A Mini SIEM/SOAR System for comprehensive cybersecurity monitoring of Microsoft Azure

TAOFEEK OLAYINKA AGBOOLA[1], PUSHKAR OGALE[2]
[1, 2]*Department of Computer Science, Stephen F. Austin State University, Nacogdoches, Texas*

Abstract- ***Outsourcing security management has gained traction among numerous organizations, often serving as the sole viable option in the absence of internal proficiency and infrastructure. The implementation of modern systems alone is no longer adequate for robust cybersecurity threat management. Managed security service providers now offer a comprehensive set of mature security monitoring and management capabilities, including security information and event management, strategic oversight of organizational governance, enterprise risk, and compliance with regulatory standards, making them a favored choice for a multitude of organizations. In an era of escalating cyber threats and data flood, the critical role of Security Operations Centers (SOCs) in safeguarding organizations' digital assets cannot be overstated. This work investigates how cybersecurity capabilities can be improved by creating and deploying a scaled-down version of Security Orchestration, Automation, and Response (SOAR) within Security Information and Event Management (SIEM) systems in Microsoft Azure environments. This setup would enable monitoring of various aspects including Network Security Group "firewall," endpoints, networks, and cloud resources. Acknowledging the mounting challenges faced by traditional security operation centers (SOC), they are overwhelmed with the ever-increasing volumes of data/alerts, while cyberattacks grow more sophisticated, often eluding conventional detection methods.***

Indexed Terms- ***SOC, SIEM, SOAR, Logic App, Incident Response, Azure***

I. INTRODUCTION

A. Overview

In today's digital world, cyberattacks are a constant threat, targeting everyone from individuals to large corporations as the persistent threat of cyberattacks poses a significant challenge to organizations worldwide. Sensitive data, often stored on computer systems, is increasingly vulnerable. As hackers employ more sophisticated tactics, organizations need to react swiftly to security breaches before attackers gain a foothold or access to critical systems.

A Security Operation Center (SOC) acts as a central command center, staffed with experts, processes, and technology. SOC continuously monitors an organization's security posture, analyzing security data to identify and prevent potential threats. By maintaining a real-time picture of an organization's security landscape, a SOC can react rapidly to suspicious activity, minimizing damage and protecting valuable data.

The ever-increasing reliance on interconnected systems raises the stakes. Widespread cyberattacks can disrupt essential services and cause significant financial losses. As the cybersecurity landscape constantly evolves with new and complex threats, SOCs have become a vital tool for organizations of all sizes to proactively combat cybercrime.

Our work embarked on enhancing cybersecurity defenses by deploying a mini-SIEM/SOAR system within an Azure environment, aiming to automate incident response and bolster security measures against potential cyberattacks. Azure uses large-scale virtualization at Microsoft data centers worldwide and it offers more than 600 services [1].

B. Problem Statement

Security professionals are drowning in a sea of security alerts generated by traditional, reactive security solutions. This overwhelming volume leads to a delay in detecting and responding to cyberattacks. On average, companies take approximately 20.9 hours to respond to cyberattacks, which equates to over two working days [2]. Also, an average SOC receives over 4000 alerts daily and it takes a minimum of 10 minutes to investigate and analyze an incident/alert with nearly 50% false positive alerts [3] . False positive alerts and others result in alert overload causing fatigue in the

security staff. To address that organizations resort to tuning out certain alerts. Figure 1 shows results from a survey of this practice. This results in slower response to security incidents. This slow response time significantly increases the risk of data breaches, costing organizations millions and damaging organizational reputation (Microsoft Azure). Our work proposes a proactive and efficient approach by leveraging cloud technologies to build a mini SIEM/SOAR platform with the capability automate incident response after utilizing honeypots to generate genuine alerts or valuable security logs and enhance compliance with industry regulations (NIST framework).

By automating incident response and improving detection times, our work aims to significantly reduce the risk of data breaches, enhance incidence response time, and empower security teams to be more effective. This ultimately strengthens an organization's overall cybersecurity posture and fosters a proactive security culture. This problem statement is geared towards security professionals, IT managers, and executives seeking innovative solutions to improve organization's cybersecurity posture.

