Magna Scientia Advanced Research and Reviews

eISSN: 2582-9394

Cross Ref DOI: 10.30574/msarr

Journal homepage: https://msarr.magnascientiapub.com/

(RESEARCH ARTICLE)

AUTOMATED SECURITY ORCHESTRATION IN CLOUD INFRASTRUCTURE: A FRAMEWORK FOR REAL-TIME THREAT DETECTION AND RESPONSE

Abhilash Koka *

Independent Researcher, USA.
* Corresponding Author

Magna Scientia Advanced Research and Review, 2022, 05(01), 090–097

Publication history: Received on 10 April 2022; revised on 22 June 2022; accepted on 28 June 2022

Article DOI: https://doi.org/10.30574/msarr.2022.5.1.0042

ABSTRACT

Modern enterprise computing now relies on cloud infrastructures that demonstrate both a growing and dynamic attack surface due to its multi-tenant and application programming interface (API)-driven characteristics and exposes organizations to a growing attack surface, which cannot be effectively defended by traditional, manually operated Security Operations Centers (SOCs). The paper presents an Automated Security Orchestration Framework (ASOF), a stacked architecture that combines real-time telemetry with the collection, hybrid machine-learning-based anomaly detection, threat-intelligence enhancement, and automated response based on playbooks into a single closed system adapted to both the public and hybrid cloud environment. The framework integrates a lightweight ingestion and normalization layer with a correlation engine that combines supervised classification with unsupervised anomaly scoring to operated a core of decision-logic orchestration that autonomously chooses and implements remediation playbooks, including identity revocation, network isolation and configuration rollback. ASOF was compared to a traditional manual SOC workflow through a proof-of-concept testbed on a simulated multi-cloud environment in five common cloud threat categories. The findings suggest that there are significant decreases in the mean time to detect and mean time to respond, as well as a reduced false-positive rate and decreased workload on analysts, implying that automation by orchestration can significantly reduce the window of exposure in cloud environments without a corresponding increase in operational cost.

Keywords: Cloud Security; Security Orchestration, Automation and Response (SOAR); Real-Time Threat Detection; Incident Response Automation; Cloud Infrastructure Security

1 INTRODUCTION

The shift to enterprise workloads in the public and hybrid cloud environments has fundamentally altered the form of the corporate attack surface. Infrastructure that previously existed behind a few well-articulated network boundaries is now distributed on elastic, API-addressable services, ephemeral containers and identity-based access-control planes which can be reconfigured within a few seconds. The benefits of this elasticity are well documented and the benefits are numerous, however, the assets, credentials and configuration states that are continually monitored by a security team also multiply. Improperly configured storage buckets, excessively privileged identity and access management (IAM) identities, and publicly accessible application programming interfaces have become common root causes of massive

cloud breaches, and the rate at which attackers can spin within a compromised cloud tenant often vastly outstrips the rate at which a human attack analyst can triage an alert.

Conventional Security Operations Centers use manual triage: alerts sent by a Security Information and Event Management (SIEM) system are sent to a human operator, who should triage the alert by correlating with contextual data, determine severity, and manually trigger remediation measures like revoking a credential or putting a workload under quarantine. There are three compounding flaws of this model in a cloud context. First, alert volumes in the cloud environments are significantly greater in comparison to traditional data centers due to the number of services managed there and their fine-grained logging. Second, there is chronic lack of proficient security analysts, which implies that the alert queues often out-perform the available workforce, resulting in alert fatigue and false alarms. Third, the sequential process of investigation, decision, and action is not automated in the order, and thus it does create latency where the speed is the most critical to the attack lifecycle: containment.

Security Orchestration, Automation and Response (SOAR) platforms have been developed in the industry to respond to these pressures and have attempted to bring together heterogeneous security tools with orchestrated playbooks capable of performing routine investigation and remediation actions without requiring a human operator to intervene at each step. Much of the literature available, however, either conducts a high-level survey of the concepts of SOAR, or addresses a single sub-problem, like intrusion detection, or assesses orchestration in non-cloud or IoT contexts. A gap exists to a framework that is explicitly cloud-native, closes the loop between detection and automated response, and is measured against realistic latency-based metrics that are important to cloud operators.

