International Journal of Cloud Computing and Cyber Security (IJCCCS)
ISSN: XXXX-XXXX / Volume 1 Issue 2 November-December 2026
Paper Id: JCCCS -V1I2P103/ Page No: 31-43
Doi: XX.XXXXX/XXXXXXX/ IJCCCS -V1I2P103

Original Article

A Risk-Aware Scheduling Method for Autonomous Security Tasks in Multi-Cloud Systems

Banu Priya[1], Fathima Mary[2]

[1,2]*Servite Arts & Science College for Women , Trichy.*

Received Date: 11 November 2026 Revised Date: 14 November 2026 Accepted Date: 17 November 2026

Abstract: *In multi-cloud environments, organizations can use multiple cloud service providers (CSPs) to provide their applications, data and computing resources with greater availability, flexibility and resilience. However, using various CSPs creates challenges related to securing the environment as well as the ability to detect threats, assess vulnerabilities, respond to incidents, control access and allocate resources in each of those environments. Organizations need to effectively schedule the execution of autonomous security tasks while taking into consideration the level of risk associated with each task; the urgency of each task; how much it will cost to execute each task; if there are any dependent tasks that need to be completed prior to executing each task; and the availability of the CSP services needed to execute each task. This paper presents a risk-aware scheduling mechanism for autonomous security tasks performed in multi-cloud environments. The proposed scheduling mechanism evaluates the relative priority of each security activity based upon a dynamic risk score computed from the severity of each identified threat; the criticality of each protected asset; the degree to which each asset is exposed to the identified vulnerabilities; and the possible consequences of delaying the execution of each activity. Additionally, the proposed scheduling mechanism considers each security activity's dependencies; the availability of computing resources; the unique capabilities of each CSP; and the deadline by which each activity must be completed. Ultimately, the objective of the proposed scheduling mechanism is to find an optimal balance between the effectiveness of security measures; response time to threats; computational overhead required to implement security measures; and the continuous operation of business operations. Finally, the proposed scheduling mechanism includes a feedback loop that allows organizations to update the priority of security activities as new threats emerge; as resource levels change; or as previous security activity executions produce differing results. The proposed framework establishes a basis for implementing adaptive security orchestration throughout distributed cloud infrastructure. To evaluate the effectiveness of the proposed scheduling mechanism, organizations may evaluate it using metrics such as task completion time; reduction in risk; scheduling overhead; resource utilization; and compliance with deadlines.*

Keywords: *Risk-Aware Scheduling, Autonomous Security, Multi-Cloud Systems, Security Orchestration, Dynamic Risk Assessment, Threat Prioritization, Cloud Computing, Resource Allocation, Incident Response, Adaptive Scheduling.*

I. INTRODUCTION

The use of cloud-based systems is changing the way companies deliver software applications; store data; and operate as digital service providers. As a result of the increased adoption of cloud-based systems, many organizations have implemented "multi-cloud" strategies in order to utilize multiple cloud providers' offerings to increase system availability; minimize reliance on a single vendor; support geographic distribution of systems; and fulfill unique performance or compliance needs. While multi-cloud strategies provide increased flexibility in terms of operating the business, they create significant challenges for managing the security of the systems. Since each cloud provider typically utilizes different methods for authentication/identity management; monitoring interfaces; security policies; network configurations; and incident management processes, organizations must continually coordinate the implementation of consistent security measures across each environment in order to maintain a consistent level of protection. Automated security actions such as scanning for vulnerabilities; detecting anomalous activity; verifying access control policies; analyzing threat intelligence; auditing configurations; and containing incidents can greatly decrease the requirement for continuous manual intervention.

However, the success of these automated actions depends not only on their ability to perform technical functions but also upon when, where and in what sequence they are executed. If security actions are scheduled without regard for the risks involved, then critical assets could be left vulnerable to attack while less urgent activities utilize scarce computing and network resources. Therefore, there needs to be a formalized scheduling methodology in place to allow organizations to schedule security

This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/2.0/)

operations based on current threats, changes to infrastructure, and organizational priorities.Task scheduling has traditionally focused on how fast a task is executed, what resources are available to complete the task, how much work can be completed within a given amount of time (throughput), how evenly distributed the workload will be across all resources, or if a task can meet its completion deadline. Although these factors are still relevant to task scheduling in a cloud environment, they do not always accurately reflect the urgency required to handle security-related operations. For example, a vulnerability assessment of an Internet accessible application with a critical vulnerability requires immediate attention.

On the other hand, a routine configuration audit of a non-critical development resource can wait. Likewise, if an alert indicates potential unauthorized access, an investigation should occur immediately; however, the associated security task may require a significant amount of computing power. Multi-cloud environments make task scheduling more complex since tasks may rely on output from other tasks, require specific capabilities of a particular cloud provider, or need access to data that cannot be easily transferred between regions. Resource contention, API limitations, network delays, temporary service interruptions, and policy restrictions can also influence task execution. A scheduler that uses static priorities or fixed execution sequences will often fail to adapt to changing risk levels. Conversely, assigning each alert the highest priority could cause excessive strain on security operations, raise operating expenses, and prevent other crucial tasks from being performed. Therefore, the main problem is to find a way to schedule tasks based upon security risk while preserving acceptable levels of performance, efficient utilization of resources, and operational continuity. This approach must provide clear distinctions between task importance and task urgency, consider the implications of delaying task execution, and allow for adjusting schedules as new information is received.

The authors of this paper present a risk-aware scheduling algorithm for performing automated security operations in multi-cloud environments. The algorithm dynamically prioritizes each operational task by taking into account several factors: the threat's severity, the critical nature of the assets involved, how vulnerable they are, what kind of delay might occur if the task isn't performed, the costs associated with performing the task, and any dependencies that may exist among other tasks. Instead of simply treating all security tasks identically, the scheduler selects those tasks that will help minimize risk to the organization, and then assigns those tasks to the appropriate cloud environment(s) where those tasks can be executed successfully. The scheduler also takes into consideration whether the task can be completed successfully within the specified time frame, given the provider's compatibility, access privileges, data location, and available resources. When new risks are detected by monitoring systems, when evidence is generated as a result of executing a task, or when cloud resource availability changes, the scheduler adjusts the task's priority accordingly. The ultimate goal of this research is to allow organizations to minimize their exposure to security threats without incurring undue scheduling overhead or disrupting the operations of legitimate users.