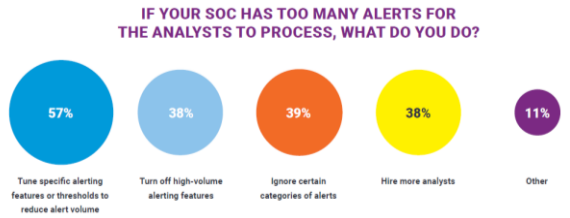


Fig. 1. Alert overload survey result by Critical Start

C. Literature Review

The integration of SIEM and SOAR within cloud environments represents a forward-thinking approach to cybersecurity (Microsoft Azure). SIEM systems collect, and aggregate log data generated across an organization’s technology infrastructure, from host systems and applications to network devices, providing real-time analysis to enable threat detection, compliance, and security incident management. SOAR platforms, on the other hand, are designed to automate the response to cyber threats, thereby reducing the time from detection to resolution and enabling security teams to focus on more strategic tasks [4]

The shift towards cloud computing has significantly impacted the design and implementation of cybersecurity systems. Azure, Microsoft's cloud computing service, offers robust capabilities for deploying SIEM and SOAR solutions, providing scalability, flexibility, and access to advanced analytics tools [5]. The integration of Azure Security Center and Azure Sentinel provides a comprehensive framework for security monitoring, threat detection, and automated incident response, embodying the principles of a mini SIEM/SOAR system [6]

Research by [7] underscores the importance of cloud-based SIEM systems in enhancing the security posture of organizations. They argue that cloud platforms offer the scalability necessary to process vast amounts of data generated by digital infrastructures, a critical capability for effective SIEM solutions. In IBM’s Cyber Resilient Organization Study 2021, it was emphasized that SOAR (Security Orchestration, Automation, and Response) platforms play a vital role in managing the growing volume and intricacy of cyber threats. The study revealed that 29% of surveyed organizations had implemented 31–50 distinct security tools and technologies, while 23% had deployed 51–100 tools. However, these tools are not inherently compatible, necessitating manual integration by security operations centers (SOCs) in response to each security incident. SOAR platforms offer a centralized console where these tools can be harmonized into optimized threat response workflows, automating repetitive tasks, and streamlining incident management. By doing so, SOARs help reduce mean time to detect (MTTD) and mean time to respond (MTTR), ultimately enhancing overall security posture [8].

The deployment of SIEM/SOAR systems in Azure specifically benefits from the cloud's elasticity, allowing organizations to dynamically adjust resources according to their needs. This is particularly relevant for small and medium-sized enterprises (SMEs) that may not have the resources to implement large-scale cybersecurity operations [9]. The implementation of SIEM in cloud environments can significantly reduce the cost and complexity associated with traditional SIEM deployments, making it an attractive option for organizations with limited IT budgets [10].

However, designing and implementing a mini SIEM/SOAR system in Azure is not without challenges. Security and privacy concerns, data integration issues, and the need for skilled personnel are among the obstacles that organizations must navigate. Despite these challenges, the potential benefits of enhanced security, compliance, and operational efficiency make the pursuit of such systems a worthwhile endeavor for organizations aiming to strengthen their cybersecurity defenses.

D. Proposed Solution

Our work addresses the challenge of overwhelmed security teams by building a cost-effective and scalable mini-SOC in Microsoft Azure. This mini-SOC leverages cloud technologies to provide real-time security monitoring and automated incident response capabilities.

Honeypot Setup: To help generate logs, the solution will utilize honeypots "intentionally vulnerable systems" with MSSQL – to attract attackers. Logs generated from honeypot interactions will be used to create a detailed attack map and incident providing a broader view of attacker tactics.

Log collection and analysis: Log Analytic Workspace (LAW) functions as the central repository of logs. Azure Sentinel serves as the Security Information and Event Management (SIEM) platform, that collect logs from network devices, endpoints, firewalls, and cloud resources for centralized analysis in the Log Analytic Workspace.

Secure the Cloud: In securing the cloud, we will utilize the private endpoint connection in Azure Private Link by provisioning the key vault and storage in a subnet and denying them access to public internet. Just like the instance of an intranet that does not allow users outside its network to connect.

NIST 800-53 (Security and Privacy Control for Information System and Organization) will be used to secure the environment and resources while NIST 800-61 (Computer Security Incident Handling Guide) provides a set of guidelines that will be used to develop security playbooks in Azure Logic App. These playbooks outline the strategic actions to be taken in response to different types of security incidents from preparation to post incidence activities.

Automated incident response: Set up a rule that automatically response to incidents based on the set conditions. In accomplishing this, Azure Logic App will be provisioned as part of the incident response process, and the task will be executed as detailed in the playbook.