This paper fills this gap by presenting the Automated Security Orchestration Framework (ASOF). The contributions of this work, in particular, are the following:

- A cloud-agnostic, layered design where telemetry ingestion, hybrid machine-learning detection, threat-intelligence enrichment, and automated execution of playbooks are combined into one closed-loop pipeline.
- A risk-scored orchestration and decision-logic core which chooses between graduated response playbooks, with a human-in-the-loop override on high-impact actions.

An experimental comparison of a proof-of-concept multi-cloud testbed of ASOF versus a traditional manual SOC workflow of five typical categories of cloud threats, evaluated using detection and response latency, false-positive rate, and workload of an analyst as the metrics.

The rest of this paper will be structured as follows. Section 2 is a review of related literature on security orchestration and cloud security. Section 3 shows the proposed ASOF architecture. Section 4 explains the implementation and experimental design. Section 5 contains the results and their discussion, and the paper is concluded and directions of future work are mentioned in Section 6.

2 RELATED WORK

Initial research on cloud security mostly enumerated the threats posed by multi-tenancy, virtualization, and shared responsibility schemes. Mell and Grance [1] defined the basic nature of cloud computing and gave the taxonomy which was further applied to identify vulnerabilities like data breaches, insecure application programming interfaces, and account hijacking in future surveys on security like the one conducted by Modi et al. [4] and Subramanian and Jeyaraj [11]. These surveys define the problem space without going to the extent of suggesting a working mechanism of continuous and automated defense.

A second body of literature has studied orchestration as a means of coordination of heterogeneous security tools. Feitosa et al. [3] suggested an orchestration strategy to detect unwanted internet traffic, whereas Koyama et al. [7] outlined orchestration of a global threat-intelligence platform to associate indicators of compromise with a base of customers of a service provider. Chauhan et al. [10] followed this with a reference architecture to provision security tools as a service, and Luo and Salem [9] investigated the coordination of software-defined security services on the network layer. These experiments prove that orchestration can decrease the tool fragmentation but in general they consider detection and response as different issues instead of a system with closed loops.

Detection-based studies have been concerned with the accuracy and scalability of anomaly and intrusion detection in the virtualized and cloud environment. Chung et al. [5] suggested NICE, an attack-graph-based countermeasure-selection tool of virtual networks and Alshamrani et al. [12] provided a survey of detection techniques in the domain of advanced persistent threat through the cyber-attack lifecycle. In [14], Sun et al. conducted a review of data-driven methods of cybersecurity incident prediction and emphasized the increasing importance of machine learning in predicting attacks before they are fully achieved. Although these works make significant progress in enhancing

detection accuracy, most of them end at the alert generation stage and fail to indicate how a detected threat is to be converted into an automated remediation measure.

Security Orchestration, Automation and Response platforms are a category of the most directly related body of work. A thorough multivocal study of the literature on security orchestration carried out by Islam, Babar and Nepal [6] revealed interoperability, interpretability, and automation as the three critical challenges to the adoption of SOAR. Kinyua and Awuah [15] surveyed the increased application of artificial intelligence and machine learning in SOAR pipelines, and Zheng et al. [13] suggested an orchestration model to fit Internet of Things security automation. As industry studies, like the SOAR solutions market guide by Gartner [16] and the list of highest-ranked cloud threats presented by the Cloud Security Alliance [17] can support, the high demand among practitioners in orchestration platforms that decrease the workload of analysts. The process of incident-handling itself is standardized in guidances like NIST Special Publication 800-61 [2], which outlines the detection, containment, eradication, and recovery lifecycle that any and all orchestration framework must eventually automate.

