and W. Farrelly, "A Graph Neural Network-based Security Posture-aware Cloud Service Provider Selection for Multi-cloud," 2023 31st Irish Conference on Artificial Intelligence and Cognitive Science (AICS), Letterkenny, Ireland, 2023, pp. 1-6, doi: 10.1109/AICS60730.2023.10470882.
5. Dantas, Y. G., Nigam, V., & Schöpp, U. (2024). A Model-Based Systems Engineering Plugin for Cloud Security Architecture Design. SN Computer Science, 5(5), 553.
6. Sibi Chakkaravarthy Sethuraman, Devi Priya, Saraju P Mohanty, "Flow based containerized honeypot approach for network traffic analysis: An empirical study", Computer Science Review, Elsevier, vol. 50, 100600, 2023.
7. S. An, A. Leung, J. B. Hong, T. Eom and J. S. Park, "Toward Automated Security Analysis and Enforcement for Cloud Computing Using Graphical Models for Security," in IEEE Access, vol. 10, pp. 75117-75134, 2022, doi: 10.1109/ACCESS.2022.3190545.
8. B. Alouffi, M. Hasnain, A. Alharbi, W. Alosaimi, H. Alyami and M. Ayaz, "A Systematic Literature Review on Cloud Computing Security: Threats and Mitigation Strategies," in IEEE Access, vol. 9, pp. 57792-57807, 2021, doi: 10.1109/ACCESS.2021.3073203.
9. K. A. Torkura, M. I. H. Sukmana, F. Cheng and C. Meinel, "CloudStrike: Chaos Engineering for Security and Resiliency in Cloud Infrastructure," in IEEE Access, vol. 8, pp. 123044-123060, 2020, doi: 10.1109/ACCESS.2020.3007338.
10. Devi Priya, Sibi Chakkaravarthy Sethuraman, Muhammad Khurram Khan, "Container Security: Precaution levels, Mitigation Strategies, and Research Perspectives", Computers & Security, Elsevier, vol. 135, 103490, 2023.
11. S. Majumdar et al., "User-Level Runtime Security Auditing for the Cloud," in IEEE Transactions on Information Forensics and Security, vol. 13, no. 5, pp. 1185-1199, May 2018, doi: 10.1109/TIFS.2017.2779444.
12. S. Feng, Z. Xiong, D. Niyato, P. Wang, S. S. Wang and S. X. Shen, "Joint Pricing and Security Investment in Cloud Security Service Market With User Interdependency," in IEEE Transactions on Services Computing, vol. 15, no. 3, pp. 1461-1472, 1 May-June 2022, doi: 10.1109/TSC.2020.2996382.
13. S. Majumdar et al., "ProSAS: Proactive Security Auditing System for Clouds," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 4, pp. 2517-2534, 1 July-Aug. 2022, doi: 10.1109/TDSC.2021.3062204.

14. Gopinath M, Sibi Chakkaravarthy Sethuraman, “A comprehensive survey on deep learning based malware detection techniques”, Computer Science Review, Vol. 47, February 2023, Elsevier.
15. A. Sahi, D. Lai and Y. Li, "A Review of the State of the Art in Privacy and Security in the eHealth Cloud," in IEEE Access, vol. 9, pp. 104127-104141, 2021, doi: 10.1109/ACCESS.2021.3098708.
16. K. Muniasamy, R. Chadha, P. Calyam and M. Sethumadhavan, "Analyzing Component Composability of Cloud Security Configurations," in IEEE Access, vol. 11, pp. 139935-139951, 2023, doi: 10.1109/ACCESS.2023.3340690.
17. A. Nhlabatsi et al., "Threat-Specific Security Risk Evaluation in the Cloud," in IEEE Transactions on Cloud Computing, vol. 9, no. 2, pp. 793-806, 1 April-June 2021, doi: 10.1109/TCC.2018.2883063.
18. Devi Priya V S, Sibi Chakkaravarthy Sethuraman, “Containerized cloud-based honeypot deception for tracking attackers”, Scientific Reports, Nature, 2023.
19. M. Ali, K. Bilal, S. U. Khan, B. Veeravalli, K. Li and A. Y. Zomaya, "DROPS: Division and Replication of Data in Cloud for Optimal Performance and Security," in IEEE Transactions on Cloud Computing, vol. 6, no. 2, pp. 303-315, 1 April-June 2018, doi: 10.1109/TCC.2015.2400460.
20. J. K. Liu, K. Liang, W. Susilo, J. Liu and Y. Xiang, "Two-Factor Data Security Protection Mechanism for Cloud Storage System," in IEEE Transactions on Computers, vol. 65, no. 6, pp. 1992-2004, 1 June 2016, doi: 10.1109/TC.2015.2462840.
