**Scheduled Task Threat Detection and Mitigation in Azure using Microsoft Sentinel SIEM**

By

Kiprop, Valerie Cheruto

146640

A Computer Networks and Cyber Security Project II submitted to the School of Computing and Engineering Sciences in partial fulfilment of the requirements for the award of the Bachelor’s Degree in Computer Networks and Cyber Security of Strathmore University

School of Computing and Engineering Sciences

Strathmore University

Nairobi, Kenya

July 2024

Declaration and Approval

I declare that this work has not been previously submitted and approved for the award of a Bachelor’s degree by this or any other University. To the best of my knowledge and belief, the proposal contains no material previously published or written by another person except where due reference is made in the proposal itself.

Student Name: **Kiprop, Valerie Cheruto**

Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Approval**

The Computer Networks and Cyber Security Project II proposal of Valerie Kiprop was reviewed and approved (for examination) by:

Supervisor’s Name: **Humphrey Owuor**

Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Acknowledgement

I would like to extend my heartfelt thanks and deep appreciation to the individuals who have offered me invaluable support and guidance throughout the course of this project.

I express my gratitude to Mr. Humphrey Owuor for his exceptional guidance, mentorship, and expertise. His feedback, encouragement, and belief in my abilities have driven the success of the project. His dedication to my research growth and development has been inspiring, and I am grateful for the invaluable lessons learned under his supervision.

I express my gratitude to my family for their unwavering love, understanding, and support throughout my journey. Their belief in me and sacrifices have been a constant source of motivation. Their encouragement during challenging and triumphant moments has been indispensable, and I am forever grateful for their presence in my life. I am grateful to all these individuals for their invaluable contributions, encouragement, and belief in my abilities. Their support has been instrumental in the successful completion of this project.

Abstract

This project aims to implement Security Information and Event Management (SIEM) capabilities in cloud environments using Microsoft Sentinel, with a specific focus on detecting and mitigating threats associated with scheduled tasks. As businesses increasingly rely on Microsoft Azure for critical services and infrastructure, the risk posed by malicious actors exploiting scheduled tasks to execute persistent threats becomes a significant concern. This study involves creating a scheduled task that mimics an attack, which will create a baseline of distinguishing between benign and malicious scheduled tasks and implementing an automated mitigation strategy using Azure Sentinel. By leveraging Sentinel’s analytics and automated response features, the project seeks to improve threat detection accuracy, minimize false positives, and ensure the prompt isolation of compromised virtual machines (VMs). The proposed solution aims to provide a framework for improving cloud security in Azure environments, thereby protecting organizations from potential data breaches, financial losses, and reputational damage.

Table of Contents

[Declaration and Approval ii](#_Toc176739196)

[Acknowledgement iii](#_Toc176739197)

[Abstract iv](#_Toc176739198)

[Table of Contents v](#_Toc176739199)

[List of Figures vii](#_Toc176739200)

[List of Tables viii](#_Toc176739201)

[List of Abbreviations ix](#_Toc176739202)

[Chapter 1: Introduction 1](#_Toc176739203)

[1.1 Background Information 1](#_Toc176739204)

[1.2 Problem Statement 2](#_Toc176739205)

[1.3 Aim 2](#_Toc176739206)

[1.4 Specific Objectives 2](#_Toc176739207)

[1.5 Research Questions 3](#_Toc176739208)

[1.6 Justification 3](#_Toc176739209)

[1.7 Scope and Limitations 3](#_Toc176739210)

[1.7.1 Scope 3](#_Toc176739211)

[1.7.2 Limitations 4](#_Toc176739212)

[Chapter 2: Literature Review 5](#_Toc176739213)

[2.1 Introduction 5](#_Toc176739214)

[2.2 Microsoft Sentinel SIEM 5](#_Toc176739215)

[2.3 Scheduled Tasks Threat Detection 7](#_Toc176739216)

[2.4 Existing Solutions 7](#_Toc176739217)

[2.4.1 Splunk Enterprise Security 7](#_Toc176739218)

[2.4.2 IBM QRadar 8](#_Toc176739219)

[2.4.3 ArcSight ESM 8](#_Toc176739220)

[2.5 Gaps 9](#_Toc176739221)

[2.6 Conceptual Framework 10](#_Toc176739222)

[Chapter 3: Research Methodology 12](#_Toc176739223)

[3.1 Introduction 12](#_Toc176739224)

[3.2 Research Paradigm 12](#_Toc176739225)

[3.3 Methodology 12](#_Toc176739226)

[3.3.1 Requirement Gathering 13](#_Toc176739227)

[3.3.2 Design 14](#_Toc176739228)

[3.3.3 Tools and Techniques 16](#_Toc176739229)

[3.4 Testing 16](#_Toc176739230)

[3.5 Deliverables 17](#_Toc176739231)

[Chapter 4: System Analysis and Design 18](#_Toc176739232)

[4.1 Introduction 18](#_Toc176739233)

[4.2 System Requirements 18](#_Toc176739234)

[4.2.1 Functional Requirements 18](#_Toc176739235)

[4.2.2 Non-Functional Requirements 18](#_Toc176739236)

[4.3 System Architecture Diagram 19](#_Toc176739237)

[4.4 System Analysis Diagrams 20](#_Toc176739238)

[4.4.1 Network Diagram 20](#_Toc176739239)

[4.4.2 Use Case Diagram 21](#_Toc176739240)

[4.4.3 Sequence Diagram 22](#_Toc176739241)

[4.4.4 Flow Chart 23](#_Toc176739242)

[References 24](#_Toc176739243)

[Appendices 26](#_Toc176739244)

[Appendix 1 Gantt Chart 26](#_Toc176739245)

List of Figures

[Figure 2.1 Microsoft Sentinel 6](#_Toc176739325)

[Figure 2.2 Conceptual Framework 11](#_Toc176739326)

[Figure 3.1 Modified Waterfall Methodology 13](#_Toc176739327)

[Figure 4.1 System architecture diagram 20](#_Toc176739328)

[Figure 4.2 Network Diagram 21](#_Toc176739329)

[Figure 4.3 Use Case Diagram 22](#_Toc176739330)

[Figure 4.4 Sequence Diagram 23](#_Toc176739331)

List of Tables

[Table 2.1 Gaps 9](#_Toc172113055)

List of Abbreviations

SIEM Security Information and Event Management

VM Virtual Machine

SOAR Security Orchestration Automated Response

APTs Advanced Persistent Threats

SSAD Structured Systems Analysis and Design Method

KQL Kusto Query Language

NSG Network security groups

AAD Azure Active Directory

# Introduction

## Background Information

Microsoft Azure is a widely used cloud computing platform that offers a range of services, including virtual machines, databases, storage, and networking. Its security features make it a preferred choice for businesses of all sizes, enabling them to deploy and manage enterprise-level IT infrastructure efficiently (Verma et al., 2019). However, the widespread adoption of Azure also makes it a significant target for cyber threats, necessitating proper security measures to protect its infrastructure.