Table 1 summarizes the representative studies in relation to the area of focus, methodological approach and major limitation of each of the studies and places the proposed ASOF in relation to this literature. As the table shows, current literature is inclined to discuss either the survey/taxonomy layer, or the orchestration-of-tools layer, or the detection layer separately. A relatively understudied area is a single, empirically tested, closed-loop architecture encompassing both telemetry gathering and automated, risk-rated remediation in the context of elastic multi-cloud infrastructure. ASOF aims to address this gap.

Table 1: Comparative summary of representative studies on cloud security and security orchestration.

Study (Year)	Focus Area	Approach	Key Limitation
Feitosa et al. (2012)	Traffic-level orchestration	Rule-based orchestration of unwanted traffic identification	Narrow scope; no cloud-native integration
Modi et al. (2013)	Cloud security survey	Taxonomy of cloud security issues and countermeasures	Descriptive; no automation or orchestration layer proposed
Chung et al. (2013)	Virtual network intrusion detection	Attack-graph-based countermeasure selection (NICE)	High computational overhead at scale; limited response automation
Chauhan et al. (2017)	Tools-as-a-service architecture	Reference architecture and meta-model for tool provisioning	Focused on provisioning, not real-time threat response
Islam et al. (2019)	Security orchestration	Multivocal literature review of SOAR platforms	Synthesises literature; lacks empirical validation on live cloud systems
Alshamrani et al. (2019)	Advanced persistent threats	Survey of APT detection techniques across the kill chain	Detection-oriented only; no closed-loop response mechanism
Zheng et al. (2020)	IoT security automation	Orchestration model for IoT security automation	Designed for IoT edge context, not elastic multi-cloud IaaS/PaaS
Kinyua & Awuah (2021)	AI/ML in SOAR	Review of machine-learning techniques applied to SOAR	Conceptual review; limited discussion of orchestration scalability
Proposed ASOF (this study)	Cloud-native orchestration	Closed-loop ML detection with automated, risk-scored playbook execution	Requires labelled training data and periodic playbook governance review

3 PROPOSED FRAMEWORK: THE AUTOMATED SECURITY ORCHESTRATION FRAMEWORK (ASOF)

ASOF has the structure of a seven block closed loop pipeline as shown in Figure 1. The design idea is to reduce the time interval between when a malicious event happens in the cloud environment and when a suitable, commensurate remediation activity is finished, without compromising auditability and maintaining the ability to override high-impact decisions by humans.

3.1 Telemetry Collection

The framework starts with on-going consumption of cloud telemetry, such as identity and access management (IAM) event logs, virtual private cloud flow logs, API call histories, container runtime logs, and storage-access logs. Since cloud providers make this telemetry visible via heterogeneous, provider-specific schemas, ASOF views telemetry collection as a pluggable adapter layer, where new log sources can be added without altering downstream components.

3.2 Ingestion and Normalization

Raw telemetry is transformed into a common event schema by the ingestion and normalization engine. This layer does field mapping, time stamp alignment, deduplication and enrichment with static asset metadata (such as resource tags, ownership and criticality classification) to create a normalized event stream which is not dependent on the cloud provider of origin.

3.3 Correlation and Machine-Learning Detection

A hybrid correlation and detection engine, which uses a supervised classifier trained on labelled examples of known attack techniques, and an unsupervised anomaly-scoring model that raises flags on statistically unusual sequence of actions, is fed normalized events, even in the absence of a labelled signature. The results of both models are combined into one confidence score per candidate incident, which helps reduce the tendency of purely signature-based systems to overlook new attack variants and reduces the volume of false-positives common to purely unsupervised techniques.

3.4 Threat Intelligence and Risk Scoring

Incidents involving candidates are enriched with external and internal threat-intelligence feeds, such as known indicator-of-compromise lists, organization-specific watchlists, and assigned a composite risk score that considers detection confidence, asset criticality, and impact of the affected identity or resource based on its size. This risk score is the main input to the decision logic of orchestration engine.

