



**SHAHEED ZULFIKAR ALI BHUTTO  
INSTITUTE OF SCIENCE AND TECHNOLOGY**

---

## **FINAL YEAR PROJECT**

---

# **Security Process Fusion Through Automation**

---

**Project Team:**

**Umair Khan 2012412**

**Ali Iqbal Rashid 2012286**

**Project Supervisor:**

**Dr. Husnain**

**Date: 29<sup>th</sup> September, 2024**

**Submitted in partial fulfillment of the requirements for the degree of  
Bachelor of Science in Computer Science in the  
Faculty of Computing and Engineering Sciences  
Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology (SZABIST)  
Karachi Campus**

## Plagiarism free certificate

This is to certify that we, **Umair Khan** S/O Deedar Ali and **Ali Iqbal Rashid** S/O Iqbal Rashid, are the members of FYP group **Security Process Fusion Through Automation** under registration numbers **2012412** and **2012286** respectively, at the Department of Computer Science at SZABIST, Karachi. We certify that our FYP documentation has been reviewed by our advisor and the work presented is our own.

Name of Advisor: Dr. Husnain  
Designation: Associate Professor

Signature: \_\_\_\_\_

PAPER NAME

AUTHOR

**Security Process Fusion Through Autom  
ation****-**

WORD COUNT

**4983 Words**

CHARACTER COUNT

**32281 Characters**

PAGE COUNT

**49 Pages**

FILE SIZE

**37.3KB**

SUBMISSION DATE

**Sep 29, 2024 10:20 PM GMT+5**

REPORT DATE

**Sep 29, 2024 10:20 PM GMT+5****● 13% Overall Similarity**

The combined total of all matches, including overlapping sources, for each database.


- 11% Internet database
- 4% Publications database
- 0% Submitted Works database

# 13% Overall Similarity

Filters

Match Groups

Sources

Show overlapping sources 



1

Internet



**www.coursehero.com**

4%

 9 text blocks  221 matched words

2

Publication



**Kevin Lynn McLaughlin. "Cybersecurity Operations..."** 1%

 9 text blocks  59 matched words

3

Internet



**discuss.elastic.co**

1%

 1 text block  53 matched words

## **Declaration of Authorship**

We, Umair Khan (2012412) and Ali Iqbal Rashid (2012286), declare that this report titled, “Security Process Fusion Through Automation” and the work presented in it are our own. We confirm that: This is solely for the purpose of the completion of our bachelor’s degree at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology. This report has not been submitted to any university before this and where use of external text exists, that has been clearly stated. The source of references has been given where applicable. With the exception of such quoted texts, this report contains text from our own work. Where the report is based on work done by us jointly with others, we have clearly stated who has contributed to what area of the report.

Signed:

Umair Khan (2012412)

Ali Iqbal Rashid (2012286)

---

Date: 29th September, 2024

## **Project Description**

The rapid increase in cyber threats and the complexity of managing security incidents necessitate a robust and efficient incident response framework. Security Orchestration, Automation, and Response (SOAR) offers a comprehensive solution to streamline and enhance incident response processes. This introduction serves as a guide to understand the importance of implementing a SOAR project.

## Acknowledgements

In the name of ALLAH, the most beneficent and merciful, who gave us the knowledge and courage to work on this project.

We are grateful for the outcome and success of this project over the year are gratitude towards the people who have provided us with the guidance and assistance to be able to complete this project in such a difficult time.

We would like to thank our supervisor “Dr. Husnain” of the Computer Science faculty at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology. He was integral part in the project as he was always there when we would get stuck at a point in our project. He consistently guided us, motivated us and cooperated with us throughout the duration of this project.

We would like to thank to the teachers at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology, who guided us and taught us throughout our time in the university. We would also like to express our gratitude to our parents and family members who helped and encouraged us during this time. Furthermore, we would like to thank the staff at SZABIST for allowing us to use their labs and services to be able to complete the project.

Lastly, we would like to extend our gratitude to everyone at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology for creating an environment for students to thrive in. The quality of education, the cooperative faculty members and the motivation provided by them.

# Table of Contents

Plagiarism free certificate .....	2
Declaration of Authorship.....	5
Project Description.....	6
Acknowledgements .....	7
Project Proposal .....	11
1. Introduction .....	12
2. Objective.....	12
3. Problem Description .....	12
4. Methodology.....	13
5. Project Scope .....	13
6. Feasibility Study .....	13
<b>7. Solution Application Areas .....</b>	<b>14</b>
8. Tools/Technology .....	14
9. Expertise of the Team Members.....	15
10. Milestones .....	15
11. Project Schedule.....	15
12. References .....	16
Security Process Fusion through Automation.....	18
1, Introduction.....	19
1.1. Purpose.....	19
1.2. Document Conventions.....	19
1.3. Intended Audience and Reading Suggestions.....	19
1.4. Product Scope .....	19
1.5. References.....	20
2. Overall Description.....	20
2.1. Product Perspective.....	20
2.2. Product Functions .....	20
2.3. User Classes and Characteristics .....	21
2.4. Operating Environment.....	21
2.5. Design and Implementation Constraints.....	21
2.6. User Documentation .....	21
2.7. Assumptions and Dependencies .....	22



3.	External Interface Requirements .....	22
3.1.	User Interfaces .....	22
3.2.	Hardware Interfaces .....	22
3.3.	Software Interfaces .....	23
3.4.	Communications Interfaces .....	23
4.	System Features .....	23
5.	Other Nonfunctional Requirements .....	25
5.1	Performance Requirements .....	25
5.2	Safety Requirements .....	25
5.3	Security Requirements .....	25
5.4	Software Quality Attributes .....	25
5.5	Business Rules.....	25
6.	Other Requirements .....	26
	Software Design.....	27
	Specification .....	27
1.	Introduction: .....	28
1.1	Purpose of this document: .....	28
1.2	Scope of the development project: .....	28
1.3	Definitions, acronyms, and abbreviations: .....	28
<b>1.4.</b>	<b>References:</b> .....	28
1.5.	Overview of document:.....	29
2.	System architecture description:.....	29
2.1	Section Overview: .....	29
2.2	General Constraints: .....	29
2.3.	Data Design:.....	29
2.4.	Program Structure: .....	30
3.	Detailed description of components: .....	31
3.1	Section Overview: .....	31
3.2	SDS component table .....	31
4.	User Interface Design: .....	32
4.1	Section Overview: .....	32
4.2	Interface Design Rules: .....	32
4.3.	GUI Components: .....	32
4.3.1.	Analytics: .....	32

4.3.2. Observability: .....	32
4.3.3. Security: .....	32
4.3.4. Management: .....	33
4.4. Detailed Description: .....	33
5. Reuse and relationships to other products: .....	53
6. Design and Tradeoffs: .....	54
7. Pseudocode for components: .....	54
8.1. Class Diagram: .....	55
8.2. Object Diagram: .....	55
8.3. State chart Diagram: .....	55
8.4. Activity Diagram: .....	56
8.5. Sequence Diagram: .....	58
8.6. Collaboration Diagram: .....	59
8.7. Deployment diagram: .....	61
8.8. System Block Diagram: .....	62
Test Cases: .....	63
Advisor Form: .....	67

# Security Process Fusion Through Automation

## Project Proposal

Supervisor

Supervisor Name

Dr. Husnain

Submitted by

Umair Khan

2012412

Ali Iqbal

2012286

**Faculty of Computing and Engineering Sciences**

Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology, Karachi.