II. SOLUTION DESCRIPTION

A. Glossary

VM	Virtual Machine
SOC	Security Operation Center
SIEM	Security Information and Even Management
SOAR	Security Orchestration, Automation, and Response
NIST	National Institute of Standard and Technology
VNet	Virtual Network
NSG	Network Security Group
MSSQL	Microsoft SQL Server
SSMS	SQL Server Management Studio
AD	Active Directory
AAD	Azure Active Directory
LAW	Log Analytic Workspace
MDC	Microsoft Defender for Cloud
PE	Private Endpoint

B. Description of Solution

Our work involves setting up a cybersecurity environment on Microsoft Azure, focusing on creating a honeypot system to attract potential attackers to derive/trigger an alert for incident handling.

Microsoft Account: A paid Microsoft account is created to accommodate project needs.

Virtual Machines (VMs): Two Windows and a Linux VMs were set up, with one Windows VM hosting Microsoft SQL Server to attract attackers.

Resource Group: Acts as a container for project resources, ensuring consistency across configurations.

Virtual Network (VNet): Connects VMs and other resources, assigns IP addresses, and facilitates internet access.

Network Security Group (NSG): Acts as a firewall, allowing all inbound traffic to generate logs for analysis.

MSSQL Server: Installed on a Windows VM to create an enticing target for attackers.

Logging: Enabled to ensure comprehensive monitoring and incident response, with logs stored in Windows Event Viewer.

Azure Active Directory (Entra ID): Used for enhanced security, authentication, and access control, generating logs for audit trails.

Log Analytic Workspace (LAW): Central database for all logs, used for analysis and querying.

Microsoft Sentinel: SIEM application integrated with LAW for comprehensive monitoring and incident management.

GeoIP Watchlist: Ingested into SIEM to structure logs and facilitate threat detection.

Microsoft Defender for Cloud (MDC): Provides security insights and forwards logs to LAW.

Azure Storage Account: Used for storing NSG flow logs.

Log Analytic Agent: Installed on VMs for logging and monitoring.

Blob Storage and Key Vault: Set up as part of a functioning Security Operation Center (SOC) to manage password, hash and encryption keys.

Sentinel Attack Map: Visualizes malicious activity worldwide, based on analytic rules.

Analytic Rules: Developed using KQL to query and generate alerts for security incidents.

Run Unsecure Environments: VMs configured to receive logs from any source to generate comprehensive logs.

Secure Cloud Configuration: Implements NIST 800-53 security controls and Azure Private Links for secure communication.

Secure Score: Assesses and ensures the security posture of Azure resources.

Automation Rules: Uses scripted automation for incident handling based on predefined conditions.

C. Solution Architecture

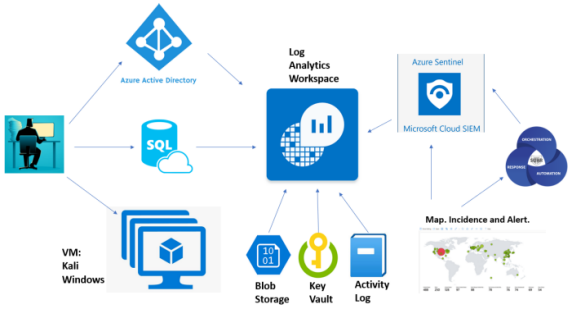


Fig. 2. Solution Architecture

III. IMPLEMENTATION

A. Azure Environment Setup and Tools Configuration

A Step-by-step approach of the Azure environment setup is completed. This includes virtual machine configuration, network security group settings, the installation of MSSQL Server, blog for storage of unstructured data like image, document, videos etc., key vault to store password, hashes and keys, Microsoft Defender for Cloud, Entra ID, Virtual Networks, Log Analytic Workspace, implement NIST 800-53 for security and privacy control, building the attack map and NIST 800-61 to develop a playbook workflow that defines expected actions for respective incidences. It details the process of configuring Azure Sentinel for centralized log management and the deployment of SOAR capabilities for incident automation.

B. Microsoft Sentinel Attack Maps Configuration

Some samples of the configuration settings needed are depicted below.