This project focuses on scheduled tasks which are a fundamental feature of operating systems allowing users and administrators to automate routine operations such as backups, updates, and maintenance tasks (Kannan et al., 2017). In Windows environments, the Task Scheduler service facilitates the creation, modification, and execution of these tasks based on predefined triggers and schedules. While scheduled tasks enhance system efficiency and reliability, they also present a security risk. Malicious actors can exploit scheduled tasks to perform unauthorized actions, persist on a compromised system, escalate privileges, or execute malicious code at specific times (Jang-Jaccard, J., & Nepal, S. 2018). Detecting and mitigating these threats is crucial to maintaining the integrity and security of IT environments.

Microsoft Sentinel is a cloud native SIEM (Security Information and Event Management) and SOAR (Security Orchestration Automated Response) solution that offers toolsets for implementing security in cloud environments. SIEM systems are essential components of modern cybersecurity frameworks, providing real-time analysis of security alerts generated by applications and network hardware. They aggregate and correlate data from multiple sources, offering a centralized view of an organization's security posture. Microsoft sentinel can be used to identify potential security incidents and reducing false positives. Sentinel's threat detection capabilities and automated response actions enable organizations to quickly mitigate risks, ensuring a more secure cloud environment (Tuyishime, E et al., 2023).

By implementing in Sentinel proper analytics and response actions, organizations can better detect and mitigate threats posed by malicious scheduled tasks within Azure environments. This integration improves the overall security posture, ensuring that potential threats are identified and addressed promptly, thereby minimizing the risk of widespread damage. This project aims to implement SIEM functionality in cloud environments by using Microsoft Sentinel to specifically target and manage the threats associated with scheduled tasks, providing a more secure cloud infrastructure

## Problem Statement

The problem this research addresses is the manipulation of scheduled tasks within Microsoft Azure environments by malicious actors. These tasks, designed for routine operations, can be exploited to execute APTs (Advanced Persistent Threats), posing a significant security threat (Uçar, Mesut. 2023). According to studies, attackers can use scheduled tasks to persist in the system, elevate privileges, and spread malware, often going undetected due to their legitimate appearance.

Ideally, scheduled tasks should only be used for benign purposes, such as maintenance and updates. However, the reality is that attackers can and do exploit these tasks to carry out harmful activities This creates a critical gap between the intended use of scheduled tasks and their potential misuse. The consequences of this gap are severe, leading to compromised systems, data breaches, and significant financial and reputational damage to affected organizations. Businesses relying on cloud for example Azure for their operations are particularly at risk, as a single compromised VM can threaten the entire network's security (Z. S. Zainudin et al 2018).

## Aim

To implement the detection and mitigation of malicious scheduled tasks within Microsoft Azure environments using tools found in Microsoft sentinel, thereby improving overall cloud security.

## Specific Objectives

1. To investigate the ability of Microsoft Sentinel’s Security Information and Event Management (SIEM) in detecting and mitigating threats posed by scheduled tasks in cloud environments.
2. To assess the challenges associated with detecting malicious scheduled tasks in Azure and review existing solutions.
3. To design and develop an Azure Sentinel solution to distinguish between benign and malicious tasks by implementing tools components in Azure Sentinel.
4. To test the developed solution to determine its ability to identify and isolate malicious scheduled tasks.

## Research Questions

* 1. How well does Microsoft Sentinel work in detecting and mitigating threats from scheduled tasks compared to traditional SIEM systems in cloud environments?
  2. What challenges exist in detecting malicious scheduled tasks in Azure environments?
  3. How can Azure Sentinel be utilized to differentiate between benign and malicious scheduled tasks?
  4. How well does the developed solution work in real-world scenarios for identifying and mitigating malicious scheduled tasks?

## Justification

This research is crucial because it addresses a security gap in azure cloud environments, that are widely used across various industries. By implementing the detection and mitigation of malicious scheduled tasks, the project aims to protect organizations from potentially devastating cyberattacks. The benefits of this research extend to improving the overall security posture of businesses and institutions, ensuring the integrity of their operations, and protecting sensitive data.

Moreover, this project contributes to the field of cybersecurity by providing a practical solution to an unacknowledged but hazardous problem. The development and testing of an automated detection and response system within Azure Sentinel can serve as a blueprint for similar initiatives, offering valuable insights and methodologies that can be replicated and adapted across different contexts. Ultimately, this research aims to create a more secure digital and cloud environment, reducing the risk of cyber threats and enhancing the resilience of cloud-based infrastructures.

## Scope and Limitations

### Scope

This study focuses on implementing the detection and mitigation of malicious scheduled tasks specifically within Microsoft Azure environments. The scope includes the assessment of current challenges, design and development of a solution using Azure Sentinel, and the testing of the implemented solution. The project covers azure cloud platform and scheduled tasks attacks in azure only and this focused approach ensures a thorough investigation and solution development.

### Limitations

Anticipated limitations include potential variability in the types of attacks executed using scheduled tasks and the evolving nature of cyber threats, which may affect the solution's long-term effectiveness. Additionally, the research relies on the capabilities of Azure Sentinel and the accuracy of its analytics, which may pose constraints. However, these limitations are acknowledged, and the project aims to provide a good foundation that can be adapted and enhanced as needed to address emerging threats.