[May 30<sup>th</sup>, 2023]

# 1. Introduction

The rapid increase in cyber threats and the complexity of managing security incidents necessitate a robust and efficient incident response framework. Security Orchestration, Automation, and Response (SOAR) offers a comprehensive solution to streamline and enhance incident response processes. This introduction serves as a guide to understand the importance of implementing a SOAR project.

# 2. Objective

Our project objective is to develop and implement an affordable and efficient SOAR (Security Orchestration, Automation, and Response) platform specifically designed for mid-level industries. The key goals include enhancing cybersecurity capabilities by automating incident response, threat detection, and remediation processes. We aim to provide enterprise-level functionalities at a lower cost, enabling mid-level industries to access advanced security orchestration and automation within their budget constraints. The platform will enable lightning-fast threat detection and response, streamlined incident management, seamless integration with existing security tools, customization and scalability to accommodate diverse industry needs, and comprehensive knowledge transfer for user empowerment. Through continuous innovation, we strive to equip organizations with the tools they need to proactively safeguard their operations and assets from evolving cyber threats.

# 3. Problem Description

- **Inadequate cybersecurity infrastructure:** Mid-level industries lack the necessary resources and financial capacity to invest in comprehensive cybersecurity solutions, including SOAR platforms.
- **Limited availability of affordable SOAR solutions:** The current market primarily caters to larger enterprises, leaving mid-level industries without cost-effective options for implementing a SOAR platform.
- **Increased vulnerability to cyber threats:** Without a robust SOAR solution, these organizations face significant cybersecurity vulnerabilities, making them attractive targets for cybercriminals.
- **Manual incident response processes:** Due to the absence of automation and orchestration capabilities, mid-level industries heavily rely on manual interventions, leading to delays in threat detection and response.
- **Inefficient utilization of security tools:** The lack of a comprehensive SOAR platform hinders the effective integration and management of various security tools, resulting in suboptimal utilization of available resources.

- Limited threat intelligence integration: Mid-level industries struggle to incorporate up-to-date threat intelligence feeds, making it difficult to identify and respond to emerging threats effectively.

## 4. Methodology

Our methodology for the SOAR (Security Orchestration, Automation, and Response) project involves a systematic approach to ensure successful development and implementation of the platform. We begin by gathering comprehensive requirements through discussions and interviews with stakeholders, followed by designing a scalable and flexible architecture. The development phase involves coding, testing, and integration with existing systems, while data integration and configuration ensure seamless ingestion and processing of security event data. Automation rules and playbooks are created to enable rapid threat detection and response. Rigorous testing and quality assurance measures are implemented to ensure the platform's functionality, performance, and security. Once tested, the platform is deployed, and comprehensive training and documentation are provided to users. Continuous improvement is prioritized through user feedback, updates, and integration of new technologies. Throughout the process, we adhere to project management best practices, employing agile methodologies for effective collaboration, communication, and risk management.

## 5. Project Scope

Our project is focused on delivering lightning-fast malware detection and remediation capabilities on an enterprise level. With the use of advanced technologies, we aim to quickly identify and respond to malware threats, minimizing their impact on the organization's systems and data. Additionally, we will employ rapid analysis techniques to identify and mitigate phishing emails, ensuring that employees are protected from falling victim to these deceptive attacks. Furthermore, our platform will be equipped to swiftly detect and address network-level Nmap scanning activities, enabling proactive defense against potential vulnerabilities. By offering these lightning-speed detection and remediation features, our project aims to significantly enhance the cybersecurity defenses of organizations, safeguarding their digital assets and maintaining a secure operational environment.

## 6. Feasibility Study

With above defined scope, we should be able to meet our project schedule

**Risks Involved:** Developing a SOAR project involves risks that need to be considered and addressed. These include the lack of stakeholder support, inadequate requirements analysis, integration challenges, data security and privacy risks, performance and scalability issues, skill and knowledge gaps, resistance to change, regulatory and compliance considerations, project delays and budget overruns, and vendor dependency. Mitigating these risks requires engaging stakeholders, conducting thorough requirements analysis, addressing integration complexities, implementing robust security measures, ensuring performance and scalability, addressing skill gaps, managing change effectively, complying with regulations, monitoring project timelines and

budgets, and managing vendor relationships. Proactive risk management throughout the project lifecycle is vital for successful implementation and deployment of the SOAR platform.

**Resource Requirement:** The minimum system requirements for a SOAR project can vary based on the specific software and technologies used. Generally, it is recommended to have a multi-core processor, a minimum of 8 GB RAM, sufficient storage space, a stable network connection, and an operating system compatible with the chosen SOAR platform. Consideration should also be given to software dependencies, virtualization requirements if applicable, and implementing backup and recovery mechanisms. It is important to consult the documentation and system requirements provided by the selected SOAR platform or software vendor for precise guidance on system configuration.

## 7. Solution Application Areas

Yes, our project is of real value, and we are targeting mid-level organizations which cannot afford expensive software. Those organizations will be able to get our software in much cheaper cost and will be able to prevent cyber-attacks much faster

The aims of a SOAR implementation revolve around improving incident response processes, enhancing security operations, and strengthening overall cybersecurity posture. Here are the key aims and objectives of our project

- Streamline Incident Response
- Accelerate Response Time
- Automate Routine and Manual Tasks
- Enhance Visibility and Situational Awareness

## 8. Tools/Technology

Here are some softwares that we will be using in our project:

### **SOAR Technology:**

- Opsgenie
- Shuffle

### **SIEM Tools:**

- Elastic stack

### **EDR Tools:**

- Elastic EDR

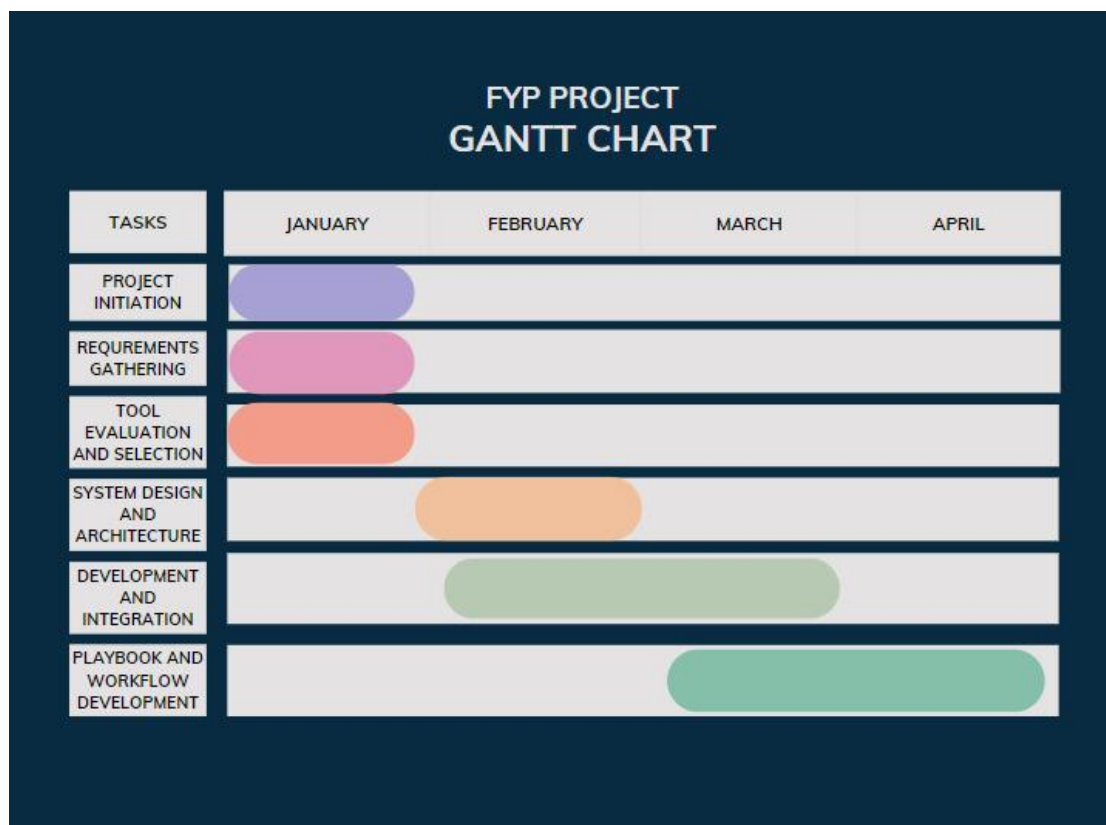
## 9. Expertise of the Team Members

We have enough knowledge about our project and we are still learning. We have done few courses from Coursera and other online platforms.

## 10. Milestones

- Implementing a prototype of a system that can achieve lightning-fast malware detection, phishing email analysis, and Nmap scanning detection with automated remediation is a great starting point for enhancing your organization's security capabilities.
- Paid subscription models can provide the necessary resources and funding to support ongoing development, maintenance, and integration of new technologies into the system.
- As part of the future work, we can consider incorporating additional technologies, such as advanced machine learning algorithms, threat intelligence platforms, behavior-based analytics, and advanced threat hunting techniques.
- Investing in the continuous improvement of the prototype and considering a paid subscription model will demonstrate your commitment to maintaining a robust security ecosystem.

## 11. Project Schedule





## 12. References

### Official Documentation and Websites of SOAR Platforms:

Demisto: <https://www.paloaltonetworks.com/products/cortex/demisto>

Splunk Phantom: [https://www.splunk.com/en\\_us/software/splunk-security-orchestration-automation-and-response.html](https://www.splunk.com/en_us/software/splunk-security-orchestration-automation-and-response.html)

Siemplify: <https://www.siemplify.co/>

### Cybersecurity News and Research Portals:

Security Intelligence: <https://securityintelligence.com/>

Dark Reading: <https://www.darkreading.com/>

SecurityWeek: <https://www.securityweek.com/>

### Industry Organizations and Communities:

SANS Institute: <https://www.sans.org/>

ISACA: <https://www.isaca.org/>

OWASP: <https://owasp.org/>

### Technology Blogs and Forums:

Medium (Security and Technology Categories): <https://medium.com/>

Stack Exchange (Security and IT Categories): <https://stackexchange.com/>



**Research Papers and Publications:**

IEEE Xplore: <https://ieeexplore.ieee.org/Xplore/home.jsp>

ACM Digital Library: <https://dl.acm.org/>

---

# **Software Requirements Specification**

**Security Process Fusion through Automation**

**Version 1.0 approved**

**Prepared by:  
Umair Khan 2012412  
Ali Iqbal 2012286**

**SZABIST**

**June 2024**

# **1, Introduction**

## **1.1. Purpose**

This project aims to develop a Security Orchestration, Automation, and Response (SOAR) system according to the unique needs of organizations. The project aims to enhance cybersecurity measures, streamline incident response processes, accelerate response times, protect sensitive data, ensure business continuity, and improve cost-effectiveness for organizations with limited financial resources. By implementing efficient incident response procedures and automating routine tasks, the project seeks to minimize the impact of security incidents, providing a cost-effective solution that safeguards critical information assets and strengthens the overall cybersecurity posture.

## **1.2. Document Conventions**

Main heading: bold

Font size heading1(18), heading2(14), font size (12)

Font: Times New Roman

Main heading: bold because the user can easily distinguish it from other

## **1.3. Intended Audience and Reading Suggestions**

This document is intended for developers, testers, users, and supervisors. This document contains several parts, an introduction, an overall description, external interface requirements, system features, and other non-functional requirements. By reading this document a reader can get a full idea of how this application works, and what things and ideas are implemented in creating this application.

## **1.4. Product Scope**

Our project is focused on delivering lightning-fast malware detection and remediation capabilities on an enterprise level. With advanced technologies, we aim to quickly identify and respond to malware threats, minimizing their impact on the organization's systems and data.

Additionally, we will employ rapid analysis techniques to identify and mitigate brute force attacks, ensuring that employees are protected from falling victim to these deceptive attacks. Furthermore, our platform will be equipped to detect and address malware detection. By offering these lightning-speed detection and remediation features, our project aims to significantly enhance the cybersecurity defenses of organizations, safeguarding their digital assets and maintaining a secure operational environment.

## 1.5. References

- <https://www.elastic.co/guide/en/cloud/current/ec-getting-started.html>
- <https://www.elastic.co/guide/en/security/current/install-endpoint.html>

## 2. Overall Description

### 2.1. Product Perspective

The system will be a central component within the organization's cybersecurity infrastructure. It serves as an integral part of the broader security ecosystem, interacting with various security tools, technologies, and processes.

### 2.2. Product Functions

- **Automated Remediation:** Upon detection of security threats, the system will automatically initiate remediation actions, such as isolating affected systems or quarantining malicious files.
- **Customizable Workflows:** Users can create and customize incident response workflows and playbooks to tailor responses to specific security requirements.
- **Automation Engine:** The system will automate routine and manual security tasks to improve efficiency and reduce response times.
- **Threat Intelligence Integration:** It will integrate with external threat intelligence sources to enhance threat detection and response.
- **User Access Control:** The system will manage user permissions, access control, and auditing to ensure secure system usage.
- **Reporting and Analytics:** Users will access reporting and analytics capabilities to gain insights into security incidents and system performance.

- **Notification and Alerting:** The system will send timely notifications and alerts to inform security teams about potential threats or incidents.
- **User-Friendly Interface:** The system will provide a user-friendly interface for efficient interaction, promoting ease of use and accessibility for security teams.

### **2.3. User Classes and Characteristics**

Security Analysts, System Administrators, Threat Intelligence Analysts, and End users will use this System. Each user class must have an active internet connection in order to access the system.

### **2.4. Operating Environment**

The minimum system requirements for a SOAR project can vary based on the specific software and technologies used. Generally, it is recommended to have a multi-core processor, a minimum of 8 GB RAM, sufficient storage space, a stable network connection, and an operating system compatible with the chosen SOAR platform. Consideration should also be given to software dependencies, virtualization requirements if applicable, and implementing backup and recovery mechanisms. It is important to consult the documentation and system requirements provided by the selected SOAR platform or software vendor for precise guidance on system configuration.