21. V. Casola, A. De Benedictis, S. Di Martino, N. Mazzocca and L. L. L. Starace, "Security-Aware Deployment Optimization of Cloud–Edge Systems in Industrial IoT," in IEEE Internet of Things Journal, vol. 8, no. 16, pp. 12724-12733, 15 Aug.15, 2021, doi: 10.1109/JIOT.2020.3004732.
22. H. Xu, X. Qiu, Y. Sheng, L. Luo and Y. Xiang, "A Qos-Driven Approach to the Cloud Service Addressing Attributes of Security," in IEEE Access, vol. 6, pp. 34477-34487, 2018, doi: 10.1109/ACCESS.2018.2849594.
23. Sibi Chakkaravarthy Sethuraman, Aditya Mitra, Kuan-Ching Li, Anisha Ghosh, M Gopinath, Nitin Sukhija, "Loki: A Physical Security Key Compatible IoT Based Lock for Protecting Physical Assets", Vol. 10, Pages. 112721-112730, IEEE Access, 2023.
24. M. Y. Shakor, M. I. Khaleel, M. Safran, S. Alfarhood and M. Zhu, "Dynamic AES Encryption and Blockchain Key Management: A Novel Solution for Cloud Data Security," in IEEE Access, vol. 12, pp. 26334-26343, 2024, doi: 10.1109/ACCESS.2024.3351119.
25. A. A and K. Achuthan, "Threat Modeling and Threat Intelligence System for Cloud using Splunk," 2022 10th International Symposium on Digital Forensics and Security (ISDFS), Istanbul, Turkey, 2022, pp. 1-6, doi: 10.1109/ISDFS55398.2022.9800787.
26. A. K. Daou, F. Li and S. Shiaeles, "A Cost-Efficient Threat Intelligence Platform Powered by Crowdsourced OSINT," 2023 IEEE International Conference on Cyber Security and Resilience (CSR), Venice, Italy, 2023, pp. 48-53, doi: 10.1109/CSR57506.2023.10225008.
27. V. E. Jyothi, D. L. Sai Kumar, B. Thati, Y. Tondepu, V. K. Pratap and S. P. Praveen, "Secure Data Access Management for Cyber Threats using Artificial Intelligence," 2022 6th International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, 2022, pp. 693-697, doi: 10.1109/ICECA55336.2022.10009139.
28. S. Sibi Chakkaravarthy, D. Sangeetha, Meenalosini Vimal Cruz, V. Vaidehi and Vaidehi V, “Design of Intrusion Detection Honeypot using Social Leopard Algorithm to detect IoT ransomware attacks", IEEE Access, IEEE, vol. 8, pp.169944-169956, 2020.
29. K. Mahajan, B. Madhavidevi, B. R. Supreeth, N. V. S. SreeRathna Lakshmi, K. Joshi and S. Bavankumar, "Detecting and Responding to Cloud Security Incidents based on AI and Forensic Approach," 2023 International Conference on Innovative Computing, Intelligent

Communication and Smart Electrical Systems (ICSES), Chennai, India, 2023, pp. 1-6, doi: 10.1109/ICSES60034.2023.10465380.
30. A. Gupta, S. Sinha, H. K. Singh and B. Bhushan, "Vulnerability Assessment of Security Breach and Deadly Threat in Cloud Computing Environment," 2023 IEEE 15th International Conference on Computational Intelligence and Communication Networks (CICN), Bangkok, Thailand, 2023, pp. 433-442, doi: 10.1109/CICN59264.2023.10402297.
31. M. Chethan, Channakrishnaraju, R. Rajeswari and M. Selvam, "Cyber Attack Detection System in University Private Cloud Using Machine Learning," 2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India, 2023, pp. 1080-1085, doi: 10.1109/ICSSAS57918.2023.10331663.
32. S. An, A. Leung, J. B. Hong, T. Eom and J. S. Park, "Toward Automated Security Analysis and Enforcement for Cloud Computing Using Graphical Models for Security," in IEEE Access, vol. 10, pp. 75117-75134, 2022, doi: 10.1109/ACCESS.2022.3190545.
33. S. Han, K. Han and S. Zhang, "A Data Sharing Protocol to Minimize Security and Privacy Risks of Cloud Storage in Big Data Era," in IEEE Access, vol. 7, pp. 60290-60298, 2019, doi: 10.1109/ACCESS.2019.2914862.
34. J. He et al., "Customized Network Security for Cloud Service," in IEEE Transactions on Services Computing, vol. 13, no. 5, pp. 801-814, 1 Sept.-Oct. 2020, doi: 10.1109/TSC.2017.2725828.