**SYSTEM REQUIREMENTS**

**P06: ANOMALOUS LOGIN DETECTION USING ELK (SECURITY PROJECT)**

## <TEAM MEMBER NAMES & IDS>

|  |  |
| --- | --- |
| **STUDENT ID** | **NAME** |
| **26100015** | **MUHAMMAD AAFFAN KHAN NIAZI** |
| **26100286** | **MOHAMMAD MUSTAFA** |
| **25100022** | **SHEHROZ FARYAD** |
| **26100399** | **MUSTAFA HUSSAIN** |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Content** | **Totals** | **Obtained** |
| Introduction | 5 | 5 |
| System actors | 10 | 10 |
| Functional Requirements | 30 | 20 |
| Non-functional requirements | 10 | 10 |
| Security requirements | 30 | 30 |
| Use of Generative AI | 5 | 0 |
| Who did what | 5 | 5 |
| Review checklist | 5 | 5 |
| GitHub folder structure penalty  (if not created properly) | -20 |  |
| Late submission penalty | -20 |  |
| **Grand Total** | **100** | **85** |
| **General Comments: rename your github repo with the following convention “P06-ProjectName”. Also follow the naming convention for document like “P06\_SystemRequirement”** | | |

## TABLE OF CONTENTS

1. Introduction............................................................................................................................. 3
2. System Actors..........................................................................................................................4
3. Functional Requirements.........................................................................................................5
4. Non-functional Requirements / Quality Attributes................................................................. 6
5. Security Requirements.............................................................................................................7
6. Security Engineers...................................................................................................................8
7. Use of Generative AI...............................................................................................................9 8. Who Did What?.....................................................................................................................10 9. Review checklist....................................................................................................................10

# 1. Introduction

This project focuses on detecting anomalous login activities using the ELK (Elasticsearch, Logstash, Kibana) stack with Wazuh integration. The system is designed to provide real-time anomaly detection, visualization, and alerting for suspicious logins across different platforms (Windows/Linux). The primary users include Security Engineers, SOC Analysts, and System Administrators who require efficient detection and response capabilities to safeguard against brute force, privilege escalation, and unusual login behaviors.

Each section is too brief. Please rethink and add more details in each section.

# 2. System Actors

<List down the names of the system actors and give a 2-3 lines description of the role of each actor>

|  |  |
| --- | --- |
| **Actor Name** | **Description** |
| Security Engineer | Configures detection rules, reviews anomalies, manages alerts. |
| SOC Analyst | Monitors dashboards, investigates anomalies, and responds to incidents. |
| System Administrator | Provides system-level logs, manages user accounts, and oversees infrastructure security. |
| End User | Generates login activity that the system monitors for anomalies. |

Add more functional requirements like:

* SOC Analyst: “I want to investigate anomalies with Wazuh reports so I can identify brute force patterns.”
* End User: “I want to receive anomaly alerts so I can secure my account.”

# 3. Functional Requirements

<Write system requirements from users’ (actors) perspective. Actor names have been highlighted in the sample requirements below. >

|  |  |
| --- | --- |
|  | **Requirements of Security Engineer** |
| **Sr#** | **Requirement** |
| 1 | I want to define anomaly detection thresholds so that suspicious logins can be flagged. |
| 2 | I want to manage detection rules to adapt to emerging attack vectors. |
| 3 | I want to view dashboards for anomaly trends to support investigations. |

|  |  |
| --- | --- |
|  | **Requirements of System Administrator** |
| **Sr#** | **Requirement** |
| 1 | I want to provide authentication logs to ensure comprehensive monitoring. |
| 2 | I want to manage accounts and permissions to reduce exposure to malicious logins. |

|  |  |
| --- | --- |
|  | **Requirements of End User** |
| **Sr#** | **Requirement** |
| 1 | I want my login activity to be monitored to protect my account from unauthorized access. |

# 4. Non-functional Requirements / Quality Attributes

<Requirements must be testable>

<Security requirements fall in the category of “Non-functional requirements”; however, you need to list them separately in the section **Security Requirements** later in this document.>

|  |  |
| --- | --- |
| **Sr#** | **Requirements** |
| 1 | The system should not utilize more than 1 GB of memory at any time during its execution. |
| 2 | The system should not fail more than 3 times every 24 hours; if it does, it should recover within 5 minutes. |
| 3 | The system should be able to process at least 10,000 log entries per second without performance degradation. |
| 4 | The alerting mechanism should deliver notifications within 30 seconds of anomaly detection. |

# 5. Security Requirements

< Go through OWASP top 10 security risks in the following categories: I. OWASP Top Ten:<https://owasp.org/www-project-top-ten/>

II. OWASP Mobile Top 10:<https://owasp.org/www-project-mobile-top-10/> III. OWASP Machine Learning Security Top Ten:

<https://owasp.org/www-project-machine-learning-security-top-10/>

IV. OWASP Top 10 API Security Risks:<https://owasp.org/API-Security/editions/2023/en/0x11-t10/>

1. Select **security risks** that you think are primary threats for your system. While doing this, carefully consider the information/functionality that is most vulnerable from security perspective in the context of your project.
2. For each security risk (identified above), identify **potential losses** (e.g., financial loss, total business loss, litigation etc.) if you do not take necessary measures to address the identified security risks.
3. Identify the **controls** (e.g., input validation, audit logs, multi-factor authentication, user roles etc.) that should be implemented in your system to address the identified security risks.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr #** | **Security Risks** | **Potential Losses** | **Controls** |
| 1 | Broken Access Control | Sensitive user login data is exposed. | Only security engineers will have update rights. |
| 2 | Input Manipulation Attack | Repeated attempts can cause the system to ignore the real threats. | Simulate the fake inputs to ensure the system’s accuracy remains unaffected. |
| 3 | Data Poisoning Attack | Can cause the system’s detection quality to drop over time. | Pre-process the data. |
| 4 | Using outdated versions of client-side and/or server-side components | Exploitation of known vulnerabilities in the older versions. | Obtain components from  their official links |
| 5 | Hardcoding credentials | Credentials are exposed so hackers may be able to gain access | A security testing process would take place in order to ensure credentials are not exposed in such ways. |

## 6. Security Engineer

**<** Each team must designate one member as the **Security Engineer**. While the entire team is responsible for implementation of the project's security features, the Security Engineer will take the lead in overseeing and ensuring the overall security of the project. **>**

|  |  |
| --- | --- |
| **Name of the Security Engineer** | Muhammad Aaffan Khan Niazi |

# 7. Use of Generative AI

<Mention here how generative AI was used in preparation of this artifact.>

# 8. Who Did What?

|  |  |
| --- | --- |
| **Name of the Team Member** | **Tasks done** |
| Affan | Introduction, Actors |
| Mustafa Hussain | Functional requirements, review |
| Muhammad Mustafa | Security Requirements |
| Shehroz | Non Functional requirements |

# 9. Review checklist

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

|  |  |
| --- | --- |
| **Section** **Title** | **Reviewer Name(s)** |
| Introduction | Muhammad Aaffan Khan Niazi |
| Actors | Muhammad Aaffan Khan Niazi |
| Functional Requirements | Mustafa Hussain |
| Non-functional requirements | Shehroz Faryad |
| Security Requirements | Mohammad Mustafa |
| Use of Generative AI |  |