### **2.5. Design and Implementation Constraints**

- The server system must be running 24/7
- Users can access the system from any environment that is compatible with legacy systems.
- Interoperability of various tools should be considered
- Availability and reliability of external data sources can constrain integration.
- Internet connectivity is required.

### **2.6. User Documentation**

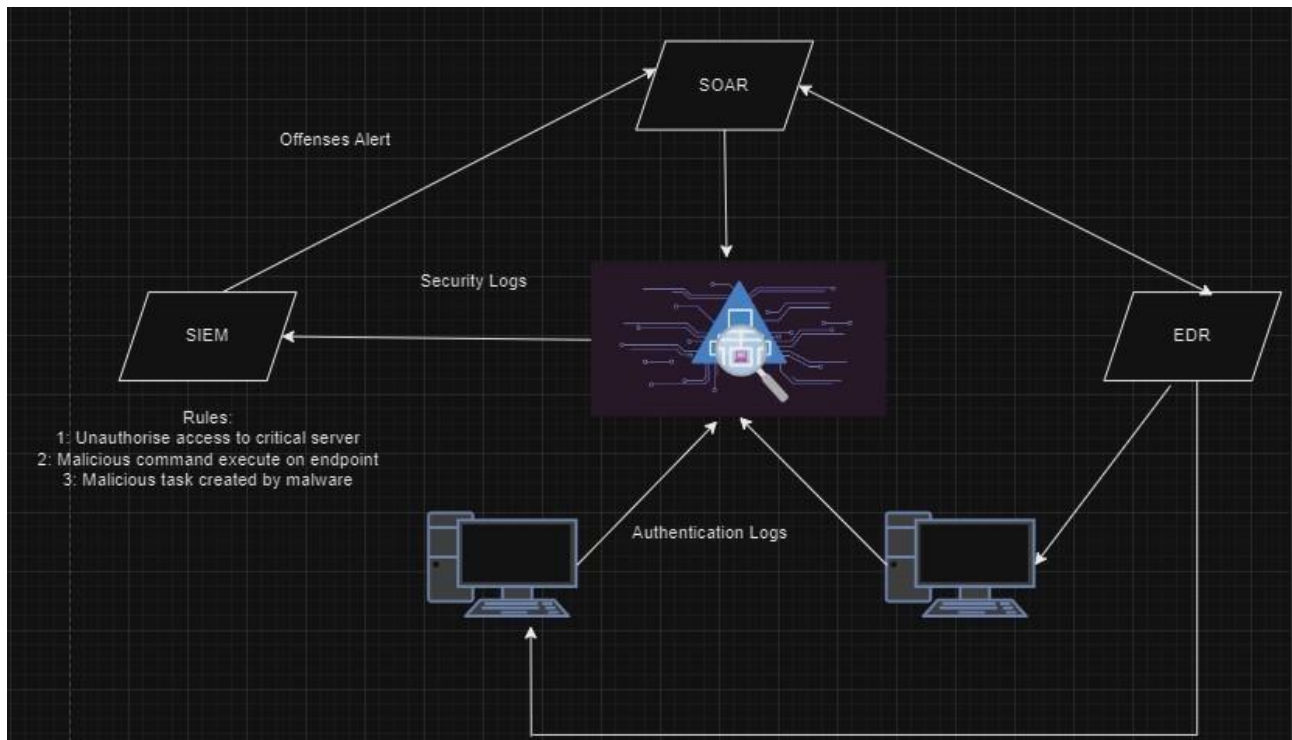
SRS, SDS, and Test Cases will be provided along with the application which will include all the technical design technical details of the application.

## 2.7. Assumptions and Dependencies

It is expected that users will have data availability for analysis and functional network infrastructure.

# 3. External Interface Requirements

## 3.1. User Interfaces



## 3.2. Hardware Interfaces

The hardware interface requirements for this project may include servers or virtual machines, multicore processors, at least 8 GB of RAM, sufficient storage, network hardware, backup solutions, security appliances, and in larger setups, high-availability configurations, and cloud resources. These components should align with the specific needs and the chosen SOAR platform's minimum requirements.

### 3.3. Software Interfaces

The software interface requirements for this project may include compatible operating systems, database management systems, web servers, browser support, virtualization and containerization software, security software, development tools, backup and recovery solutions, threat intelligence integration, compliance, and reporting tools, and incident response systems. Additionally, Shuffle and Opsgenie for SOAR and Elastic Stack for SIEM will be used.

### 3.4. Communications Interfaces

Communication interfaces for this project include network interfaces, email integration, APIs for third-party tool interactions, and integration with external threat intelligence sources. These interfaces facilitate data exchange, incident reporting, and integration with external systems and services.

## 4. System Features

- **Brute Force Attacks**

Use Case	Preventing Brute Force Attacks	
Summary	Implement measures to protect against unauthorized access.	
Actors	User, Security System, Elastic Stack, Windows 10 pro.	
Pre-conditions	User accounts are configured on Windows Server 2019. Brute force prevention measures are implemented. Security logging and monitoring systems (Elastic Stack) are integrated.	
	Actors Action	Systems Action
<i>Happy Path</i>	-	-
Flow of Events	1. User initiates a login attempt.	2. Security System monitors login attempts. 3. Windows 10 pro enforces an account lockout policy after detecting multiple failed attempts. 4. Security System logs the authentication events, including failed attempts, using Elastic Stack.
Post-conditions	<ul style="list-style-type: none"><li>• User accounts are protected against brute force attacks.</li><li>• Security logs are updated with authentication events, failed attempts, and alerts.</li></ul>	

Alternative Path	<input type="checkbox"/> The brute force attack is not detected by SIEM, requiring the user or IT Security Team to identify it manually.
Author	Ali Iqbal

## • Malware Detection and Remediation

Use Case	Malware Detection and Remediation	
Summary	Detect and remediate malware on the system.	
Actors	User, Security System, Elastic Stack, Windows 10 pro, Elastic Defend	
Pre-conditions	Systems are configured with updated antivirus and anti-malware software. Malware prevention measures are implemented. Security logging and monitoring systems (Elastic Stack, Elastic Defend) are integrated.	
	Actors Action	Systems Action
Happy Path	-	-
Flow of Events	<ol style="list-style-type: none"> <li>1. Reports suspicious activity or observes unusual system behavior</li> <li>4. Verifies the alert, confirming the presence of malware through analysis of the alert details and affected system.</li> <li>5. Initiates the remediation process by instructing the Endpoint Security System to isolate the malware.</li> <li>8. Verifies the system's integrity post-remediation to ensure no traces of malware remain and the system is functioning correctly.</li> </ol>	<ol style="list-style-type: none"> <li>2. Endpoint Security System detects potential malware based on behavioral analysis.</li> <li>3. An alert is generated and sent to the IT Security Team via the SIEM system, including details of the detected threat.</li> <li>6. The Endpoint Security System quarantines the malware to prevent further spread and damage.</li> <li>7. The Endpoint Security System attempts to clean the infected files or deletes them if cleaning is not possible.</li> <li>9. Generates and logs an incident report detailing the detection, verification, and remediation steps taken.</li> </ol>
Post-conditions	<ul style="list-style-type: none"> <li>• The system is free from malware.</li> <li>• An incident report detailing the detection and remediation process is documented.</li> </ul>	



Alternative Path	<input type="checkbox"/> The malware is not detected by the endpoint security system, requiring the user or IT Security Team to identify it manually.
Author	Umair Khan

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

The Performance requirements can vary based on the specific software and technologies used. Generally, it is recommended to have a multi-core processor, a minimum of 8 GB RAM, sufficient storage space, a stable network connection, and an operating system compatible with the chosen SOAR platform for optimum performance.