# Literature Review

## Introduction

The literature review chapter covers existing research and technologies to enhance Security Information and Event Management (SIEM) capabilities in cloud using Microsoft Sentinel specifically for detecting threats and mitigating attacks from scheduled tasks. This review is organized based on the objectives of the study, starting with an overview of the area of study, including what is SIEM, what is Microsoft Sentinel, and the risks of scheduled tasks in cloud. The chapter then goes into the specific problems the study wants to address, how these affect the organization and what has been done so far to mitigate them. Finally, the review investigates existing solutions and studies in the field and identifies the gaps that the proposed solution will address.

## Microsoft Sentinel SIEM

SIEM systems are a key part of modern cybersecurity architecture by providing real time analysis of security alerts from applications and network devices. SIEM systems aggregate and correlate data from multiple sources to give a single view of an organisation’s security posture. This is key to identifying patterns and anomalies that may indicate security incidents so you can respond quicker and manage threats better (Kavanagh & Rochford, 2020).

In the context of cloud computing Microsoft Azure is a leading platform that provides virtual machines, databases, storage and networking. Azure’s scalability, flexibility and security features make it a popular choice for businesses. But with Azure being so widely used it’s also a bigger target for cyber attacks so there is a need to have advanced security to protect the infrastructure (Diogenes, et al., 2022).

Microsoft Sentinel is a cloud-native Security Information and Event Management (SIEM) solution designed to provide security analytics and threat intelligence across an enterprise. It can integrate with various Microsoft services and third-party solutions to offer visibility and protection. Sentinel can be used in data collection by integrating with numerous sources, including azure services, on-premises systems, and other cloud environments. It can be configured to use Azure Log Analytics as the backend for storing and querying log data, improving its data handling capabilities.

Additionally, regarding threat detection, Microsoft Sentinel offers a platform for analytics using custom rules written in Kusto Query Language (KQL) to help identify threats. Users can customize detection rules and queries to tailor the system to their specific needs. In this project, sentinel this feature is used to detect potential security incidents caused by scheduled tasks and can be used to reduce false positives (Copeland, M., 2021).

In terms of incident response, Sentinel supports automated actions through playbooks created using Azure Logic Apps, this can enable quick containment and mitigation of threats. Investigation tools such as workbooks, notebooks, and a timeline view help in analysing incidents and tracing attack paths, while integrated case management features assist in collaboration among security teams.

Sentinel threat intelligence abilities come from being able to incorporate feeds from various sources and creating alerts with additional information to prioritize and understand threats better. Its cloud-native architecture leverages Azure's infrastructure for scalability, high availability, and performance, with elastic scaling capabilities to handle large data volumes and high query loads.

All these features combined can be used in effectively detecting and mitigating malicious scheduled tasks. Scheduled tasks in windows environments, controlled by the task scheduler service, allow users and administrators to automate things like backups, updates and maintenance tasks. While these tasks make the system more efficient and reliable, they are a big security risk if an attacker can use them. (Sharma, M et al., 2022).

A diagram of a cloud

Description automatically generated

Figure 2.1 Microsoft Sentinel

## Scheduled Tasks Threat Detection

This study addresses the problem of malicious actors exploiting scheduled tasks to perform unauthorized actions in cloud environments. These threats can compromise the integrity and security of resources in the cloud, resulting in data breaches, financial losses and reputational damage. Scheduled tasks are exploited because they are trusted processes in the operating system and because of this, malicious actors often take advantage of these legitimate scheduling tools and techniques, making it difficult to distinguish between normal administrative tasks and malicious activities.

Additionally, without an established baseline of normal scheduled tasks and their behaviour, it can be difficult to identify anomalies. Scheduled tasks may interact with other systems and services within the azure environment, and this affects businesses by introducing vulnerabilities that can be exploited for extended period without detection. If not addressed, the consequences are continuous threats in the system, increased attack surface and potential widespread damage in case of an attack (Zainudin & Abdul Molok, 2018).

While Azure provides several security features and tools, they may not cover all aspects of scheduled task management. Previous attempts to solve this problem are developing threat detection algorithms and monitoring systems. However, those solutions lack accuracy in differentiating between malicious and benign scheduled tasks. Furthermore, although there is improvement in detecting suspicious activity, there is still a gap in automating response to mitigate the different identified threats (Lee et al., 2023).

## Existing Solutions

To address the challenges of detecting and mitigating threats related to scheduled tasks in cloud environments we need to review existing Security Information and Event Management (SIEM) solutions. This section will review three SIEM systems: Splunk Enterprise Security, IBM QRadar and ArcSight ESM. These solutions will be analysed based on their capabilities, technology used and effectiveness in addressing malicious scheduled tasks issues in cloud environments.

### Splunk Enterprise Security

Splunk Enterprise Security (ES) is a well known SIEM solution with strong data analytics capabilities. Splunk ES provides real-time monitoring, advanced threat detection and incident response by aggregating and analysing data from multiple sources including logs, network devices and endpoints. Splunk’s machine learning algorithms identify patterns and anomalies that indicate potential security threats, it’s a powerful tool for security professionals (Ananthapadmanabhan, A., 2023).

Splunk ES integrates with cloud services including Microsoft Azure, AWS and Google Cloud Platform in cloud environments. This allows organizations to monitor and secure their cloud infrastructure. However, Splunk ES is strong in data analytics but can be complex to manage and requires a lot of resources for deployment and maintenance. Even with its capabilities there may still be challenges in differentiating between benign and malicious scheduled task as there is no established base of established baseline of normal scheduled tasks and their behaviour, and malicious tasks. (Ananthapadmanabhan, A., 2023).

### IBM QRadar

IBM QRadar is another SIEM solution known for its capabilities in threat detection, incident response, and compliance management. It provides visibility into network traffic, user activities, and endpoint behaviour, helping organizations identify and respond to security threats effectively. QRadar's strengths include its analytics, machine learning algorithms, and ability to correlate vast amounts of data to detect complex attack patterns. (Barhami, Y. Et al., 2023).

However, despite its strengths, IBM QRadar faces challenges in scaling and integrating with specialized cloud services, particularly for automating mitigating strategies for detected malicious scheduled tasks. In cloud environments, the dynamic and distributed nature of resources necessitates seamless integration and scalability, areas where QRadar can encounter difficulties. The system's deployment and resource management complexities can hinder its effectiveness in real-time threat detection and response, especially for unexpected threats like malicious scheduled tasks.