Integrating MSSQL Failed Authentication script into Microsoft Sentinel Workbook for data insight and monitoring with visualization.

```
let GeoIPDB_FULL = _GetWatchlist("geoip");
let IpAddress_REGEX_PATTERN = @"\b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b";

// Brute Force Attempt MS SQL Server
Event
| where EventLog == "Application"
//| where EventID == 18456
| project TimeGenerated, AttackerIP = extract(IpAddress_REGEX_PATTERN, 0, RenderedDescription), DestinationHostName = Computer, RenderedDescription
| project TimeGenerated, AttackerIP = extract(IpAddress_REGEX_PATTERN, 0, RenderedDescription), DestinationHostName = Computer, RenderedDescription
| evaluate ipv4_lookup(GeoIPDB_FULL, AttackerIP, network)
| project TimeGenerated, AttackerIP, DestinationHostName, RenderedDescription, latitude, longitude, city = city_name, country = country_name, friendly_location = strcat(city_name, " (", country_name, ")");
```

Fig. 3. MSSQL failed authorization configuration

Integrating Network Security Group allowed in Malicious traffic script into Microsoft Sentinel Workbook to be viewed on the attack map.

```
let GeoIPDB_FULL = _GetWatchlist("geoip");
let MaliciousFlows = AzureNetworkAnalytics_CL
| where FlowType_s == "MaliciousFlow"
| order by TimeGenerated desc
| project TimeGenerated, FlowType = FlowType_s, IpAddress = SrcIP_s, DestinationIpAddress = DestIP_s, DestinationPort = DestPort_d, Protocol = L7Protocol_s, NSGRuleMatched = NSGRules_s;
MaliciousFlows
| evaluate ipv4_lookup(GeoIPDB_FULL, IpAddress, network)
| project TimeGenerated, FlowType, IpAddress, DestinationIpAddress, DestinationPort, Protocol, NSGRuleMatched, latitude, longitude, city = city_name, country = country_name, friendly_location = strcat(city_name, " (", country_name, ")")
```

Fig. 4. NSG Malicious traffic configuration

Integrating Windows failed Remote Desktop Authentication script into Microsoft Sentinel Workbook to be viewed on the attack map.

```
let GeoIPDB_FULL = _GetWatchlist("geoip");
let WindowsEvents = SecurityEvent;
WindowsEvents | where EventID == 4625
| order by TimeGenerated desc
| evaluate ipv4_lookup(GeoIPDB_FULL, IpAddress, network)
| project TimeGenerated, Account, AccountType, Computer, EventID, Activity, IpAddress, LogonTypeName, network, latitude, longitude, city = city_name, country = country_name, friendly_location = strcat(city_name, " (", country_name, ")");
```

Fig. 5. Windows Remote Desktop Authentication configuration

Integrating Linux SSH authentication fail script into Microsoft Sentinel Workbook to be viewed on the attack map.

```
let GeoIPDB_FULL = _GetWatchlist("geoip");
let IpAddress_REGEX_PATTERN = @"\b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b";
Syslog
| where Facility == "auth"
| where SyslogMessage startswith "Failed password for"
| order by TimeGenerated desc
| project TimeGenerated, SourceIP = extract(IpAddress_REGEX_PATTERN, 0, SyslogMessage), DestinationHostName = HostName, DestinationIP = HostIP, Facility, SyslogMessage, ProcessName, SeverityLevel, Type
| evaluate ipv4_lookup(GeoIPDB_FULL, SourceIP, network)
| project TimeGenerated, SourceIP, DestinationHostName, DestinationIP, Facility, SyslogMessage, ProcessName, SeverityLevel, Type, latitude, longitude, city = city_name, country = country_name, friendly_location = strcat(city_name, " (", country_name, ")");
```