To demonstrate the potential value of the proposed method, the authors compare their dynamic prioritization algorithm to other common scheduling strategies, such as first-come-first-served scheduling, fixed-priority scheduling, and deadline-oriented scheduling. Comparisons include metrics such as average response time, weighted risk reduction, task completion rate, resource utilization, and deadline compliance. This research provides a formal structure for incorporating risk assessment into automated security orchestration across multiple heterogeneous cloud platforms. In addition, this research lays the foundation for future studies related to providing explanations for why a particular scheduling decision was made, estimating uncertainty when assessing risk, and coordinating automated security responses across a large number of systems. It should be noted that the authors consider their proposed method to be a conceptual model that needs to be validated through formal experimental testing with realistic workloads and clearly defined evaluation criteria.

II. PROPOSED RISK-AWARE SCHEDULING FRAMEWORK FOR AUTONOMOUS SECURITY TASKS IN MULTI-CLOUD SYSTEMS

A structured, collaborative method to coordinate scheduling of automated security tasks among several cloud service providers is presented in the proposed framework. Cloud resource monitoring, security event collection, risk analysis, task prioritization, execution control, and constant feedback form the basis of the architecture. Every cloud environment provides security data (e.g., system logs, network activity, access events, vulnerability reports, and configuration changes) that will be evaluated to detect potential threats and establish the urgency with respect to taking actions to address them. The risk analysis module assesses the severity of the threat, the value of the assets involved, the extent of vulnerability exposure, and the possible results of inaction. A dynamic risk score reflecting the comparative value of each task will then be assigned. Different types of security functions such as vulnerability scanning, access policy validation, threat investigation, and incident containment are considered by the framework.The risk-aware scheduler will use the risk scores that it has created to help decide when to run

each job and how much resources to give them. When jobs are identified as having a great deal of potential danger to the organization's systems (e.g., severe threats), involve very sensitive information (e.g., sensitive assets), or represent a rapidly growing area of concern for the organization (e.g., rapidly increasing exposure), then those types of jobs are given a higher priority than jobs that do not pose as much danger to the organization's systems (e.g., routine jobs). However, the scheduler does not make all of its scheduling decisions based on the job's risk score. The scheduler will also consider other factors when making scheduling decisions such as: 1) Job dependencies 2) Execution deadlines 3) Expected processing time 4) Available resources 5) Cloud service compatibility 6) Data locality Example: An incident investigation may need to complete prior to being able to contain the incident.

A vulnerability scan may be able to run in a cloud environment that has enough capacity and the proper authorizations. The primary goal of the scheduler is to minimize the amount of time that an organization is vulnerable to threats, while at the same time completing jobs within a reasonable amount of time, maintaining operational continuity, and utilizing resources efficiently. In cases where multiple jobs have similar risk scores, the scheduler can use deadline urgency and estimated execution costs as additional criteria in order to determine which job should be run first. Constraints in the organization's policies prevent jobs from being assigned to resources that are inappropriate for that job, and jobs cannot be run unless they have been authorized. By combining both risk-based prioritization and feasibility considerations, organizations are able to make better scheduling decisions than if they used either a fixed-priority or first-come-first-served method.

The last part of the dynamic risk-aware scheduling framework is a continuous monitoring and feedback system, which continually monitors task execution and updates regarding the status of the multi-cloud environment. Upon assignment of a task, the dynamic risk-aware scheduling system will record when the task completed, what was the response time for completing the task, how much resources were consumed during the execution of the task, and any observed security implications of executing the task. If a new intrusion alert is received, a vulnerability increases in priority, or a cloud provider's availability decreases, then the risk assessment module will recalculate the relevant priority levels for tasks and the dynamic risk-aware scheduling module will revise all pending assignments. Tasks that have been completed may also produce information that alters the priority or dependency of subsequent tasks. For example, if a critical vulnerability is identified as a result of a task being completed, then the identification of the critical vulnerability could result in additional investigation. Likewise, if a task confirms that a resource has been compromised, then the confirmation of the compromised resource could result in elevating the containment of the compromised resource over routine auditing. In order to avoid constantly changing priorities resulting in instability of execution, the implementation may include minimum priority thresholds, controlled rescheduling intervals, and protection mechanisms for tasks that are currently executing.



Figure 1: Framework for Risk-Aware Task Scheduling In Multi-Cloud Security

III. RISK ASSESSMENT AND DYNAMIC TASK PRIORITIZATION

The risk assessment process is a vital part of automated security scheduling within multi-cloud environments as it provides the foundation for determining what security actions need to be performed immediately. All of the cloud-based systems produce ongoing streams of data generated by intrusion detection systems (IDS), vulnerability scanners, identity management platforms, network monitoring tools and configuration audits. The proposed approach uses all of the available data sources to create an estimate of the risk related to each individual security event. Several key elements are taken into account when evaluating the

risk including threat level, value of the asset, exposure to vulnerabilities, probability of exploitability, and potential consequences of a delay in responding. For instance, an incident involving a suspected compromise of a production database that contains sensitive data would have higher priority for response than a simple alert about a configuration issue in a less critical development environment. Additionally, risk assessment must account for the reliability and currency of the input data used in order to prevent creating scheduling plans based upon old or incorrect alerts. Risk assessment also creates a standardized representation of risk across the multiple cloud platforms, allowing the scheduler to evaluate and prioritize tasks across all of the platforms, rather than simply scheduling them based upon arrival time or static task priorities.

Dynamic prioritization provides flexibility in scheduling by allowing the scheduling mechanism to adjust to changing threats as well as changes in available cloud resources. Autonomous security tasks are prioritized with a priority score that is based on several normalized factors: risks, deadlines, expected execution time, and estimated operational costs. The importance of each factor can be adjusted depending on the policy of an organization and the type of security operation. When evaluating threats that require immediate response (e.g., incidents) vs. those that are computationally expensive yet have a longer deadline (e.g., vulnerability assessments), the system will always prioritize responding to threats first. However, the system does not ignore low-priority tasks forever. As tasks age, they can increase in priority. Deadlines can help to ensure that tasks that need to be executed quickly receive adequate attention. There are also task dependencies that must be considered. Some tasks can only occur after other tasks have been completed. For example, evidence gathering would typically need to occur prior to isolating a resource if preserving evidence is critical and doing so doesn't significantly increase risk. Therefore, the scheduler must find a balance between reducing risk immediately and providing fair access to all tasks, ensuring tasks can be successfully executed, and meeting operational needs.