### ArcSight ESM

ArcSight Enterprise Security Manager (ESM) by Micro Focus is a SIEM solution that focuses on centralised security monitoring and advanced threat detection. ArcSight ESM aggregates and correlates security data from multiple sources and provides real-time analysis and response to security incidents. Its data analytics and machine learning capabilities can detect sophisticated threats.

ArcSight ESM supports deployment in cloud environments including Microsoft Azure and AWS. This allows organizations to extend their security monitoring and threat detection to their cloud infrastructure. However, there is challenge in finding a single service that can specifically be used for detection and mitigation of malicious scheduled tasks in azure. Additionally, no clear strategies have been set dealing with detecting threats associated with scheduled tasks (Sekharan, S., et al 2022).

## Gaps

There are notable inadequacies in addressing the specific challenge of dealing with persistent attacks.

Table 2.1 Gaps

|  |  |  |  |
| --- | --- | --- | --- |
| **Other Works** | **Cited article** | **Gaps Identified** | **Proposed solution** |
| The paper “Advanced Persistent Threats (APT): evolution, anatomy, attribution and countermeasures” studies different types of persistent attacks. | Sharma, A., Gupta, B.B., Singh, A.K. et al. (2023). | This paper does not specify scheduled tasks as a way used by attackers to execute malicious tasks on a machine. | The use of malicious scheduled task to achieve persistence in the machines and its significant consequences will be well researched, documented and tested. |
| The paper “Towards the Automatic and Schedule-Aware Alerting of Internetwork Time Series," discusses monitoring and alerting for attacks. | D. Perdices, J. L. García-Dorado, J. Ramos. et al. (2023). | This paper focuses on monitoring and alerting of administrators in the event of an attack. It does not focus on providing a mitigation solution that can minimise the effect of the attack | This project uses Sentinel and integrates it with with Azure Logic App to implement seamless automation to detect and isolate a compromised VM to mitigate malicious scheduled tasks a type of persistent attack |
| The paper "Cloud Storage Defense Against Advanced Persistent Threats: A Prospect Theoretic Study,"  This paper discusses ways to defend against persistent attacks in the cloud. | L. Xiao, D. Xu, C. Xie, N. B. Mandayam and H. V. Poor, (2017) | This paper uses prospect theory (PT) is applied to formulate the interaction between the defender of a cloud storage system and an attacker. This method leaves room for various attacks as it only defends against attacks that can be anticipated or prospective attacks | This project uses log analytics workspace to connect to virtual machines thus analysing **all** tasks executed to detect any kind of attack |

## Conceptual Framework

The proposed solution involves integrating Microsoft Sentinel with azure resources to improve on its SIEM capabilities, specifically targeting the detection and mitigation of threats associated with scheduled tasks. The framework involves the following process:

First is scheduled task execution, where a scheduled task is executed on the Azure VM, generating a security event log. This is followed by log collection, the logs from the VM are collected and sent to azure log analytics workspace. Next is log analysis in azure sentinel, the collected logs are analysed using custom analytic rules and custom scripts to identify suspicious behaviour. The fourth step is detection and alert generation, when Sentinel identifies a malicious task, its configured to generate an alert. The fifth step is automated response: The alert triggers an automated response configured using the azure logic app tool. And the last step is mitigation and isolation, the response to the alert is to isolate the affected VM to mitigate the threat.

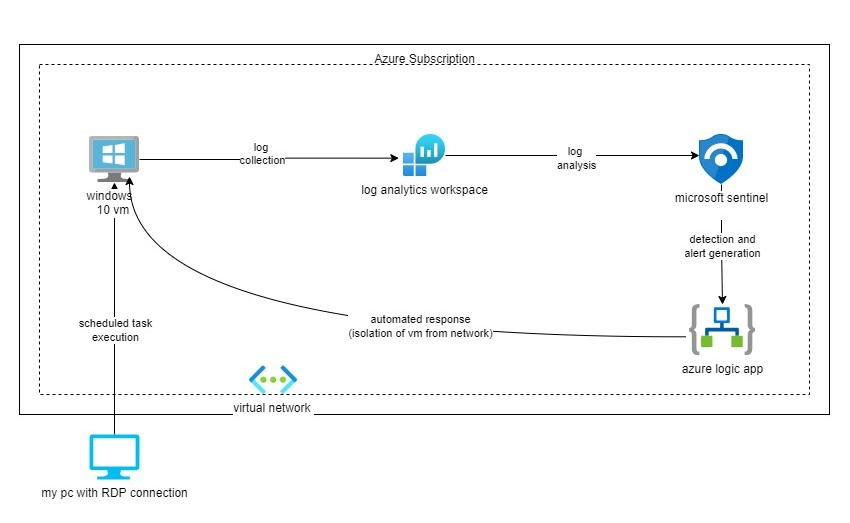


Figure 2.2 Conceptual Framework

# Research Methodology

## Introduction

This chapter details the methodology used for implementing SIEM capabilities in the cloud using Microsoft Sentinel for detecting and mitigating threats from scheduled tasks. It uses the Structured Systems Analysis and Design Method (SSAD) to provide a systematic approach to the analysis and design of the solution. The Agile methodology is chosen because of its flexibility and iterative nature, allowing for continuous feedback and adaptation throughout the project. The chapter covers the phases of ideation and requirement analysis, design, development, and testing, leading to the delivery of a well implemented and functional SIEM.

## Research Paradigm

The research paradigm chosen for this project is the Structured Systems Analysis and Design Method (SSAD). SSAD is a methodological approach that involves breaking down complex systems into smaller, manageable components through a series of systematic phases. It includes detailed analysis and design stages, ensuring a thorough understanding of the system requirements and a structured path to system development (Dennis, A., Wixom, B., & Tegarden, D. 2015).

SSAD is particularly suited for this project as first, its structured approach allows accurate documentation of the requirements and processes involved in implementing SIEM capabilities with Microsoft Sentinel. For instance, the detailed examination of scheduled task threats requires a clear and systematic method to identify potential vulnerabilities and define how these threats can be detected and mitigated. SSAD allows for this by providing tools and techniques to map out the processes, data flows, and system interactions. Its systematic approach ensures thorough documentation, precise analysis, and detailed design, making it the best choice for this project.