3.5 Security Orchestration Engine

The heart of ASOF decision making is the orchestration engine. It projects risk-scored incidents onto a library of predefined playbooks using a graduated response policy: low-risk anomalies invoke passive monitoring and notification to an analyst only; medium-risk anomalies invoke semi-automated actions that require a human analyst confirmation; and high-confidence, high-risk anomalies invoke fully automated playbooks like temporary credential suspension, network-segment isolation, or automatic rollback of a misconfigured resource to its last known-good state. It is a graduated structure aimed at trade offing response speed with operational risk of an erroneous automated action.

3.6 Automated Response and Feedback

The playbooks are implemented using cloud-provider application programming interfaces, and all actions are logged out to an immutable execution log, which drives both the alerting and ticketing system with human analysts and a feedback channel into the detection layer. This feedback loop enables the analyst dispositions on borderline cases to be provided as further training signal and then the risk-scoring thresholds can be recalibrated and the risk-scoring models retrained periodically as the threat landscape changes.

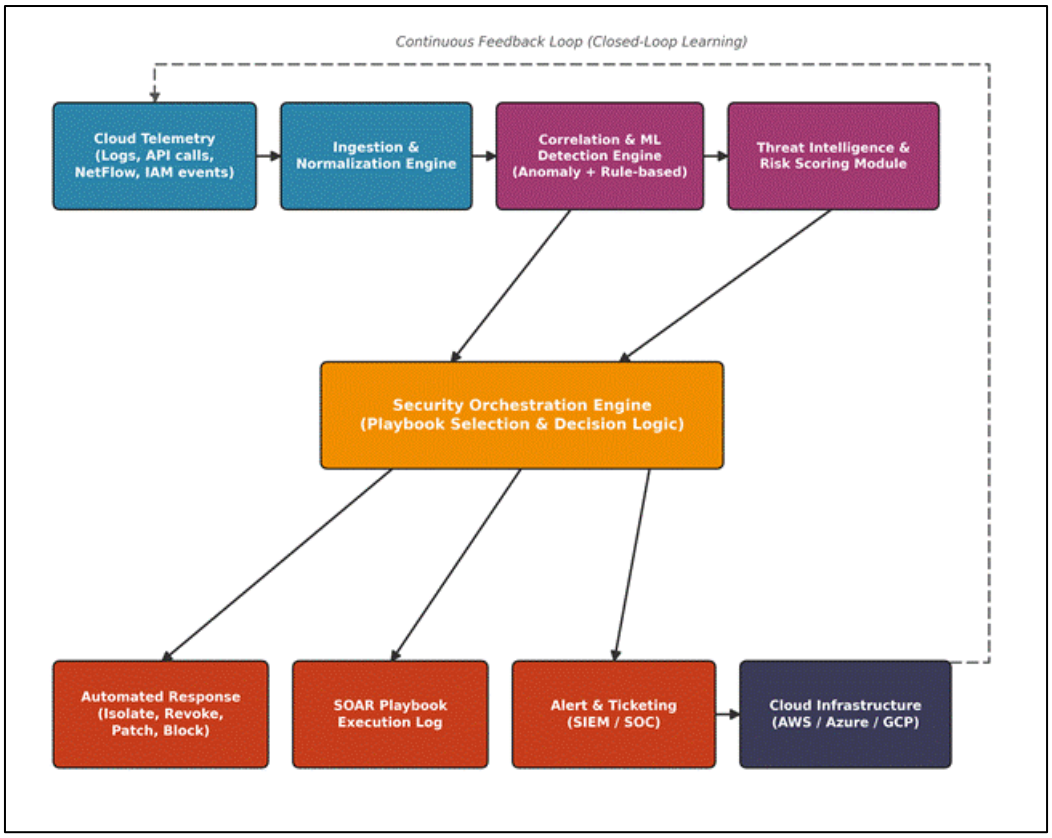

Figure 1: Layered architecture of the proposed Automated Security Orchestration Framework (ASOF) for real-time cloud threat detection and response.