As well as resource allocation, another factor that will impact on the success of risk-aware scheduling is the fact that each cloud provider provides varying levels of computing power, security services, geographic location, and access controls. In evaluating whether a chosen cloud environment can complete a task in a timely manner and with the appropriate security policy, the scheduler must evaluate processing capability, network delay, API rate limiting, service availability, and data sovereignty rules. In instances where there are two or more cloud providers that provide similar capabilities, the system may choose the most appropriate provider based upon the following criteria; reduction of risk, anticipated completion date, resource usage, and expense. Moving sensitive data from one cloud provider to another to expedite security assessment may create additional exposure and compliance issues. Therefore, the scheduling process should prefer executing tasks at an authorized execution location and reduce the transfer of unnecessary data. In the event that a resource is no longer available or the task execution delay exceeds the allowed amount of delay, then the system may re-evaluate the pending tasks and determine if there are other eligible resources. These decisions should be documented in such a way so that they can be audited, including documentation regarding why a task was scheduled prior to other tasks, what constraints influenced the assignment of that task, and if the final execution results were as anticipated.

The final piece of the proposed model is "continuous monitoring" which will link the scheduling of tasks to the actual results of the security actions taken as a result of those schedules. Once a task has completed, the system will record when the task was finished, how long it took to complete, what resources were consumed, what types of security threats were detected during execution, and whether or not the security actions implemented were successful. All of this data will be used to modify current risk assessments and help inform future scheduling decisions. For example, if a vulnerability scan identifies an extremely serious flaw in one of the systems being managed, then additional investigation or remediation tasks may be given a higher priority. Conversely, if a pattern emerges where many of the alerts generated turn out to be false positive, then the risk assessment process can incorporate that knowledge into its calculations rather than automatically discarding all subsequent alerts. Evaluation of the performance of the proposed approach will involve comparing the performance of the proposed approach against that of traditional approaches to scheduling with respect to workload levels and experimental setups. Performance indicators that can be considered include: Average response time; Weighted risk reduction; Task completion percentage; Deadline meeting percentage; Resource usage; Scheduling overhead. Experimental variables that can be altered include: Threat intensity variations; Variations in the capacity of each cloud provider; Task dependency relationships; Temporary unavailability of cloud services. Although the framework provides a systematic foundation for adaptive security orchestration, its ability to function effectively in real-world scenarios must be demonstrated through empirical studies. This type of study will allow us to determine the optimal configuration parameters for our framework, recognize any shortcomings in the model, and guide the development of improved models for reliable, transparent, and scalable multi-cloud security management.

III. IMPLEMENTATION STRATEGY AND PERFORMANCE OPTIMIZATION

A. Task Classification and Risk Scoring

Task classification forms the base for efficient and risk-aware scheduling in a multi-cloud environment. Security tasks performed autonomously are classified based upon their goal, how quickly they need to be executed, what types of resources will be required to execute them, and the possible impact on security. Examples of some of the common classes of security tasks are threat detection, vulnerability assessment, access control validation, incident investigation, and threat containment. A risk score is assigned to each security task based on several elements including: the severity of the threat being investigated, the importance of the assets being protected by the task, the level of exposure to vulnerabilities (vulnerability exposure), the probability that a vulnerability can be exploited (exploitation likelihood), and the consequences of delaying execution of the security task. To enable consistent comparison of risk scores among different cloud service providers, each element of the risk score can be normalized to a standard scale. In addition, the scoring mechanism must take into consideration the uncertainty in security alerts since incomplete or inaccurate information can result in distorted priorities. An organization can determine the relative importance of each risk element in accordance with its specific security policy and operational requirements. Task classification must occur periodically to ensure that the priority assigned to a task remains current as additional evidence emerges. Task classification provides the scheduler with the ability to distinguish between urgent security operations and routine maintenance operations and allocate resources based on quantifiable security needs.

B. Scheduling Algorithm and Task Dependencies

Scheduling algorithms determine how to order the execution of security-related actions (tasks) while considering limits on resources, deadlines, and dependencies among tasks. Tasks related to high-risk threats will be prioritized, however, the algorithm must also take into account the time it takes to execute a task; the availability of cloud services; and the amount of resources needed to perform an action. A task dependency graph can be used to describe the relationships among actions. This will allow for the collection of diagnostic data prior to beginning an incident investigation. Once an incident investigation begins, there may be subsequent actions taken to contain or remediate the situation. The algorithm must ensure that incompatible tasks are not scheduled onto unauthorized or unavailable cloud resources. In cases where multiple tasks are given equal priority, other factors such as deadline urgency, estimated completion time, and resource usage can be used to establish their execution order. To avoid starvation, waiting tasks may have their priorities increased as long as they remain in the queue. Rescheduling should be minimized when task priorities are constantly changing. These techniques will help create a balanced workload; increase scheduling stability; and help provide security response capabilities within heterogeneous cloud environments.

C. Multi-Cloud Resource Allocation and Coordination

Resource allocation in multi-cloud refers to choosing a relevant cloud environment for executing an autonomous security task. There are several factors that distinguish different cloud providers including computing capacity, service monitoring, access methods, and security features. The presented approach considers all such differences prior to allocating a task to a particular resource. Examples of key considerations include processing capability, network latency, application programming interface (API) availability, execution costs, geographic location of data storage, and the company's access policy. For example, a task containing sensitive information must be run only in a validated environment regardless of whether there is a faster processing option with another provider. If there exist multiple eligible resources for a given task, then the scheduler will evaluate which resource has a lower estimated time until task completion and has lower resource requirements in order to determine where the task should be allocated. Coordination mechanisms enable security tasks to run on multiple providers simultaneously while maintaining identical priorities and tracking of executions. When a provider is down, the scheduler may reassign pending tasks to alternative eligible resources provided that the company's policies allow for this. In addition to considering the cost associated with migrating the task to a different provider and the potential disruption to ongoing operations due to the migration, when reassigning pending tasks to alternative eligible resources, the scheduler must also assess whether the new provider can provide the same level of confidentiality, availability, and compliance as the original provider. Resource coordination maximizes infrastructure utilization while providing protection against unauthorized access to confidential information, ensuring availability of systems and services, and enabling companies to meet regulatory compliance standards.