### 5.2 Safety Requirements

Safety requirements typically apply to systems where physical safety and health are major concerns. However, in the context of cybersecurity and a SOAR system, the focus is primarily on data and information security, hence there are no safety requirements.

### 5.3 Security Requirements

Security requirements include access control, data encryption, authentication, incident data privacy, audit logging, threat intelligence integration security, secure communication, vulnerability management, malware protection, backup and recovery, compliance, user training, incident response planning, and third-party security. These measures safeguard the system against cyber threats and ensure data integrity and confidentiality.

### 5.4 Software Quality Attributes

Software quality attributes include security, reliability, scalability, usability, performance, maintainability, interoperability, compliance, data integrity, and availability. These attributes collectively ensure the system's effectiveness, reliability, and alignment with security needs and user expectations.

### 5.5 Business Rules

- The system must be able to detect malware infections in real-time.
- The system must be easy to use and configure.

- The system must be able to integrate with existing security infrastructure.

## **6. Other Requirements**

No other Requirements.



# **Software Design Specification**

## **Security Process Fusion through Automation**

**Version 1.0 approved**

**Prepared by:  
Umair Khan 2012412  
Ali Iqbal 2012286**

**SZABIST**

**June 2024**

**Software Design Specification:**

Facing a speedy surge in cyber threats and dealing with the intricate nature of handling security incidents calls for a solid and effective incident response framework. Enter Security Orchestration, Automation, and Response (SOAR), a complete solution designed to simplify and boost incident response procedures. Let's dive into this introduction to grasp why implementing a SOAR project is crucial.

## **1. Introduction:**

### **1.1 Purpose of this document:**

This project aims to develop a Security Orchestration, Automation, and Response (SOAR) system according to the unique needs of organizations. The project aims to enhance cybersecurity measures, streamline incident response processes, accelerate response times, protect sensitive data, ensure business continuity, and improve cost-effectiveness for organizations with limited financial resources. By implementing efficient incident response procedures and automating routine tasks, the project seeks to minimize the impact of security incidents, providing a cost-effective solution that safeguards critical information assets and strengthens the overall cybersecurity posture.

### **1.2 Scope of the development project:**

Our project is focused on delivering lightning-fast malware detection and remediation capabilities on an enterprise level. With advanced technologies, we aim to quickly identify and respond to malware threats, minimizing their impact on the organization's systems and data. Additionally, we will employ rapid analysis techniques to identify and mitigate brute force attacks, ensuring that employees are protected from falling victim to these deceptive attacks. Furthermore, our platform will be equipped to detect and address malware detection. By offering these lightning-speed detection and remediation features, our project aims to significantly enhance the cybersecurity defenses of organizations, safeguarding their digital assets and maintaining a secure operational environment.

### **1.3 Definitions, acronyms, and abbreviations:**

No abbreviations for now.

### **1.4. References:**

- [https://www.server-world.info/en/note?os=CentOS\\_Stream\\_9&p=elasticstack8&f=1](https://www.server-world.info/en/note?os=CentOS_Stream_9&p=elasticstack8&f=1)
- [https://www.elastic.co/guide/en/elasticsearch/reference/current/\\_installation.html](https://www.elastic.co/guide/en/elasticsearch/reference/current/_installation.html)

## **1.5. Overview of document:**

The following sections will describe the project's design, outlining high-level components and their interactions. It will address potential issues in the user interface. The system architecture description provides an overview of components, their structure, relationships, and identifies user interface challenges. Section 3 offers an in depth component description, while section 4 provides an overall system interface description. Section 5 identifies relationships with other products, section 6 delves into design trade-offs, and section 7 details pseudo codes for components.

## **2. System architecture description:**

### **2.1 Section Overview:**

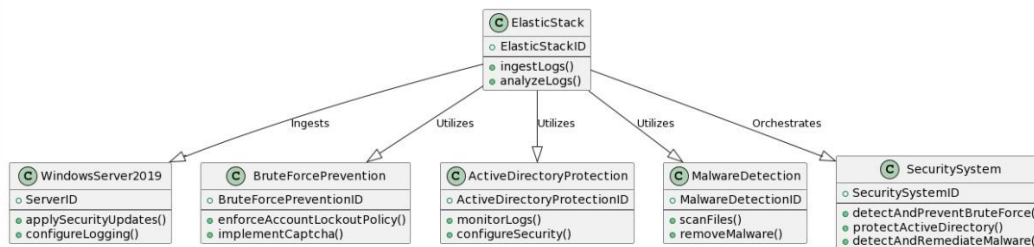
This section goes over the possible constraint a user might have to deal with in order to work with the application. The constraints may be result of external environments and the specific limitations for the application to perform optimally. This section also deals with the database design of the project the structure of the program and any alternatives considered for accomplishing the desired results and use cases.

### **2.2 General Constraints:**

- Technological Tools: VMware, CentOS, Windows server, and stable internet connection.
- Hardware and Software:
  - CPU: i5 10<sup>th</sup> gen processor or higher. □ RAM: 16 GB or more. □ Operating System: Latest (with a modern web browser).

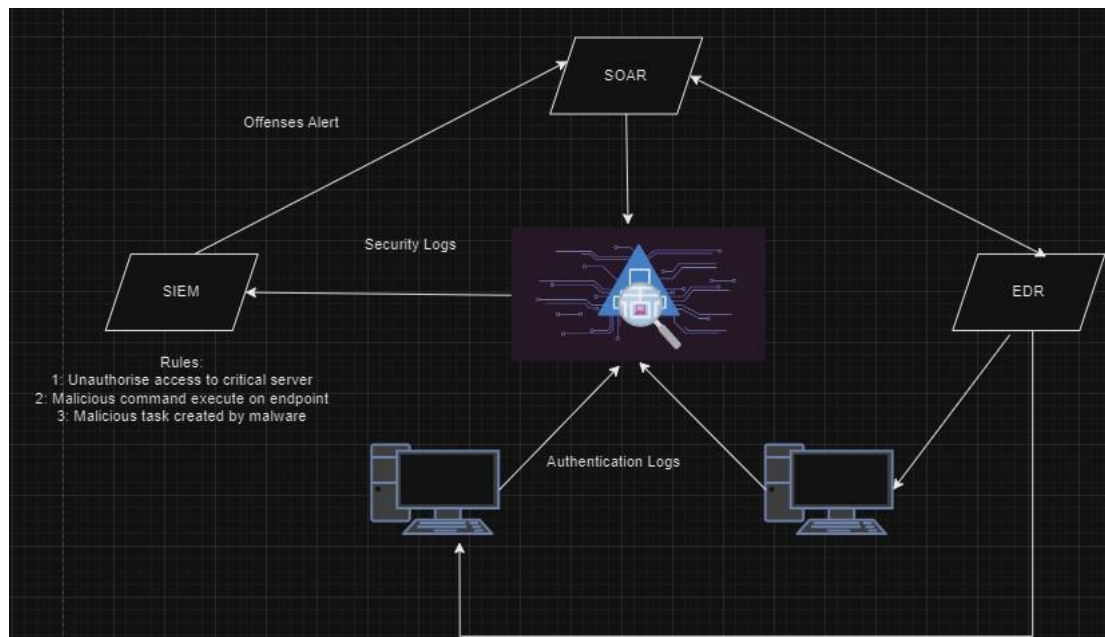
### **2.3. Data Design:**

The server system must be running 24/7 .Users can access the system from any environment that is compatible with legacy systems. Interoperability of various tools should be considered. Availability and reliability of external data sources can constrain integration. Internet connectivity is required.