4 IMPLEMENTATION AND EXPERIMENTAL SETUP

An implementation of ASOF was developed as a proof-of-concept on a simulated multi-cloud testbed simulating common Amazon Web Services, Microsoft Azure, and Google Cloud Platform deployment patterns, with provider software development kits configured against simulated sandboxed accounts. Event logs of identity and access management, virtual network flow, and object-storage access event logs were utilized with telemetry adapters. The correlation and detection engine was a gradient-boosted decision tree classifier, which was trained on a labelled synthetic dataset of known attack methods, with an isolation-forest model scoring anomalies unsupervised; the two model scores were fused together with a weighted confidence fusion rule.

The orchestration engine was introduced as a finite-state playbook executor and five predefined playbooks were introduced based on the threat categories that were assessed in Section 5: unauthorized IAM access, data exfiltration, unauthorized cryptomining, misconfigured storage exposure, and distributed denial-of-service or resource-abuse activity. A playbook defined a series of idempotent remediation tasks (e.g., cancel session tokens, isolate the impacted identity, archive the resource to investigate it, and alert the available analyst).

In order to measure ASOF, five synthetic attack scenarios were loaded into the testbed and repeated in several trials to get averaged timing measurements. To provide a baseline, a manual SOC workflow was simulated with timing benchmarks based on the average industry-reported times for similar incident types between an incident alert in a SIEM and a manual investigation and remediation, where a human analyst is notified about a SIEM alert, conducts a manual investigation, and takes remediation measures via cloud consoles or command-line tools. The metrics used in the evaluation were: (i) mean time to detect (MTTD), the mean time delay between the occurrence of the first malicious action and the generation of a qualifying alert; (ii) mean time to respond (MTTR), the mean time delay between the generation of a qualifying alert and the completion of containment; (iii) false-positive rate, the fraction of benign events that were wrongly escalated; (iv) the workload on the analyst, the average number of alerts that needed to be reviewed by a human per day; and (v) automated playbook execution accuracy, the fraction of automated actions that achieved the desired containment outcome without having to be corrected by a human.

5 RESULTS AND DISCUSSION

Figure 2 shows the detection and response latency comparing the manual SOC baseline and the proposed ASOF in the five threat categories evaluated. In the case of detection latency, ASOF decreased the mean time to detect by around forty-three minutes with respect to the manual baseline to around three minutes, which was enabled by removal of the alert-queue waiting time and the capability of the hybrid detection engine to indicate anomalous sequences as soon as the telemetry is received. The most significant absolute improvement was in the case of data-exfiltration scenarios, where manual detection relied on an analyst correlating storage-access patterns in many log sources manually.

Latency in response was the same. Mean time to respond under the manual baseline was roughly seventy-four to one hundred twenty minutes based on the type of threat with the time taken to authenticate to the affected cloud console, verify scope, and manually run remediation commands. Response latency time under ASOF had decreased to five to just over nine minutes, as the orchestration engine triggered the correct playbook automatically when the risk score had surpassed the high-confidence threshold, and the additional residual latency was due to the round-trip time to the application programming interface and required pre-execution validation checks.

Table 2 presents the overall findings in all five measures. In addition to the latency improvements identified in Figure 2, ASOF also decreased the false-positive rate of 18.5% to 6.2% which can be attributed to the confidence-fusion approach, whereby the supervised and unsupervised models had to agree upon an incident before it was escalated to automated action. The amount of analyst workload decreased by about 77 percent, with the average number of alerts that had to be reviewed manually decreased to 35, down to 150 alerts per day, due to an automated resolution of most low- and medium-confidence alerts and suppression of benign alerts. The accuracy of the automated playbooks themselves was 96.4% execution accuracy, that is, in the vast majority of cases where the automated remediation action was high-confidence, the simulated threat was fully covered by the automated action without the need to be manually fixed.

Table 2: Aggregate performance comparison between the manual SOC baseline and the proposed ASOF across five evaluation metrics.