D. Continuous Monitoring and Performance Evaluation

Continuous monitoring allows the scheduling framework to adapt as the threat environment changes and as the infrastructure changes. When tasks have finished executing, the scheduling system will record when the task completed, how long it took to execute, the resources used during execution, any vulnerabilities found during execution, and the security

implications of all of the above. This data can then be utilized to modify the current risk score and determine which tasks should be executed next based on previous experience. An example of an adaptive decision would be if a new high-priority vulnerability is discovered then it would likely be given a higher priority than other tasks. On the other hand, if there was an indication of possible unauthorized access then further investigation would be required. Performance evaluation of the adaptive decision making process should focus on measuring whether the adaptive decision making process improved security while minimizing the additional processing required to make the adaptive decision. Metrics that can be used to evaluate the adaptive decision making process include but are not limited to average response times, risk reduced on average, percentage of tasks completed on average, percentage of tasks meeting deadlines on average, average resource utilization, and average overhead associated with making adaptive decisions. The adaptive decision making process proposed herein should be evaluated against static scheduling algorithms such as First Come First Served (FCFS), Fixed Priority Scheduling (FPS), and Deadline Based Scheduling (DBS) in similar environments with the same parameters. Evaluation scenarios should vary in terms of threat levels, resource contention, failure rates of providers, and dependency relationships among tasks. Experimentation with multiple trials can aid in establishing reliability of the results and identifying optimal parameter settings for each workload type. This information can then be used to assess the feasibility of implementing the scheduling framework within large scale multi-cloud environments.

V. SECURITY CHALLENGES AND RELIABILITY CONSIDERATIONS

A. Challenges in Risk-Aware Multi-Cloud Scheduling

The implementation of risk-aware scheduling for autonomous security tasks in multi-cloud environments poses numerous technical and operational issues. The first issue is the diversity of cloud providers, each using unique monitoring tools, security services, identity management systems, and methods of allocating resources. To integrate these components into a single system requires that each component communicate using a common format and provide reliable communication between each of the various security platforms located throughout the multi-cloud environment. The second issue is that there is no guarantee as to the accuracy of the risk assessment provided by the various sources (i.e. Threat Intelligence Feeds, Vulnerability Reports, Security Alerts) used to assess the current level of risk. If the risk assessment provides inaccurate data then the scheduler will prioritize less critical tasks over more important tasks that require immediate attention. Resource contention among high-priority tasks vying for the same limited computational resources creates another issue related to performance of the scheduler.

Frequent changes in the threat landscape will lead to constant rescheduling of tasks resulting in increased computational overhead and decreased stability of operations. Finally, there are both regulatory and data privacy concerns regarding how sensitive data is transmitted across different cloud providers or geographic locations. Additionally, task dependencies, execution deadlines, and provider-specific permissions add to the complexity of making decisions regarding task scheduling. To overcome these issues the proposed framework should employ techniques such as data normalization, confidence-aware risk scoring, dependency management, and policy-based resource selection. Priority aging can alleviate starvation issues while controlled rescheduling can eliminate frequent rescheduling events. The framework should also generate audit trails so that decisions concerning task prioritization and resource allocation can be explained. These mechanisms can provide consistency, transparency, and reliability; however their utility needs to be demonstrated through experimental testing with realistic workloads, varying types of heterogeneous cloud configurations, and dynamic security landscapes.

B. Reliability, Scalability, and Future Improvements

To operate successfully within large, distributed, multi-cloud environments, a risk-based security scheduling solution must possess high levels of reliability and scalability. For example, if a single cloud service becomes unavailable, a monitoring interface experiences delays, or communication among the various cloud providers fails, the system must remain functional. To minimize the probability of missing critical security functions, the system can utilize several strategies including; utilizing redundant monitoring components, using persistent task queues, tracking execution status, and employing controlled retry mechanisms. Prior to attempting to re-execute a failed function, the system's scheduler should evaluate if repeating the function is appropriate based upon its potential impact on the system, specifically with regard to isolating resources, revoking access rights, etc. Scalability can be enhanced by spreading the process of evaluating functions throughout multiple scheduling components and simultaneously executing multiple independent security functions. However, distributing functions across multiple scheduling components creates synchronization issues due to differences in observations regarding current threats and resource availability. To resolve these synchronization issues, periodically reconcile state information across all components and define centralized policies. Machine Learning can be utilized in future iterations of the system to estimate how long each function will take to complete, predict how many resources will be required by each function, and recognize trends related to new threats.

Explainable Artificial Intelligence can explain why certain functions are prioritized, allocated resources, and/or rescheduled. Reinforcement Learning can be explored as a possible approach to adaptively scheduling functions, so long as the safety constraints limit the experimentation with new policy options that delay executing critical security functions. Scalability tests, fault injection experiments, risk reduction, response time/latency, and resource overhead should be included in the evaluation of the system. Comparative experiments with traditional approaches to function scheduling can be used to determine if the enhancements described herein lead to measurable improvements. In general, a robust implementation of the system will integrate both adaptive prioritization and human intervention (oversight), access control, auditability, and well-defined recovery processes. This integration can support automated security operations in diverse cloud environments without losing control over the operational aspects of those environments.

VI. Comparative Evaluation of Risk-Aware Scheduling Methods

The proposed risk-aware scheduling technique will be compared against existing scheduling methods in the cloud. First come first served (FCFS) scheduling executes jobs based on their arrival time. Fixed priority scheduling executes jobs based on their predefined importance level. Deadline based scheduling focuses on job execution within a specified deadline. Deadline based scheduling does not always evaluate the security implications of delaying some jobs. The proposed risk-aware scheduling technique utilizes dynamic risk scores, task dependencies, resource availability and job completion deadlines to determine how best to schedule each job. Table 1 provides an overview of the differences among the various types of scheduling. This comparison provides a conceptual framework for evaluation rather than experimentally proven data. Therefore, the actual performance improvement provided by the proposed technique can only be demonstrated via controlled experiments utilizing standardized workloads and evaluation criteria.