## Methodology

The methodology chosen for this project is the Modified Waterfall methodology. The Modified Waterfall approach was selected because it combines the structured, phase-based progression of the traditional Waterfall model with the flexibility of iterative feedback, making it well-suited for complex systems like SIEM that require planning and adaptability to evolving threats and requirements.

The Modified Waterfall methodology follows a sequential design process but allows for iterations between phases, enabling adjustments based on testing and feedback. This approach ensures thorough documentation and clear progression while accommodating necessary changes. In this project it can be used to develop the system incrementally, focusing on core functionalities first logging, then detection of malicious scheduled tasks, then alerting, and finally isolation. This methodology allows this process to be followed as clear steps are followed in the development process. The modified waterfall methodology provides a feedback system, there we can move back to the previous step, if there are any errors.



Figure 3.1 Modified Waterfall Methodology

### Requirement Gathering

This phase is important for understanding the problem and defining the necessary functionalities and constraints of the system to be implemented. It involves gathering information on Security Information and Event Management (SIEM), scheduled task threats, and cloud security best practices. It also involves identifying the data required to implement an SIEM that successfully detects and mitigates attacks launched using scheduled tasks.

SIEM systems aggregate and analyse log data from various sources within an IT infrastructure to detect and respond to potential security threats. Important features include real-time monitoring, threat detection through correlation and pattern recognition, incident response automation, and reporting for compliance purposes. Knowing how to integrate the SIEM with the existing environment, its scalability, and its capabilities for threat detection and response is important.

Data collection begins with the analysis of logs generated by the execution of a scheduled task on the target virtual machine. These logs include security events from virtual machines (VMs), and network security group (NSG) flow logs. The security events from VMs capture detailed information about scheduled task executions, when they were created, the user that created them, what task is executed and its respective path. These logs are ingested into Microsoft Sentinel where they are analysed to identify patterns that suggest malicious activity.

The data acquired from these logs is crucial in designing and implementing a solution that can be used to identify malicious scheduled attacks and create a mitigation strategy (Zannone, N et al., 2023).

### Design

The research approach will utilize Structured Systems Analysis and Design Method (SSAD), which supports the implementation of the SIEM system. SSAD is a method of analysis and design of a system that models it as a group of interacting systems, each broken down into manageable components through a series of systematic phases (Saad, A et al., 2019). This approach is particularly beneficial for interconnected systems as it allows for clear representation of system functionalities and interactions, making the system easier to manage, extend, and modify. This method is particularly suited for this project’s SIEM system as it is made up of different components coming together to create a whole solution as seen in my system architecture discussed below.

SSAD will be employed to systematically address the design of the SIEM system using Microsoft Sentinel for scheduled task threat detection in a cloud environment. The system's architecture will be visualized and documented using various analysis diagrams:

* **Use Case Diagrams**: These diagrams will illustrate the interactions between different users (actors) and the system, highlighting the various use cases or functionalities that the system must support. For example, a use case diagram may show how a security analyst interacts with the system to review alerts or isolate a compromised VM.
* **Class Diagrams**: These diagrams will define the static structure of the system by showing its classes, attributes, operations, and the relationships between objects. For instance, classes might include "LogEntry," "Alert," "Playbook," and "VirtualMachine," each with specific attributes and methods that define their behavior and interactions.
* **Sequence Diagrams**: These diagrams will detail the dynamic behavior of the system by illustrating how objects interact in a particular sequence to perform a specific functionality. For example, a sequence diagram may depict the process flow from detecting a suspicious scheduled task to isolating the affected VM.

**System Architecture**

The architecture of the SIEM using Microsoft Sentinel is designed to leverage cloud-native services for log ingestion, analysis, and threat response. The primary components and their interactions are as follows:

1. **Log Collection**: Various Azure services, including VMs, Network Security Groups (NSGs), and Azure Active Directory (AAD), generate logs that capture security events, network traffic, and user activities. These logs are collected and sent to Microsoft Sentinel.
2. **Log Ingestion and Storage**: Microsoft Sentinel ingests these logs into a centralized Log Analytics workspace. This workspace serves as a repository for all collected log data, allowing for efficient querying and analysis.
3. **Data Analysis and Threat Detection**: Sentinel uses configured analytics rules to process and analyse the ingested log data. These rules, written in Kusto Query Language (KQL), are designed to identify suspicious activities, such as unusual scheduled task creations or modifications.
4. **Alert Generation**: When an analytics rule detects a potential threat, Sentinel triggers an alert that is configured. This alert contains detailed information about the suspicious activity and triggers a response mechanism.
5. **Automated Response**: The alert activates a playbook implemented using Azure Logic Apps. The playbook automates response actions, such as notifying security personnel, enriching alert data, and initiating containment procedures.
6. **VM Isolation**: As part of the containment process, the playbook isolates the compromised VM by deactivating its network interface card (NIC), applying restrictive NSG rules, and removing the VM from any load balancers. This action effectively contains the threat and prevents further malicious activities.

Using SSAD and cloud-native tools, the system is designed to be scalable, maintainable and efficient in detecting and mitigating scheduled task threats within a cloud environment.

### Tools and Techniques

Key tools and technologies to be used include:

1. **Microsoft Sentinel**: This is ideal because of its cloud-native SIEM capabilities, Sentinel offers the ability to implement threat detection and automated response mechanisms, and seamless integration with other Azure services. Its ability to analyse large volumes of data in real-time makes it an ideal tool for this project.
2. **Azure Logic Apps**: These will be configured to automate responses to detected threats. Logic Apps allow the creation of automated workflows that can trigger the isolation of compromised virtual machines, notify security personnel, and perform other remediation actions without manual intervention.
3. **Azure Monitor**: This tool will be used for collecting, analysing, and acting on data from the cloud and on-premises environments. Azure Monitor will help in tracking the performance and health of the solution, ensuring that it operates efficiently.
4. **Kusto Query Language (KQL)**: KQL is used within Sentinel to create custom analytics rules and queries for detecting suspicious scheduled task activities. Its powerful querying capabilities enable detailed analysis of log data to identify patterns indicative of security threats.