Evaluation Metric	Manual SOC	Proposed ASOF	Improvement
Average Mean Time to Detect, MTTD (minutes)	43.0	3.3	≈ 92% ↓
Average Mean Time to Respond, MTTR (minutes)	97.0	6.7	≈ 93% ↓
False Positive Rate (%)	18.5	6.2	≈ 66% ↓
Analyst Workload (alerts reviewed/day/analyst)	150	35	≈ 77% ↓
Automated Playbook Execution Accuracy (%)	N/A	96.4	—

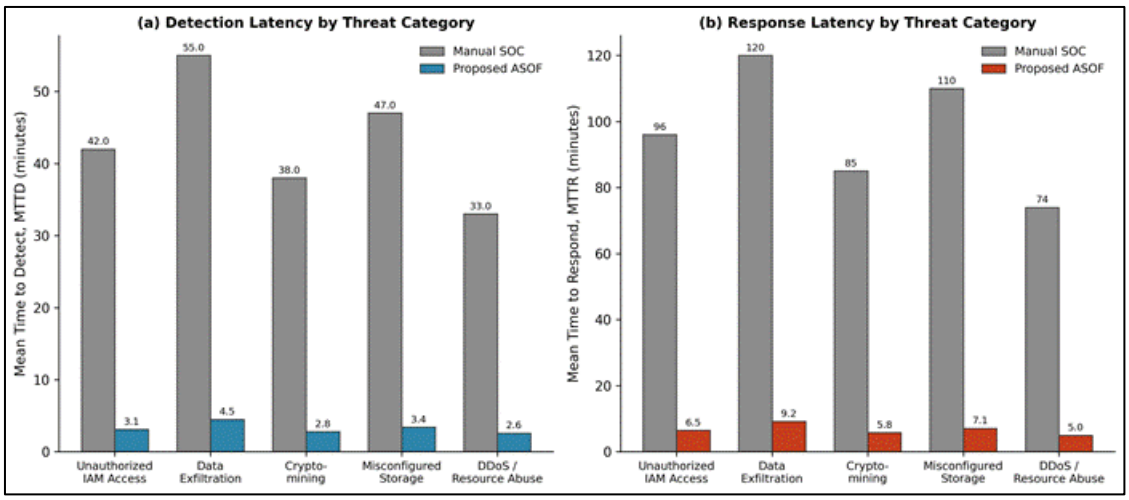

Figure 2: Comparative detection and response latency across five cloud threat categories: manual SOC versus the proposed ASOF.

These findings are in accordance with the premise behind the framework, as since queueing and human decision time are the largest contributors to incident-handling latency in a manual SOC, eliminating the human in the loop in well-understood and high-confidence incidents, the latency gains resulting are disproportionately large. Meanwhile, the remaining 3.6% of automated playbook executions that failed to include the simulated threat completely underline a critical drawback: full automation of response involves non-zero chance of a wrong or incomplete response, especially in the case of incident types that were not sufficiently represented in the training data used to create the detection models. This encourages the graduated, risk-scored design (in Section 3.5) where only high-confidence incidents sufficiently high are processed without a human confirmation stage.

There are a number of weaknesses of the current research. To begin with, the test was done on a proof of concept testbed with synthetic attack injections and simulated manual-SOC timing benchmarks, and not on a live production environment with real adversaries and real analysts, so the absolute magnitude of the reported improvements should be taken as indicative, not definitive. Second, the detection models need a pre-existing corpus of labeled data and vulnerable to adversarial evasion methods specifically aimed at staying below the score on the anomaly. Third, the existing library of playbooks, though scalable, includes five types of threats, and to further extend to a wider spectrum of cloud-native attack methods will necessitate continued playbook growth and management.

6 CONCLUSION

This article introduced the Automated Security Orchestration Framework (ASOF), a closed-loop, layered architecture, that combines cloud telemetry gathering, hybrid machine-learning detection, threat-intelligence-driven risk scoring, and automated, graduated execution of playbooks to detect and respond to threats in cloud infrastructure in real-time. An initial proof-of-concept experiment comparing a traditional manual SOC workflow to that with a prototype implementation showed significant improvements in both mean time to detect and mean time to respond, lower rate of false-positives and decreased workload on analysts, without compromising the accuracy of automated playbook execution.