Table 1: Risk-Aware Scheduling Methods for Autonomous Security Tasks

Scheduling Method	Risk Assessment	Task Priority	Deadline Handling	Resource Awareness	Adaptability	Main Limitation
First-Come-First-Served	None	Arrival order	Limited	Basic	Low	Critical tasks may wait
Fixed-Priority	Static	Predefined	Moderate	Basic	Low	Priorities become outdated
Deadline-Based	Limited	Deadline urgency	High	Moderate	Moderate	Security impact may be overlooked
Round-Robin	None	Equal turns	Limited	Moderate	Moderate	Urgent tasks may be delayed
Risk-Aware Scheduling	Dynamic	Risk and urgency	Integrated	High	High	Requires accurate risk estimates

The proposed method is developed to increase the speed of an organization's reaction to security incidents; it will consider both the potential impact of a given threat as well as the practicality of carrying out a particular course of action. In order to evaluate the effectiveness of the proposed method, we should measure average reaction times, weighted risk reductions, completion rates for all tasks, compliance with deadlines for completing tasks, utilization of available resources, and scheduling overhead. We should conduct experiments with various cloud providers, varying levels of risk associated with each incident, resource limitations, and dependency relationships among tasks. These experiments can show us if there are advantages in using dynamic risk prioritization compared to traditional approaches and if there are specific conditions under which additional optimizations may be required.

VII. EXPERIMENTAL DESIGN AND PERFORMANCE ANALYSIS

A. Experimental Environment and Workload Configuration

An experimental environment that represents a multi-cloud heterogeneous infrastructure (i.e., multiple cloud providers) and includes virtual machines, containerized applications and distributed security monitoring services will be required to evaluate the effectiveness of the proposed risk-aware scheduling approach. All environments used during the evaluation process must have varying processing capabilities, network performance, service availability levels, and security characteristics so that they closely mimic real world operating environments. The workload that will be evaluated using the proposed approach must contain several types of autonomous security tasks including but not limited to vulnerability scanning, intrusion investigation, access control verification, configuration auditing, and incident containment. Attributes associated with each task must include risk severity, asset criticality, execution duration, resource requirements, deadline, and dependencies.

Threat scenarios can be created to assess the performance of the proposed scheduling approach under normal operating conditions, higher levels of attack activity, resource contention, and temporary cloud service outages. To compare the performance of the proposed scheduling approach against other scheduling approaches such as first-come-first-served (FCFS), fixed-priority (FP), and deadline-based (DB), all evaluations must utilize the same task workload and resource limitations. Multiple evaluations utilizing different workload sizes will assist in assessing the repeatability and scalability of the results obtained from the evaluation. In addition, evaluations conducted using simulation methods must be clearly distinguished from evaluations performed using measurements collected from an actual implementation of the proposed scheduling approach. Additionally, all relevant experimental parameters (i.e., risk score weights, scheduling intervals, task arrival distributions) must be documented in order to facilitate reproducibility. This structure provides a baseline for evaluating whether risk-aware scheduling can enhance coordination among security tasks while avoiding additional computational overhead and/or disruption to legitimate cloud workloads.

B. Performance Metrics and Evaluation Procedures

A performance evaluation process should assess an organization's security performance, scheduling performance, and operations reliability through multiple measurements, as opposed to only evaluating one aspect of an organization's performance. Average response time is a measure of the amount of time that passes from when an organization identifies a security issue until it takes action to resolve the issue. Task completion time is a measure of the amount of time that passes from when an organization begins working on resolving a security issue until it completes the process. Weighted risk reduction is a measure of the amount of risk that an organization reduces as a result of completing the tasks that have the highest priority based upon the risks that those tasks represent. Deadline compliance is a measure of the percentage of tasks that an organization completes on time. Resource utilization is a measure of whether an organization is using its available processing power effectively. Scheduling overhead is a measure of the amount of processing power that an organization uses to assign priorities to tasks and to assign resources to those tasks.

Task completion rate and failure rate are also measures of an organization's reliability. Organizations should use the same set of performance metrics for all baselines and for their proposed scheduling approach under similar experimental conditions. When reporting results, organizations should include multiple trials, averages, and variability measurements (i.e., standard deviations or confidence intervals). Organizations may conduct sensitivity analyses to determine how modifications to risk factor weights, workload intensities, and resource availability impact performance. Organizations may conduct ablation experiments to eliminate various components of their scheduling system (i.e., deadline adjustment or task-aging) in order to determine what contribution each component provides. Organizations should not claim any improvements in performance until after conducting experimental studies that demonstrate those improvements. An evaluation process that utilizes this methodology provides organizations with an objective basis for comparing scheduling approaches and allows organizations to make trade-offs between reducing risk, executing tasks quickly, efficiently utilizing resources, and achieving stable scheduling.

C. Results Interpretation and Optimization Opportunities

The findings should be evaluated to determine whether the proposed scheduler has consistently assigned resources to high-risk security tasks while providing adequate performance in varying workloads. Lower average response times for critical tasks might imply improved responsiveness; however, this should be evaluated in conjunction with total task completion rates and compliance with deadlines. In addition, increased resource utilization is only positive if it does not lead to excessive contention or delays in completing essential security functions. If the proposed mechanism reduces weighted security risks, yet creates

significant scheduling overhead, then the risk-scoring process may need to be simplified or made more efficient. If low-priority tasks are subjected to excessively long waiting periods, then the task-aging mechanisms and fairness constraints can be modified. If there is an unavailability of cloud resources, then fallback assignment policies may improve resiliency as long as data residency and access restrictions continue to be met. Additionally, further optimization can investigate incremental risk-score updates, parallel task evaluations, caching provider capabilities, and selectively rescheduling tasks affected by failures rather than recalculating the entire schedule. Explainability mechanisms can also be developed to provide documentation regarding why certain scheduling decisions were made. This will assist administrators in verifying automated decisions and identifying inappropriate priorities. Future evaluations will incorporate larger task volumes, unpredictable threat arrival patterns, and failure rates from multiple types of providers. Based upon the evaluations conducted above, the framework can be refined to achieve optimal trade-offs among security effectiveness, scalability, resource consumption, and operational continuity. Conclusions reached in the end will be based upon actual measured data and clearly identified limitations associated with the findings.