Fig. 6. Linux SSH authorization failure configuration

C. Unit Test and Triggering of Sentinel Alerts

```
// Brute Force attempt/failed logon on Windows
SecurityEvent
| where EventID == 4625
```

```
| where TimeGenerated > ago(60m)
| summarize FailureCount = count() by AttackerIP = IpAddress, EventID, Activity, DestinationHostName = Computer

// Brute Force Attempt MS SQL Server
let IpAddress_REGEX_PATTERN = @"\b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b";
Event
| where EventLog == "Application"
| where EventID == 18456
// Error code 18456 = Server could not authenticate the connection
| where TimeGenerated > ago(1hr)
| project TimeGenerated, AttackerIP = extract(IpAddress_REGEX_PATTERN, 0, RenderedDescription), DestinationHostName = Computer, RenderedDescription
| summarize FailureCount = count() by AttackerIP, DestinationHostName
| where FailureCount >= 10

// Brute Force Attempt / Failed access attempts for Azure Key Vault
AzureDiagnostics
| where ResourceProvider == "MICROSOFT.KEYVAULT"
| where ResultSignature == "Forbidden"

// Brute Force Attempt / Failed AAD logon
let FailedLogons = SigninLogs
| where Status.failureReason == "Invalid username or password or Invalid on-premise username or password."
| where TimeGenerated > ago(1h)
| project TimeGenerated, Status = Status.failureReason, UserPrincipalName, UserId, UserDisplayName, AppDisplayName, AttackerIP = IPAddress, IPAddressFromResourceProvider, City = LocationDetails.city, State = LocationDetails.state, Country = LocationDetails.country, Latitude = LocationDetails.geoCoordinates.latitude, Longitude = LocationDetails.geoCoordinates.longitude
| summarize FailureCount = count() by AttackerIP, UserPrincipalName;

// Excessive Password Reset
AuditLogs
| where OperationName startswith "Change" or OperationName startswith "Reset"
| order by TimeGenerated
| summarize count() by tostring(InitiatedBy)
| project Count = count_, InitiatorId = parse_json(InitiatedBy).user.id, InitiatorUpn = parse_json(InitiatedBy).user.userPrincipalName, InitiatorIpAddress = parse_json(InitiatedBy).user.ipAddress
| where Count >= 5

// Possible privilege Escalation: Azure Key Vault critical credential retrieval or Update.
let CRITICAL_PASSWORD_NAME = "Tenant-Global-Admin-Password";
AzureDiagnostics
| where ResourceProvider == "MICROSOFT.KEYVAULT"
| where OperationName == "SecretGet" or OperationName == "SecretSet"
| where id_s contains CRITICAL_PASSWORD_NAME

// Brute Force Success Windows
let FailedLogons = SecurityEvent
| where EventID == 4625 and LogonType == 3
| where TimeGenerated > ago(1h)
| summarize FailureCount = count() by AttackerIP = IpAddress, EventID, Activity, LogonType, DestinationHostName = Computer
| where FailureCount >= 5;
let SuccessfulLogons = SecurityEvent
| where EventID == 4624 and LogonType == 3
| where TimeGenerated > ago(1h)
| summarize SuccessfulCount = count() by AttackerIP = IpAddress, LogonType, DestinationHostName = Computer, AuthenticationSuccessTime = TimeGenerated;
SuccessfulLogons
| join kind = inner FailedLogons on DestinationHostName, AttackerIP, LogonType
| project AuthenticationSuccessTime, AttackerIP, DestinationHostName, FailureCount, SuccessfulCount

//Malware Detection
Event
| where EventLog == "Microsoft-Windows-Windows Defender/Operational"
```

```
| where EventID == "1116" or EventID == "1117"

// Windows host firewall tampering
Event
| where EventLog == "Microsoft-Windows-Windows Firewall With Advanced Security/Firewall"
| where EventID == 2003
```

Fig. 7. Analytic rules for triggering of sentinel alerts

IV. RESULT

Testing involved simulating attacks to generate and analyze security logs, evaluating the effectiveness of the implemented security measures. The results demonstrated the system's capability to detect and respond to known logs effectively. We provide the results of our experimentation and the security benefits in the next section

A. Scoreboard before and after securing the SOC

The Table 1 depicts the scores achieve before securing the resources, when it was left open to public internet to generate logs for the purpose of analysis and after the resources were secured from potential and real attackers through application of all the mentioned steps.

Table 1: Score board before and after securing the SOC

Score	Before	After
Start Time	2/8/2024, 2:16:52.698 PM	2/12/2024, 9:28:19.972 PM
Stop Time	2/9/2024, 2:16:52.698 PM	2/13/2024, 9:28:19.972 PM
Security Events (Windows VMs)	8007	779
Syslog (Linux VMs)	886	5
Security Alert (Microsoft Defender for Cloud)	9	0
Security Incident (Sentinel Incidents)	94	0
NSG Inbound Malicious Flows Allowed	312	0

B. Secure Score

After protecting the boundaries with NIST 800 53, Microsoft Defender for Cloud, Azure Private Link and Firewall, we achieved a 76% secure score. Figure 8 depicts the Secure Score from the system.