## 2.4. Program Structure:

The security system is equipped with advanced features to ensure comprehensive threat management and rapid response capabilities. Upon the detection of security threats, the system autonomously initiates remediation actions, such as isolating affected systems and quarantining malicious files. Users benefit from customizable workflows and incident response playbooks, allowing them to tailor responses to specific security requirements. The automation engine within the system streamlines routine and manual security tasks, enhancing efficiency and reducing response times. Integrated with external threat intelligence sources, the system fortifies threat detection and response capabilities. User access control is meticulously managed, ensuring secure system usage through permissions, access control, and auditing. The system's robust reporting and analytics capabilities empower users to gain insights into security incidents and system performance. Timely notifications and alerts keep security teams informed about potential threats or incidents, fostering a proactive approach. With a user-friendly interface, the system promotes efficient interaction, prioritizing ease of use and accessibility for security teams, thereby bolstering overall cybersecurity measures.



### 3. Detailed description of components:

#### 3.1 Section Overview:

This section outlines the key components of our cybersecurity defensive project. The Intrusion Detection Module, stationed at the network perimeter, monitors traffic using signature-based and behavioral analysis, interfacing with threat intelligence feeds for enhanced detection. The Threat Intelligence Subprogram in the system's core collects and analyzes threat data from various sources.

Secure Data Files store sensitive information with robust encryption and access controls. Access

Control Procedures regulate user access based on roles, mitigating unauthorized entry.

Cryptographic Classes secure communication channels using encryption and public-key infrastructure. This succinct overview provides a snapshot of our project's architecture and functionality in bolsterin cyber defense.

#### 3.2 SDS component table

Identification	Command Line
Type	Bash Script
Purpose	Project purpose is to prevent/stop attacks and malware
Function	Manage logs and prevent active directory and brute force attacks and also detect malwares and remediate it.
Subordinates	No subordinates
Dependencies	No dependencies
Interfaces	Elastic, Kibana, TheHive Port and My Host IP
Resources	Hardware and Software: CPU: i5 10th gen processor or higher. RAM: 16 GB or more. Operating System: Any (with a modern web browser).
Processing	<ul style="list-style-type: none"><li>• Customizable Workflows and Playbooks</li><li>• Automation Engine</li><li>• Threat Intelligence Integration</li><li>• User Access Control and Auditing</li><li>• Reporting and Analytics</li><li>• Notification and Alerting</li></ul>

Data	
------	--

## **4. User Interface Design:**

### **4.1 Section Overview:**

This section provides the insight to the user interface of project. The detailed description to the user interface components and why they were chosen for this product. The idea behind this user interface and its components is briefly discussed in this section

### **4.2 Interface Design Rules:**

The design rules used in our interface are as follows:

- The interface is flexible
- The interface is user friendly.
- Design informative error messages that guide users on how to resolve issues.
- Interface remains usable and responsive as data volumes grow.
- Allow users to customize their dashboards and views to suit their specific needs. □ It allows user to directly manipulate interface objects.

### **4.3. GUI Components:**

#### **4.3.1. Analytics:**

In this section we can create dashboards, maps, canvas etc.

#### **4.3.2. Observability:**

In this section we can view logs, alerts etc.

#### **4.3.3. Security:**

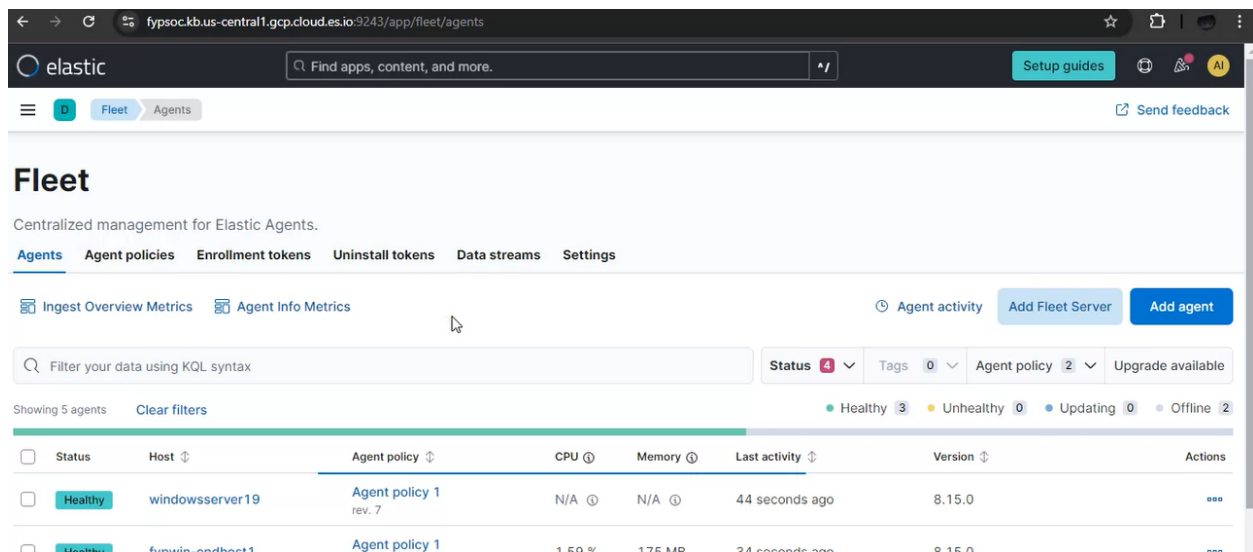
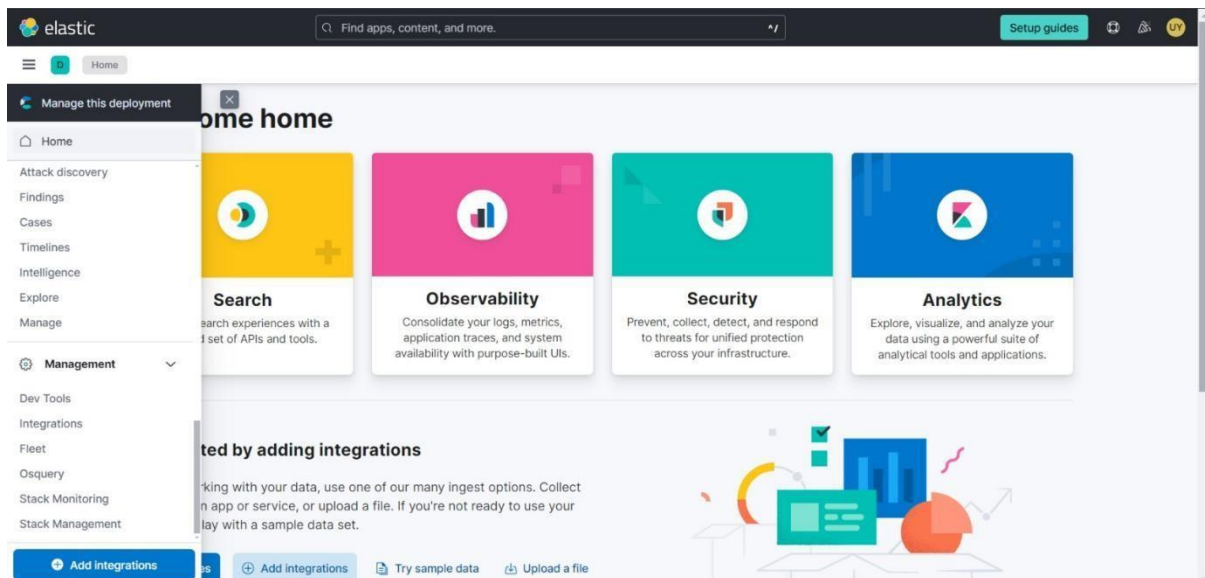
In this section we define rules, alerts and check timelines etc.