VIII. PRACTICAL APPLICATIONS AND DEPLOYMENT CONSIDERATIONS

Autonomous security task scheduling is very important for organizations that operate on more than one cloud platform. For example, financial institutions can use the proposed risk-aware scheduling to decide which tasks (like investigating suspicious transactions, verifying identities, performing vulnerability assessments, and containing incidents) need to be done first based on how much money could be lost if something went wrong, and how private the data involved is. Healthcare organizations can use risk-based scheduling to keep patient records safe, keep track of who accesses them without permission, and coordinate security assessments among all of their cloud-hosted apps. E-commerce businesses can use the framework to decide which security tasks related to their payment gateway, customer database, authentication service, or online transaction system are most urgent. Additionally, government agencies and large corporations can benefit from coordinating security monitoring across multiple cloud infrastructures in different locations. Regardless of what type of organization uses it, the framework will allow security teams to prioritize tasks based upon the value of protected assets, the severity of potential threats, and the possible negative outcomes of taking too long to respond. Because the framework allows organizations to adjust their task priorities as needed, security teams can focus their limited computing resources on tasks that may have the greatest impact on protecting the organization's assets. However, organizations must properly integrate the framework with their existing monitoring tools, identity management platforms, and incident response processes. Organizations should also verify the expected benefits of using the framework through testing specifically tailored to each organization's unique needs prior to implementing the framework into their day-to-day operations.

Successful deployment is achieved through a phased-in approach that reduces disruptions to operations and maintains current security practices. Organizations must begin by integrating monitoring data from specific cloud providers; establishing a uniform manner for representing security activities, resource requirements, and risk factors; operating the scheduler in a "recommendation" mode so that security administrators may review proposed priorities prior to automatic execution; comparing scheduled recommendations against existing operational practices; identifying errors in risk assessments or policy conflicts; introducing automation for low-risk, reversible activities (e.g., routine configuration checks, selected vulnerability assessments); requiring higher levels of authorization/approval for high-impact activities (e.g., account suspension, network isolation, disruption of production workloads) in accordance with organizational policies; incorporating access controls, encrypted communication channels, audit logging mechanisms, fail-safe recovery procedures, and periodic reviews of scheduling performance into the deployment process; ensuring that automation is accountable; and executing security-related activities only within acceptable operating parameters. Integration testing must address issues such as provider outages, API failures, delayed telemetry data, and unexpected task dependencies.

Long-term effectiveness requires continuous evaluation of scheduling parameters, maintenance of the policies, and adaptation to the evolving nature of cloud environments. As threat patterns, application architectures, provider capabilities, and regulatory requirements change over time, some previously effective scheduling parameters may no longer be appropriate. Organizations should regularly review their risk-scoring weights, task deadlines, resource allocation policies, and escalation thresholds based upon observations of actual operational performance. Metrics such as critical-task response time, weighted risk reduction, task completion rate, deadline compliance, and scheduling overhead can indicate whether the framework is achieving its goals. Additionally, administrators should assess the number of false alarms generated, the frequency of unnecessary task reassignment, and any delays impacting lower-priority activities. If machine learning is utilized to estimate risk, then machine learning models should be assessed for reliability, bias, and susceptibility to changes in data distribution. Documenting why a

particular task received priority and verifying that the execution of that task complied with relevant policies can assist auditors in understanding how decisions were made regarding tasks. Ultimately, the proposed methodology provides a structured framework for managing automated security functions across multi-cloud environments. The success of this methodology will depend on whether there are demonstrable enhancements in performance, successful integration into existing processes, transparency in decision-making, and appropriate human intervention in automated processes.

IX. RISK-AWARE SCHEDULING DECISION ANALYSIS

In order to better prioritize security actions across multiple clouds, the proposed risk aware scheduling methodology uses many different decision criteria to evaluate each autonomous security action. Examples of some of the criteria are: severity of threats, importance of assets, vulnerability exposure, deadlines, costs, resource availability, dependencies among actions, and expected risk reduction. Using all of these criteria, the scheduler can differentiate between urgent security actions and typical ones and schedule security actions on appropriate cloud resources. The following table provides examples of how the above-mentioned criteria affect scheduling decisions and provide an example of their intended operational role. All examples are conceptual; no experimental measurements were taken for any of the conceptual examples.

Table 2: Threat-Based Prioritization of Autonomous Security Tasks

Task Type	Threat Severity	Asset Criticality	Deadline Urgency	Resource Demand	Task Dependency	Scheduling Priority
Intrusion containment	Critical	High	Immediate	High	Investigation may be required	Very High
Vulnerability scanning	High	High	Urgent	Medium	Asset discovery	High
Access validation	Medium	High	Moderate	Low	Identity data available	Medium–High
Configuration auditing	Low	Medium	Flexible	Medium	Configuration inventory	Medium
Routine security reporting	Low	Low	Flexible	Low	Task results available	Low

Scheduling decisions should not rely on a single element. A task that presents an immediate danger to the organization may need to be addressed right away; however, a scan of a system with high resource consumption can be done as soon as there is enough available processing power to do so without creating unacceptable risks. In addition, scheduling decisions must respect dependency relationships between tasks and assign resources to tasks based upon what each service provider allows, what authorizations are needed to perform each service, and what the data residency requirements are for each service. Also, the scheduling system must re-evaluate all pending tasks after receiving updated threat information or execution data which will affect how risky each task appears. Performance can be evaluated through several metrics including: response time; risk reduction; deadline compliance; resource utilization; and scheduling overhead. By collecting these metrics, organizations can evaluate whether the proposed scheduling methodology successfully balances the urgency of security concerns with the need for efficient allocation of resources and consistent performance in multi-cloud environments.

X. KEY BENEFITS, LIMITATIONS, AND FUTURE ENHANCEMENTS

The proposed risk-aware scheduling technique provides a systematic method of managing autonomous security operations within diverse multi-cloud environments. The technique utilizes risk-assessment, task prioritization, resource allocation, and continuous monitoring to provide improved security response capabilities with minimum disruption to operations. This paper presents an overview of the practical advantages, existing constraints, and potential areas for further development of the proposed technique.