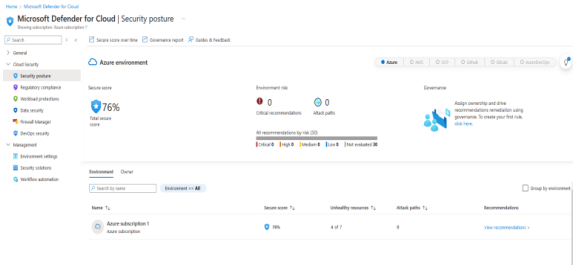
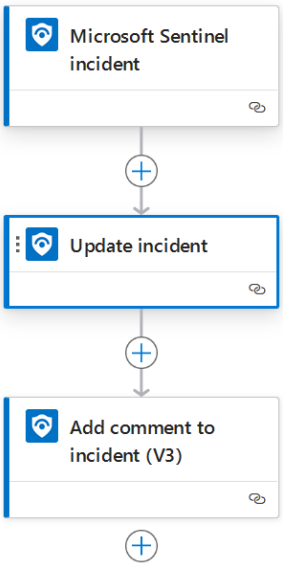


Fig. 8. Secure Score after protection with NIST 800 53

C. Logic App Workflow

Figure 9 depicts the Logic apps workflow. The workflow details the orchestrated playbook. The Logic apps playbook allows us to configure and automate when the conditions are met in the set analytics rules.



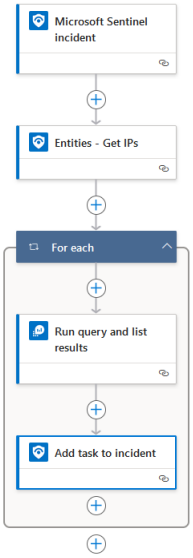


Fig. 9. Logic App workflow

CONCLUSION

Our work demonstrates the effectiveness of integrating SIEM and SOAR functionalities into Azure to enhance cybersecurity measures. The findings advocate for the adoption of cloud-based cybersecurity frameworks, emphasizing the importance of automation in cost and time savings, enabling SOC Analysts to prioritize monitoring suspicious alerts in addressing the dynamic cyber threat environment. Leveraging SOAR technology, SOC teams previously overwhelmed with repetitive and time-consuming tasks can now enhance their incident resolution efficiency, leading to cost reduction, coverage gap mitigation, and increased productivity.

Our work demonstrated developing an incidence auto-response to automate concurrent incidents by automating repetitive tasks and orchestrating workflows with Logic App. This allows and enables SOC analysts to focus on more strategic and complex security issues. It frees up SOC analysts from alert fatigue and having to tune out alerts due to the overload.

Our work demonstrates methodology to the development and execution of modern security operations centers, aiding in their planning and implementation. Moreover, established security operations centers can benefit from the insights gained from our work to enhance their current practices.

REFERENCES

[1] Wikipedia, "Microsoft Azure," [Online]. Available: https://en.wikipedia.org/wiki/Microsoft_Azure

[2] VentureBeat, "Cyberattack response time averages 2 days," October 13, 2021, https://venturebeat.com/business/cyberattack-response-time-averages-2-days-report-finds/

[3] Critical Start, "THE IMPACT OF SECURITY ALERT OVERLOAD", https://www.criticalstart.com/wp-content/uploads/2021/02/CS_Report-The-Impact-of-Security-Alert-Overload.pdf

[4] Gartner, Inc, "Innovation Insight for Security Orchestration, Automation and Response," 2017, https://www.gartner.com/en/documents/3834578

[5] Microsoft, "What is Microsoft Sentinel?", https://learn.microsoft.com/en-us/azure/sentinel/overview?tabs=azure-portal

[6] C. Davis, Cloud-Native Patterns: Designing Change-Tolerant Software, Manning Publications, 2019.

[7] H. R. Wiem Tounsi, A survey on technical threat intelligence in the age of sophisticated cyber-attacks, Elsevier, 2018.

[8] IBM, "What is SOAR?", IBM, 2021 https://mediacenter.ibm.com/media/What%20is%20SOAR%20(Security%2C%20Orchestration%2C%20Automation%20and%20Response)/1_lsc3n40h

[9] I. M. a. A. B. A. Serckumecka, "Low-Cost Serverless SIEM in the Cloud: 2019 38th Symposium on Reliable Distributed Systems (SRDS)," Lyon, France, 2019.

[10] J. N. D. Huang, "Trust mechanisms for cloud computing," Journal of Cloud Computing: Advances, Systems and Applications, 2013.