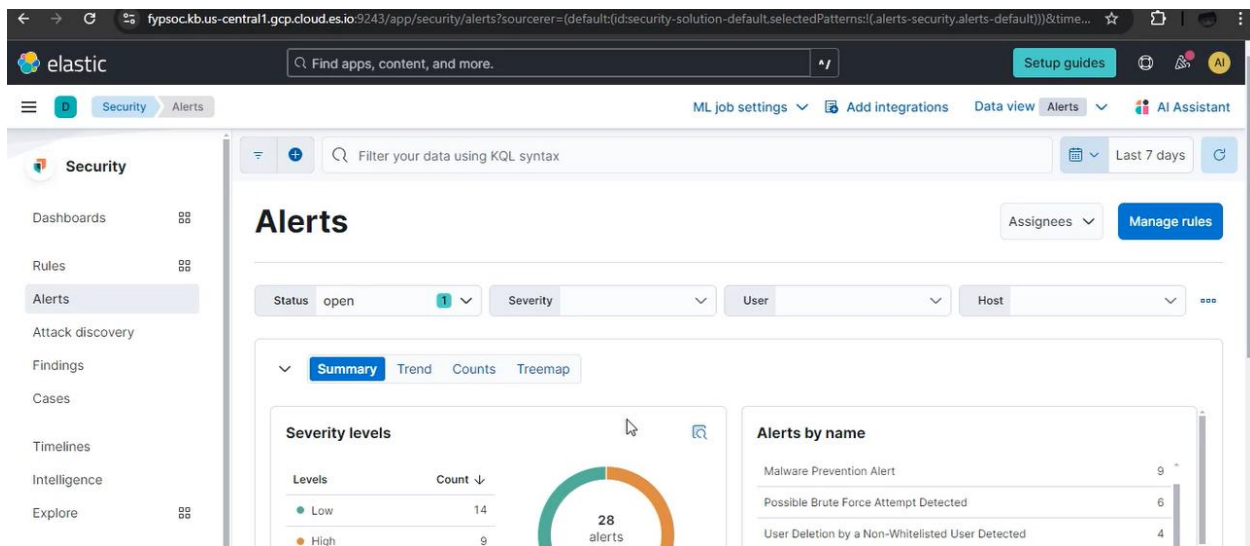


### 4.3.4. Management:

In this section we can install and manage different kind of integration tools.