## Testing

To confirm that the solution is fit for purpose, a testing strategy will be used to ensure the system can detect and mitigate threats originating from scheduled tasks. The testing process will involve two types of tests, each designed to validate different aspects of the system.

1. **Integration Testing**: This will involve testing the interaction between different system components to ensure they work seamlessly together. The integration between the virtual machine, Microsoft Sentinel, and Azure Logic Apps will be tested to verify that logs are correctly ingested, analysed, and trigger appropriate responses. Integration testing ensures that the system's components interact correctly and that data flows smoothly between them. **The specific type of integration testing to be done include:**

* Execute tasks to show that logs are being collected using log analytics workspace and displayed and analysed in sentinel.
* Execute a malicious task to show that the alert generated using sentinel triggers a response in azure logic apps.

1. **Scenario Testing**: To test whether our system works as intended, both benign and malicious scheduled tasks will be executed. This will include tasks designed to mimic common attack patterns, such as persistence mechanisms and data exfiltration attempts. The system will be observed to determine whether it can accurately detect and differentiate between benign and malicious activities. Additionally, whether the compromised virtual machine (VM) is correctly isolated as part of the automated response will be tested. The specific scenarios to be tested include:

* **Executing a benign task to determine whether it is detected as benign and a malicious task to determine whether it is detected a malicious.**
* **Executing a malicious task to determine whether the response action is performed that is isolation of the compromised virtual machine**

## Deliverables

The deliverables of this project are to:

1. Establish a secure connection between the VMs and Microsoft Sentinel Configure VMs to generate and send security events, network traffic, and user activity logs to Microsoft Sentinel
2. Create and configure a centralized Log Analytics workspace in Microsoft Sentinel to store and manage the ingested log data.
3. Develop and configure analytics rules using Kusto Query Language (KQL) in Microsoft Sentinel to identify suspicious activities and differentiate between malicious and benign scheduled tasks.
4. Set up alerts in Microsoft Sentinel to trigger when analytics rules detect potential threats.
5. Create and configure a playbook using Azure Logic Apps to automate response actions.
6. Test the system by simulating the triggering event, execution of a malicious scheduled task detected as malicious.
7. Proposal Document: A detailed proposal outlining the project scope, objectives, methodology, and expected outcomes.

# System Analysis and Design

## Introduction

This chapter explains in detail the analysis and design of the SIEM system focusing on the requirements, architecture and diagrams that represent the system’s structure, how the system is designed to function, how different components interact, and how data flows through the system. These diagrams include system architecture diagram, network diagram, use case diagram, sequence diagram and flowchart diagram. Functional and non-functional requirements are outlined, providing a clear understanding of the expected capabilities and performance standards of the system.

## System Requirements

### Functional Requirements

The functional requirements are the operations and behaviours that the system must support. These include:

1. **Logging and Reporting**: The system must log all detected events related to scheduled tasks and generate reports for further analysis. This includes detailed logs of task executions, detections and any mitigation actions taken.
2. **Detection of Malicious Scheduled Tasks**: The system should be able to detect and identify malicious scheduled tasks within the azure environment. This involves monitoring and analysing logs generated by scheduled tasks and differentiating between benign and malicious scheduled tasks to determine any suspicious activity.
3. **Automated Response and Mitigation**: On detecting a malicious scheduled tasks the system should automatically trigger a response, which is isolating the compromised virtual machine to prevent further damage.
4. **Integration with Microsoft Sentinel**: The virtual machines and azure logic apps should integrate with Microsoft Sentinel to take advantage of its SIEM capabilities including real-time monitoring, threat detection and response strategies.

### Non-Functional Requirements

The non-functional requirements are the quality attributes and performance criteria the system must meet. These include:

1. **Performance**: The system should provide real time monitoring and analysis of scheduled tasks with minimal latency, ensuring timely detection and response.
2. **Accuracy**: The detection algorithm should have a high accuracy rate to minimize false positives and false negatives in identifying malicious tasks.
3. **Reliability**: The system should be reliable with minimal downtime and should be able to handle large volumes of log data without failure.
4. **Flexibility**: The system should be flexible enough to adapt to changes in the cloud environment such as new types of scheduled tasks.
5. **Continuous Monitoring**: The system should ensure continuous monitoring for early threat detection and faster incident response.

## System Architecture Diagram

The system architecture diagram provides an overview of how various components of the system interact and work together. It includes the Azure environment, the primary cloud infrastructure where the VMs with scheduled tasks and other resources are hosted. Microsoft Sentinel that integrates with the VMs, Log analytic workspace and Azure logic app for real-time threat detection, logging and automated response. Custom scripts written in sentinel are used to differentiate between malicious and benign tasks and trigger an alert in case of a malicious task. Finally, Azure logic app contains a custom playbook that triggers mitigation actions like isolating a compromised VM when an alert of a malicious task is made.