- **Effective utilization of resources:** The scheduler assigns tasks based on the number of resources available for computation, the capability of providers to deliver those resources, the cost associated with delivering the resources, and how much of each resource is currently unused. As a result of this approach, unnecessary use of resources may be reduced and coordination among different types of cloud computing environments can be enhanced.

- **Improved coordination of responses:** The order in which security activity will occur is based upon the dependencies among tasks and their respective deadlines. This allows for improved coordination of investigations, vulnerability assessments, containment and remediation.
- **Adaptive scheduling decisions:** Continuous monitoring provides the scheduler with the ability to update pending tasks as a result of new alerts, vulnerabilities identified, and/or changes to the environment that may alter their priority.
- **Operational Transparency:** The scheduling records provide documentation of the criteria that were used to select tasks for completion, assign resources, and determine the sequence in which those tasks will be completed. This documentation supports audit activities, provides a basis for problem-solving/troubleshooting, and allows for oversight of administrative functions.
- **Uncertainty of Risk Assessment:** The presence of incomplete telemetry, inaccurate threat intelligence and/or false positives can lead to misleading risk scores and inappropriate task prioritization.
- **Multi-cloud integration is complex:** There are differences among cloud providers' API structures, identity management systems, monitoring tools, and security controls that can create difficulty when attempting to maintain a consistent schedule and coordinate tasks.
- **Excessive use of computing power:** If you are continually calculating risk levels, or constantly changing the schedule for your security team, it can use up a lot more of the computers' processing capability than necessary, especially if you have a lot of security issues that need to be addressed at once.
- **Policies and compliance restrictions:** Restrictions on the physical location of data, restrictions on who can have access to data, and/or organizational security requirements can restrict the assignment of tasks to appropriate cloud-based resources.
- **Limited experimental validation:** A framework cannot be proven effective simply by being designed well. It requires experimental control and/or real-world deployment evaluation to provide evidence of quantifiable benefits.
- **Future enhancements:** A risk prediction model based on machine learning can predict how the level of threat is changing as well as how long it takes to complete a task. This can be done by looking at previous security incidents and operational data.
- **Transparent scheduling**: Decisions can be explained by transparent mechanisms as to why some tasks have a higher priority than others and which factors contribute to selecting resources.
- **Scalability for Distributed Orchestration:** To maintain consistent policy enforcement and reliable task coordination, distributed scheduling components can handle large volumes of work.
- **Cloud-based automated recovery systems:** Detection of failure, safe retry attempts, and policy compliant reallocation may enhance resiliency if cloud based service(s) go off line.

XI. CONCLUSION

The increased use of multi-cloud computing has made the security landscape more complex. Organizations must coordinate their security efforts among different cloud providers, multiple applications, and across geographical boundaries. Many traditional scheduling approaches were designed around task arrival times, preset priority levels, resource availability, and/or execution deadlines. However, many traditional scheduling approaches do not consider how urgent security threats can change over time. Therefore, we developed a dynamic risk-aware scheduling model for autonomous security tasks in multi-cloud environments to overcome the limitations of traditional scheduling models. Our model combines dynamic risk assessments of security threats, task prioritization, resource allocations, management of task dependencies, and continuous monitoring into one unified scheduling model. Risk levels are established based upon various factors such as threat severity, asset criticality, vulnerability exposure, exploitation likelihood, and potential impacts due to delays in execution. These risk levels allow us to determine which security tasks should be performed immediately versus those that are routine maintenance. We have also included additional considerations within our model such as provider capability, execution deadlines, resource constraints, and organizational policies to create a structured way to schedule security tasks across multiple cloud environments. Our overall goal is to enhance responsiveness to security threats while minimizing unnecessary computational overhead, disruption to legitimate workloads, or disruptions to business operations.

The proposed method also highlights the necessity of having an adaptive decision-making process and a well-managed execution. Due to the rapid changes in security conditions of cloud infrastructures (e.g., discovery of new vulnerabilities, detection of suspicious behavior, fluctuation of service availability), a static priority-based scheduling scheme may fail to be effective as the operational environment changes. On the other hand, the proposed approach continuously monitors and feeds

back information to adjust task priorities based on new data. Task dependencies ensure that all prerequisites for a task are completed before that task starts executing. Resource-aware assignment of resources allows us to find appropriate execution environments for each task. Other techniques such as priority aging, controlled rescheduling, audit logs and policy-based resource selection will provide fair treatment of tasks, transparency of operations and improved reliability. This model can be applied to many different types of security functions such as intrusion investigation, vulnerability assessment, access control verification, configuration auditing and incident containment. However, the proposed model is a conceptual research idea and not a fully validated production system. Therefore, we do not know if this method will actually be effective without conducting empirical evaluations. Future experiments should evaluate this method against other scheduling methods such as first-come first-served, fixed-priority and deadline-based scheduling under similar conditions (workload, resource constraints and threat scenarios) to determine if this method produces measurable improvements.

The next phase of this research is expected to include: 1) improving the accuracy of risk estimates; 2) enhancing the scale of the framework; 3) testing the framework in real-world multi-cloud environments. Additionally, machine learning could be used to predict the severity of threats; determine how long tasks take to execute; and identify changes in workloads. Explainable Artificial Intelligence (XAI) could explain why tasks were prioritized over others and why resources were assigned to certain tasks. Distributed scheduling models may be able to support much larger systems; however, these models will require synchronization of all parts of the system; consistency in security policies; and appropriate methods for recovering from failures. Further studies should explore what happens when there are false positives in alerts; when there is incomplete monitoring data; when a provider goes down; when there is competition among resources; and when the nature of an attack changes very quickly. Security constraints such as privacy protection, data residency, authorization boundaries, and regulatory compliance will always apply during deployment. Human review of high risk actions, especially those that disrupt production workloads, suspend accounts or isolate networks, should continue to occur. Results from experiments should demonstrate when risk aware scheduling results in improved security outcomes and what the trade-offs are between responding quickly, using resources efficiently, and keeping operations stable. In summary, the risk-aware scheduling model developed here provides a structured approach to managing autonomous security tasks within multi-cloud environments. Integrating security risk with scheduling decisions allows for more reactive, adaptive, and accountable cloud security management. Ultimately, the success of the model will depend upon its rigorous evaluation, practical deployment, and continued development as multi-cloud environments and cybersecurity threats continue to evolve.