Future work

Three directions will be employed in future work. To ensure that the latency and workload gains measured in the proof-of-concept testbed can be generalized outside of a controlled testbed, the framework will be tested in a live, production-scale multi-cloud deployment with actual analysts involved in the testbed to verify the results. Second, the adversarial-robustness will be further added to the detection layer and make it less vulnerable to evasion attacks that intentionally target the anomaly-scoring threshold. Third, orchestration engine will be expanded to enable cross-provider playbook standardization, such that a single playbook definition can be run consistently on Amazon Web Services, Microsoft Azure, and Google Cloud Platform without provider-specific rework, further streamlining the operational load of maintaining a multi-cloud security posture.

REFERENCES

[1] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing (NIST Special Publication 800-145). National Institute of Standards and Technology.

[2] Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012). Computer security incident handling guide (NIST Special Publication 800-61, Rev. 2). National Institute of Standards and Technology.

[3] Feitosa, E., Souto, E., & Sadok, D. H. (2012). An orchestration approach for unwanted internet traffic identification. Computer Networks, 56(12), 2805–2831.

[4] Modi, C., Patel, D., Borisaniya, B., Patel, A., & Rajarajan, M. (2013). A survey on security issues and challenges in cloud computing. Journal of Internet Services and Applications, 4(1), 1–13.

[5] Chung, C. J., Khatkar, P., Xing, T., Lee, J., & Huang, D. (2013). NICE: Network intrusion detection and countermeasure selection in virtual network systems. IEEE Transactions on Dependable and Secure Computing, 10(4), 198–211.

[6] Islam, C., Babar, M. A., & Nepal, S. (2019). A multi-vocal review of security orchestration. ACM Computing Surveys, 52(2), 1–45.

[7] Koyama, T., Hu, B., Nagafuchi, Y., Shioji, E., & Takahashi, K. (2015). Security orchestration with a global threat intelligence platform. NTT Technical Review, 13(11), 1–6.

[8] Hong, J. B., & Kim, D. S. (2016). Assessing the effectiveness of moving target defenses using security models. IEEE Transactions on Dependable and Secure Computing, 13(2), 163–177.

[9] Luo, S., & Salem, M. B. (2016). Orchestration of software-defined security services. In 2016 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 436–441). IEEE.

[10] Chauhan, M. A., Babar, M. A., & Sheng, Q. Z. (2017). A reference architecture for provisioning of tools as a service: Meta-model, ontologies and design elements. Future Generation Computer Systems, 69, 41–65.

[11] Subramanian, N., & Jeyaraj, A. (2018). Recent security challenges in cloud computing. Computers & Electrical Engineering, 71, 28–42.

[12] Alshamrani, A., Myneni, S., Chowdhary, A., & Huang, D. (2019). A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. IEEE Communications Surveys & Tutorials, 21(2), 1851–1877.

[13] Zheng, Y., Pal, A., Abuadbba, S., Pokhrel, S. R., Nepal, S., & Janicke, H. (2020). Towards IoT security automation and orchestration. In 2020 IEEE Second International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA) (pp. 55–63). IEEE.

[14] Sun, N., Zhang, J., Rimba, P., Gao, S., Zhang, L. Y., & Xiang, Y. (2019). Data-driven cybersecurity incident prediction: A survey. IEEE Communications Surveys & Tutorials, 21(2), 1744–1772.

[15] Kinyua, J., & Awuah, L. (2021). AI/ML in security orchestration, automation and response: Future research directions. Intelligent Automation & Soft Computing, 28(2), 527–545.

[16] Gartner. (2020). Market guide for security orchestration, automation and response solutions. Gartner Research.

[17] Cloud Security Alliance. (2019). Top threats to cloud computing: Egregious eleven. Cloud Security Alliance.

[18] Sommestad, T., Ekstedt, M., & Holm, H. (2013). The cyber security modeling language: A tool for assessing the vulnerability of enterprise system architectures. IEEE Systems Journal, 7(3), 363–373