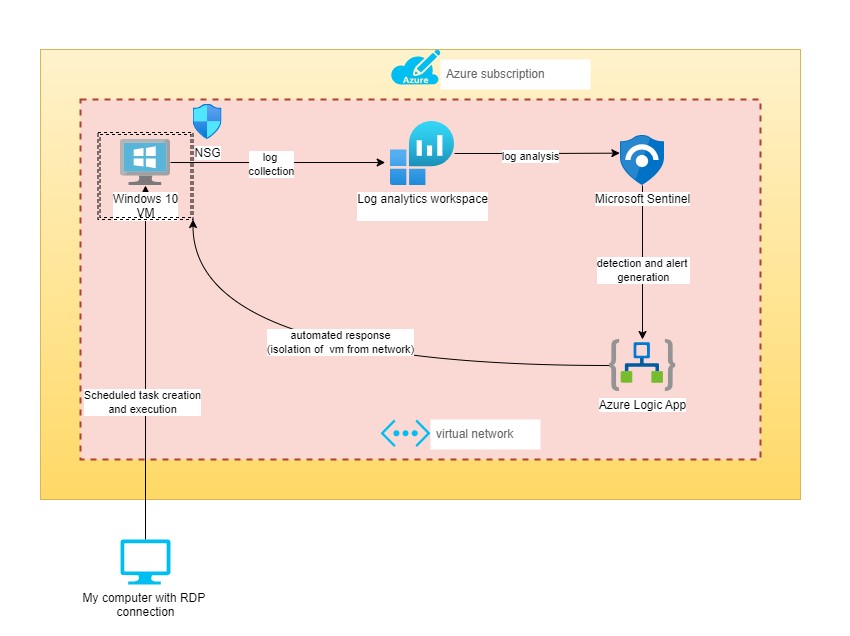


Figure 4.1 System architecture diagram

## System Analysis Diagrams

### Network Diagram

Figure 4.2 below shows the network topology of how the project is set up on Azure. When a virtual machine is deployed in Azure, that virtual machine is placed on a Virtual Network (Vnet). And assigned an IP address on that network as well as a network interface. An NSG (Network Security Group) is used to filter network traffic to and from Azure resources.

A diagram of a cloud

Description automatically generated

Figure 4.2 Network Diagram

### Use Case Diagram

Figure 4.3 represents the interaction between users, system administrators and the system. Users can schedule tasks while administrators can use the system to monitor the scheduled tasks within the environment. Administrators can access detailed logs and reports generated by the system, they can update and manage detection rules for identifying malicious scheduled tasks and can update playbooks for isolation of compromised VMs.

A diagram of a schedule

Description automatically generated

Figure 4.3 Use Case Diagram

### Sequence Diagram

Figure 4.4 visualizes the interaction between different components of the system during the detection and mitigation of malicious scheduled tasks. It includes log collection from the VMs, log ingestion, analysis and storage, Microsoft Sentinel uses these logs from the centralized Log Analytics workspace. Data analysis and threat detection occur in Microsoft sentinel using configured rules, alert generation when a threat is detected, and finally automated response implemented by a custom playbook triggered by the alert and finally VM isolation.

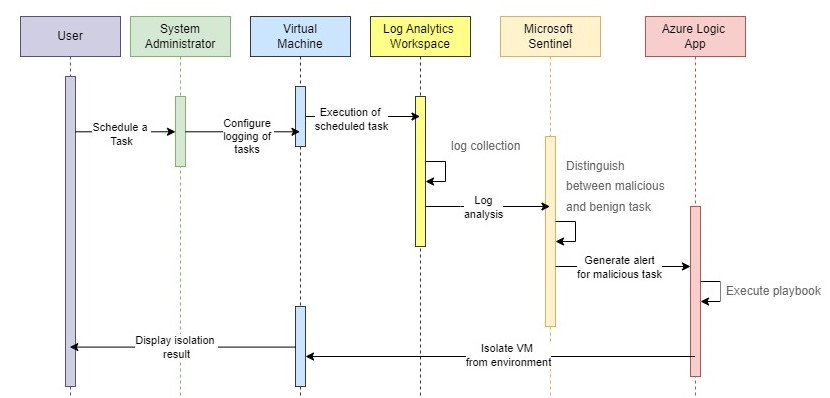


Figure 4.4 Sequence Diagram

### Flow Chart

Figure 4.5 provides a view of the data flow within the system. This involves data collection: the flow of log data from VMs to the detection scripts, detection process, the steps taken to analyse the data and identify potential threats, decision points, key decision points where the system determines whether a task is malicious or benign, and finally response actions, the flow of actions taken in response to detected threats, including the isolation of compromised VMs.

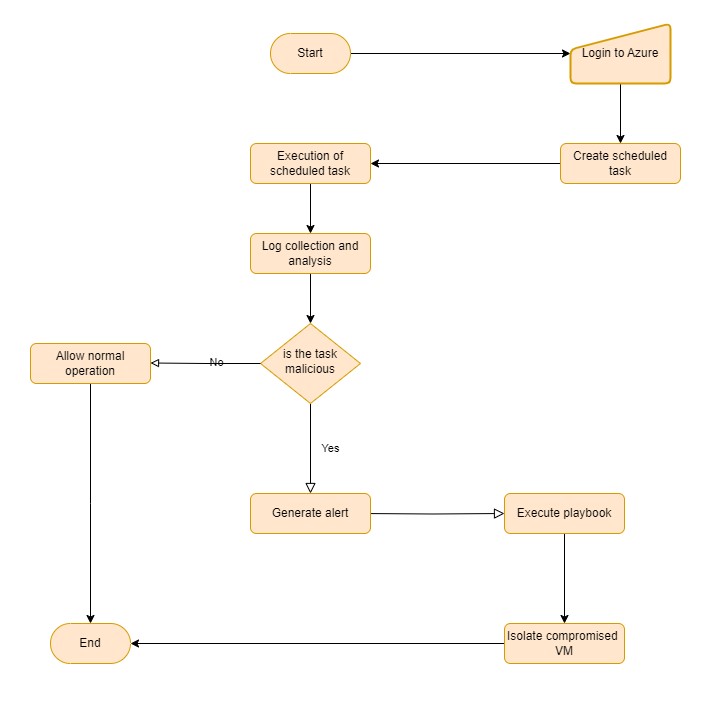


Figure 4.5 Flowchart diagram

### State Transition Diagram

Figure 4.6 below describes the different states the SIEM system goes through. From idle to log collection when tasks are executed on the VM and logs are collected. This is followed by analyzing logs to threat detection. After a threat is detected, the state moves from threat detected to alert generation and finally VM isolation.

A diagram of a computer system

Description automatically generated

Figure 4.6 State Transition Diagram

References

Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile software development: Methodologies

and trends. International Journal of Interactive Mobile Technologies, 14(11).

Ananthapadmanabhan, A., & Achuthan, K. (2022, June). Threat modelling and threat intelligence

system for cloud using splunk. In 2022 10th International Symposium on Digital Forensics

and Security (ISDFS) (pp. 1-6). IEEE.

Barhami, Y. E., Aroussi, H. K., & Bensaid, C. (2022, May). From Firewall and Proxy to IBM

QRadar SIEM. In International Conference on Advanced Intelligent Systems for

Sustainable Development (pp. 199-204). Cham: Springer Nature Switzerland.