XII. REFERENCES

[1] Armbrust, M. et al. (2010) 'A view of cloud computing', Communications of the ACM, 53(4), pp. 50–58.

[2] Beloglazov, A., Abawajy, J. and Buyya, R. (2012) 'Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing', Future Generation Computer Systems, 28(5), pp. 755–768.

[3] Buczak, A.L. and Guven, E. (2016) 'A survey of data mining and machine learning methods for cyber security intrusion detection', IEEE Communications Surveys & Tutorials, 18(2), pp. 1153–1176.

[4] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J. and Brandic, I. (2009) 'Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility', Future Generation Computer Systems, 25(6), pp. 599–616.

[5] Cai, N. and He, G. (2024) 'Multi-cloud resource scheduling intelligent system with endogenous security', Electronic Research Archive, 32(2), pp. 1380–1405.

[6] Chandola, V., Banerjee, A. and Kumar, V. (2009) 'Anomaly detection: A survey', ACM Computing Surveys, 41(3), Article 15.

[7] Dinh, H.T., Lee, C., Niyato, D. and Wang, P. (2013) 'A survey of mobile cloud computing: Architecture, applications, and approaches', Wireless Communications and Mobile Computing, 13(18), pp. 1587–1611.

[8] ENISA (2009) Cloud Computing: Benefits, Risks and Recommendations for Information Security. Heraklion: European Network and Information Security Agency.

[9] Foster, I., Zhao, Y., Raicu, I. and Lu, S. (2008) 'Cloud computing and grid computing 360-degree compared', in Proceedings of the Grid Computing Environments Workshop. Austin, TX, pp. 1–10.

[10] Grozev, N. and Buyya, R. (2014) 'Inter-connected cloud computing environments: Challenges, taxonomy, and survey', ACM Computing Surveys, 47(1), Article 6.

[11] Hashizume, K., Rosado, D.G., Fernández-Medina, E. and Fernandez, E.B. (2013) 'An analysis of security issues for cloud computing', Journal of Internet Services and Applications, 4, Article 5.

[12] Khraisat, A., Gondal, I., Vamplew, P. and Kamruzzaman, J. (2019) 'Survey of intrusion detection systems: Techniques, datasets and challenges', Cybersecurity, 2, Article 20.

[13] Lundberg, S.M. and Lee, S.-I. (2017) 'A unified approach to interpreting model predictions', in Advances in Neural Information Processing Systems 30. Red Hook, NY: Curran Associates, pp. 4765–4774.

[14] Mell, P. and Grance, T. (2011) The NIST Definition of Cloud Computing. NIST Special Publication 800-145. Gaithersburg, MD: National Institute of Standards and Technology.
[15] Moustafa, N. and Slay, J. (2015) 'UNSW-NB15: A comprehensive data set for network intrusion detection systems', in Proceedings of the Military Communications and Information Systems Conference (MilCIS). Canberra, Australia, pp. 1–6.
[16] Nikose, B. and Borkar, G.M. (2025) 'Toward efficient and secure task scheduling using trust optimizer multi-cloud scheduler', Security and Privacy, 8(6), Article e70104.
[17] NIST (2012) Computer Security Incident Handling Guide. NIST Special Publication 800-61 Revision 2. Gaithersburg, MD: National Institute of Standards and Technology.
[18] NIST (2024) The NIST Cybersecurity Framework (CSF) 2.0. NIST Cybersecurity White Paper 29. Gaithersburg, MD: National Institute of Standards and Technology.
[19] Petcu, D. (2013) 'Multi-cloud: Expectations and current approaches', in Proceedings of the International Workshop on Multi-Cloud Applications and Federated Clouds. New York: ACM, pp. 1–6.
[20] Ribeiro, M.T., Singh, S. and Guestrin, C. (2016) '"Why should I trust you?" Explaining the predictions of any classifier', in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, pp. 1135–1144.
[21] Roman, R., Lopez, J. and Mambo, M. (2018) 'Mobile edge computing, fog et computing and the Internet of Things: A survey', Future Generation Computer Systems, 78, pp. 680–698.
[22] Saleh Al Shamaa, S. and Shi, W. (2026) 'A systematic review of security-aware task scheduling in cloud, fog, and edge computing', Journal of Cloud Computing. Published online 15 July.
[23] Scarfone, K. and Mell, P. (2007) Guide to Intrusion Detection and Prevention Systems (IDPS). NIST Special Publication 800-94. Gaithersburg, MD: National Institute of Standards and Technology.
[24] Sommer, R. and Paxson, V. (2010) 'Outside the closed world: On using machine learning for network intrusion detection', in Proceedings of the IEEE Symposium on Security and Privacy. Oakland, CA: IEEE, pp. 305–316.
[25] Subashini, S. and Kavitha, V. (2011) 'A survey on security issues in service delivery models of cloud computing', Journal of Network and Computer Applications, 34(1), pp. 1–11.
[26] Toosi, A.N., Calheiros, R.N. and Buyya, R. (2014) 'Interconnected cloud computing environments: Challenges, taxonomy, and survey', ACM Computing Surveys, 47(1), Article 7.
[27] Zhang, Q., Cheng, L. and Boutaba, R. (2010) 'Cloud computing: State-of-the-art and research challenges', Journal of Internet Services and Applications, 1, pp. 7–18.
[28] Sharafaldin, I., Lashkari, A.H. and Ghorbani, A.A. (2018) 'Toward generating a new intrusion detection dataset and intrusion traffic characterization', in Proceedings of the 4th International Conference on Information Systems Security and Privacy. Setúbal: SCITEPRESS, pp. 108–116.
[29] Gunning, D. and Aha, D. (2019) 'DARPA's explainable artificial intelligence (XAI) program', AI Magazine, 40(2), pp. 44–58.
[30] Garcia-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G. and Vázquez, E. (2009) 'Anomaly-based network intrusion detection: Techniques, systems and challenges', Computers & Security, 28(1–2), pp. 18–28.