Copeland, M., & Copeland, M. (2021). Cloud Defense Strategies with Azure Sentinel. Apress.

Dennis, A., Wixom, B., & Tegarden, D. (2015). Systems analysis and design: Object-oriented

approach with UML. John wiley & sons.

Diogenes, Y., DiCola, N., & Turpijn, T. (2022). Microsoft Azure Sentinel: Planning and

implementing Microsoft's cloud-native SIEM solution. Microsoft Press.

Jang-Jaccard, J., & Nepal, S. (2018). A survey of emerging threats in cybersecurity. Journal of

Computer and System Sciences, 80(5), 973-993.

Kannan, Govindarajan & Kamburugamuve, Supun & Wickramasinghe, Pulasthi & Abeykoon,

Vibhatha & Fox, Geoffrey. (2017). Task Scheduling in Big Data - Review, Research

Challenges, and Prospects. 165-173. 10.1109/ICoAC.2017.8441494.

Kavanagh, K., & Rochford, K. (2020). Magic Quadrant for Security Information and Event

Management. Gartner

Lee, K., Lee, J., & Yim, K. (2023). Classification and analysis of malicious code detection

techniques based on the APT attack. Applied Sciences, 13, 2894.

<https://doi.org/10.3390/app13052894>

L. Xiao, D. Xu, C. Xie, N. B. Mandayam and H. V. Poor, "Cloud Storage Defense Against

Advanced Persistent Threats: A Prospect Theoretic Study," in IEEE Journal on

Selected Areas in Communications, vol. 35, no. 3, pp. 534-544, March 2017, doi:

10.1109/JSAC.2017.2659418.

Saad, A., Hambira, N., Zin, H. M., Johan, R., & Li, W. S. (2019). Object oriented VS structured

analysis and design in system development courses. Int. J. Eng. Adv. Technol., 8(6).

Sekharan, S. S., & Kandasamy, K. (2017, March). Profiling SIEM tools and correlation engines

for security analytics. In 2017 International Conference on Wireless Communications,

Signal Processing and Networking (WiSPNET) (pp. 717-721). IEEE.

Sharma, M., Kumar, M., & Samriya, J. K. (2022). An optimistic approach for task scheduling in

cloud computing. International Journal of Information Technology, 14(6), 2951-2961.

Sharma, A., Gupta, B.B., Singh, A.K. et al. Advanced Persistent Threats (APT): evolution,

anatomy, attribution and countermeasures. J Ambient Intell Human Comput 14, 9355–

9381 (2023). https://doi.org/10.1007/s12652-023-04603-y

Tuyishime, E., Balan, T. C., Cotfas, P. A., Cotfas, D. T., & Rekeraho, A. (2023). Enhancing Cloud

Security—Proactive Threat Monitoring and Detection Using a SIEM-Based Approach.

Applied Sciences, 13(22), 12359.

Uçar, M. (2023). Advanced Persistent Threats.

Verma, A., Malla, D., Choudhary, A. K., & Arora, V. (2019). A detailed study of Azure platform

& its cognitive services. In \*2019 International Conference on Machine Learning, Big Data,

Cloud. IEEE. <https://doi.org/10.1109/COMITCon.2019.8862178>

Zainudin, Z. S., & Abdul Molok, N. N. (2018). Advanced persistent threats awareness and

readiness: A case study in Malaysian financial institutions. In 2018 Cyber Resilience

Conference (CRC) (pp. 1-3). IEEE. <https://doi.org/10.1109/CR.2018.8626835>

Zannone, N., Zambon-Mazzocato, E., & Mostafaei, H. (2023). Security Analysis of Azure Active Directory Logs using Machine Learning.

Appendices

Appendix 1 Gantt Chart

A screenshot of a computer screen

Description automatically generated

**BSC. IN COMPUTER NETWORKS AND CYBER SECURITY**

**PROPOSAL MARKING GUIDE**

**CNS Project II**

**Section A – Student’s Details:**

|  |  |
| --- | --- |
| **Student Number** | **Student Name** |
|  |  |
| **Project Title:** |  |

**Section B – For examiner’s use only:**

|  |  |  |
| --- | --- | --- |
| **Area of Assessment** | **Score** | **Total** |
| **Title (2)** |  |  |
| - Is the title informative, concise, focused and appropriate? *(2)* |  |  |
| **Abstract (3)** |  |  |
| - Describes the concept clearly *(3)* |  |  |
| **Introduction (6)** |  |  |
| * Is there a clear identification of a specific problem that relates to a Computer Network and Cyber Security challenge? *(1)* * Are the objectives S.M.A.R.T and incremental? *(2)* * Do the objectives support the achievement of the project’s aim? *(1)* * Has the significance of the project and justification for undertaking the project been clearly explained? *(1)* * Is the scope achievable at the candidate’s level *(1)* |  |  |
| **Literature Review (11)** |  |  |
| * Has the objective-literature mapping (Literature Review sections to address each Specific Objective) been done correctly? *(3)* * Does the literature review analyse what has been done before in relation to the project, what has not been done (the gap), and what the project proposes to do to address the gap? *(3)* * Has adequate and up to date literature been reviewed? *(3)* * Is the conceptual framework supporting the concept discussed? *(2)* |  |  |
| **Design Methodology (6)** |  |  |
| * Has the correct/fitting design methodology been used. *(2)* * Presents practical implementation plan with a list of tools and systems that will be used *(2)* * Are the deliverables discussed and are they practical and achievable? *(2)* |  |  |
| **Overall Assessment (12)** |  |  |
| * Is there a logical mapping from the Title, to the Objectives, to the Literature Review topics, to the Methodology, to the proposed tools to be used, and lastly to the deliverables? *(3)* * Proposal document is neat and well formatted. Table of contents numbered in sequence and corresponds to page Numbers *(3)* * Evidence that figures were drawn by candidate and not copy pasted *(2)* * Has the APA referencing style been correctly applied? *(1)* * Demonstrates that candidate will learn from the project *(3)* |  |  |
| **TOTAL (40)** |  |  |

Page **1** of **2**

**Please provide further feedback for the candidate:**

**Examiner** *(initials)* **Date:**