## 4.4. Detailed Description:

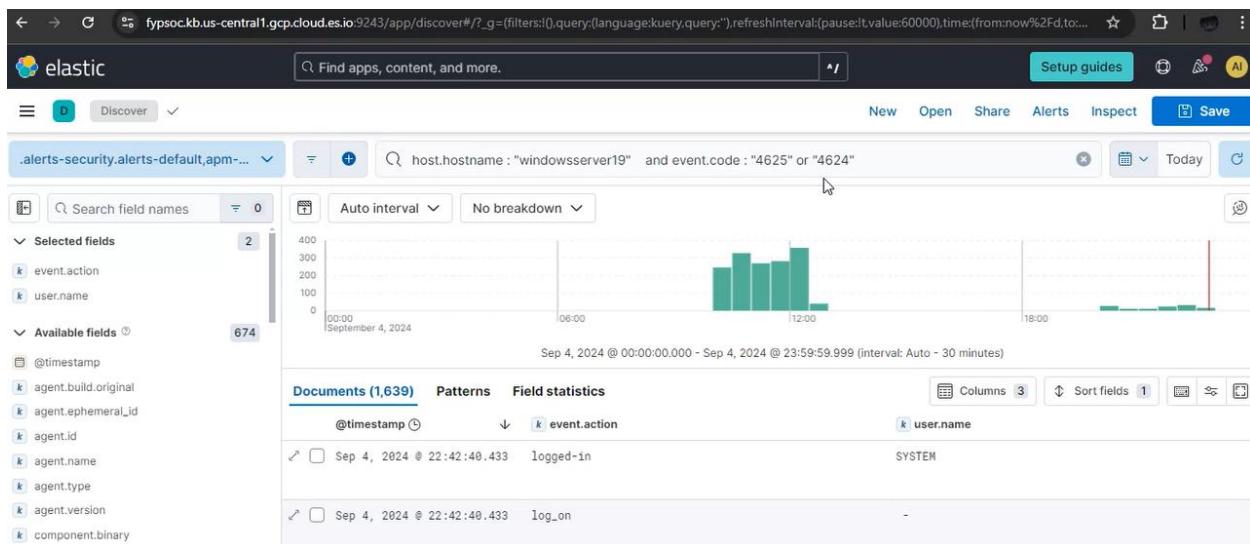


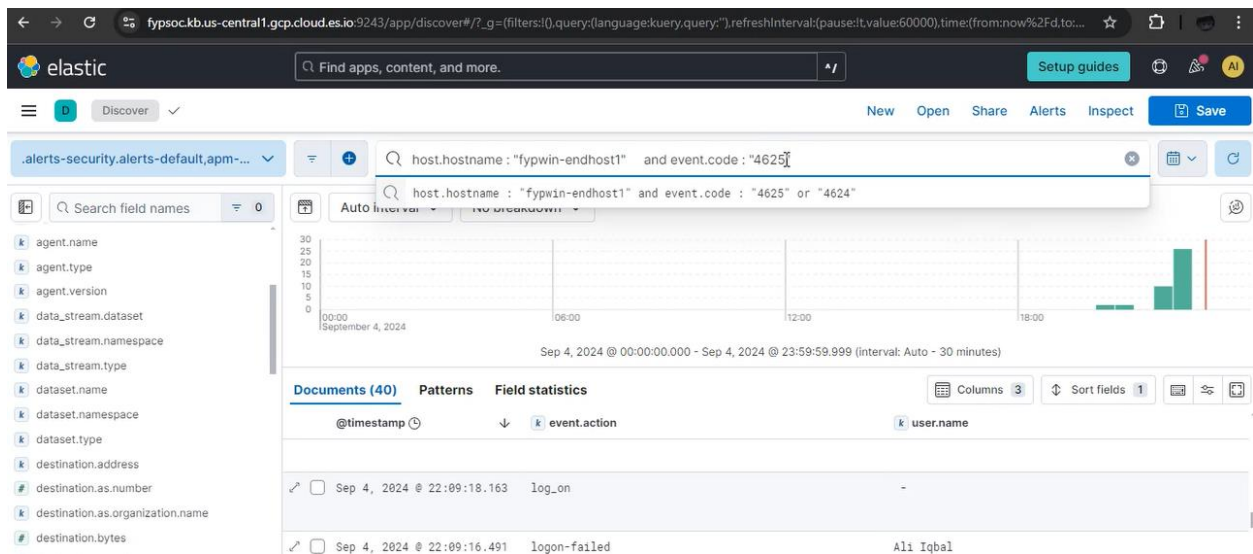
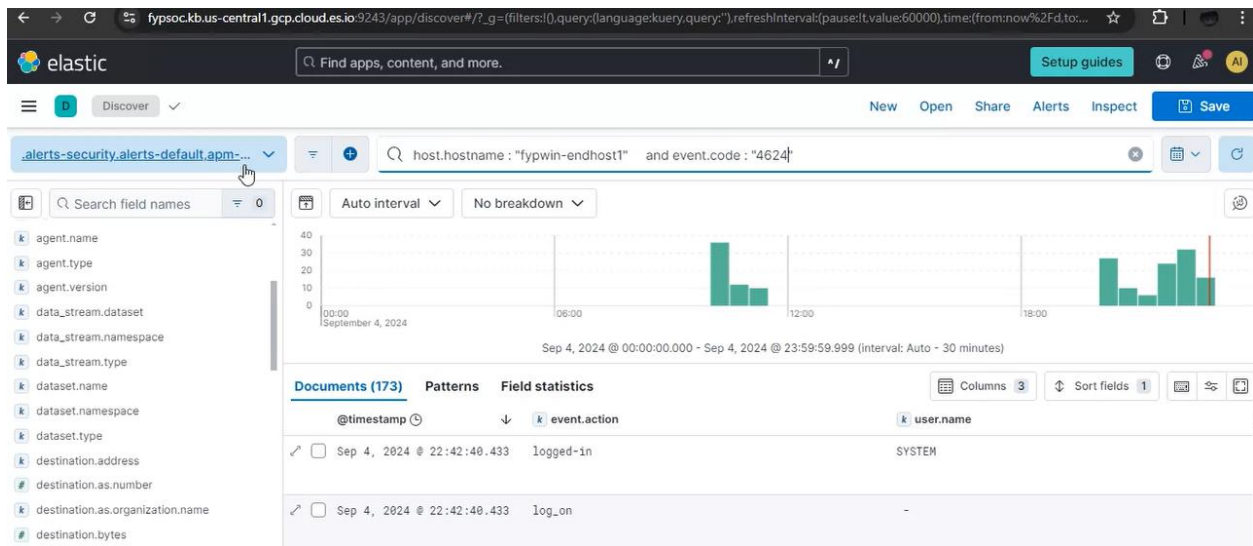


Columns: 16 Sort fields: 1 28 alerts Fields Updated 2 seconds ago Additional filters Grid view Group alerts by: None

Actions	@timestamp	Rule	Assignees	Severity	Risk Score	Reason
<input type="checkbox"/>	Sep 4, 2024 @ 22:59:02.452	User Deletion by a Non-Wh...		low	21	event by Ali Iqb...
<input type="checkbox"/>	Sep 4, 2024 @ 22:54:02.256	User Creation by a Non-Wh...		low	21	event by Ali Iqb...
<input type="checkbox"/>	Sep 4, 2024 @ 22:28:49.852	Malware Prevention Alert		high	73	malware, intrusi...
<input type="checkbox"/>	Sep 4, 2024 @ 22:23:46.538	Malware Prevention Alert		high	73	malware, intrusi...
<input type="checkbox"/>	Sep 4, 2024 @ 22:23:46.516	Possible Brute Force Attem...		low	21	event by Ali Iqb...
<input type="checkbox"/>	Sep 4, 2024 @ 22:13:40.611	Possible Brute Force Attem...		low	21	event by Ali Iqb...
<input type="checkbox"/>	Sep 3, 2024 @ 13:04:12.655	Malware Prevention Alert		high	73	malware, intrusi...
<input type="checkbox"/>	Sep 3, 2024 @ 12:59:09.582	Possible Brute Force Attem...		low	21	event by fypsoc...
<input type="checkbox"/>	Sep 3, 2024 @ 12:57:18.239	Praying Attempt...		medium	47	event by fypsoc...
<input type="checkbox"/>	Sep 3, 2024 @ 12:56:57.181	AD Attack - Brute Force de...		medium	47	event created r...

Rows per page: 10





The screenshot shows the Elasticsearch Security interface. The rule is titled "AD Attack - Brute Force detected over ...". The rule is created by Ali Iqbal on Sep 3, 2024 @ 10:50:43.369 and updated by Ali Iqbal on Sep 3, 2024 @ 10:50:57.446. The last response was "succeeded at Sep 4, 2024 @ 22:26:37.029". The rule is currently disabled.

**About**

Detected Failure/Brute Force Attempt over AD

**Severity** Medium

**Risk score** 47

**Definition**

**Index patterns** logs-\*

**Custom query** `host.hostname: "windowsserver19" and event.code: "4625"`

**Rule type** Threshold

elastic

Find apps, content, and more.

Setup guides

Security Rules Detection rules (SIE...) Possible Brute Force... Alerts

ML job settings Add integrations Data view Alerts AI Assistant

Security

Dashboards Rules Alerts Attack discovery Findings Cases Timelines Intelligence Explore

Filter your data using KQL syntax

Today

## Possible Brute Force Attempt Detected

Created by: 2503871732 on Aug 21, 2024 @ 01:53:28.828 Updated by: Ali Iqbal on Sep 3, 2024 @ 02:52:45.694

Last response: succeeded at Sep 4, 2024 @ 22:23:46.001 Notify when alerts generated

Enable Edit rule settings

### About

Detects Brute Force Attempts

Severity Low

Risk score 21

### Definition

Index patterns logs\*

Custom query event.provider : "Microsoft-Windows-Security-Auditing" and event.code : "4625"

fypsoc.kb.us-central1.gcp.cloud.es.io:9243/app/security/rules/id/432431c2-c46a-4c4b-9f5c-ba9073bbcea8/alerts?sourcerer=(default:(idsecurity-solution-default.sele...

fypsoc.kb.us-central1.gcp.cloud.es.io:9243/app/security/rules/id/432431c2-c46a-4c4b-9f5c-ba9073bbcea8/alerts?sourcerer=(default:(idsecurity-solution-default.sele...

fypsoc.kb.us-central1.gcp.cloud.es.io:9243/app/security/rules/id/432431c2-c46a-4c4b-9f5c-ba9073bbcea8/alerts?sourcerer=(default:(idsecurity-solution-default.sele...

fypsoc2024.app.opsgenie.com/alert/list

Opsgenie Alerts Who is on-call Teams Settings

16%

## Alerts

Create alert

{a} status: open Search Save

See all alerts Select All Time

### Saved searches

PREDEFINED

All

Open

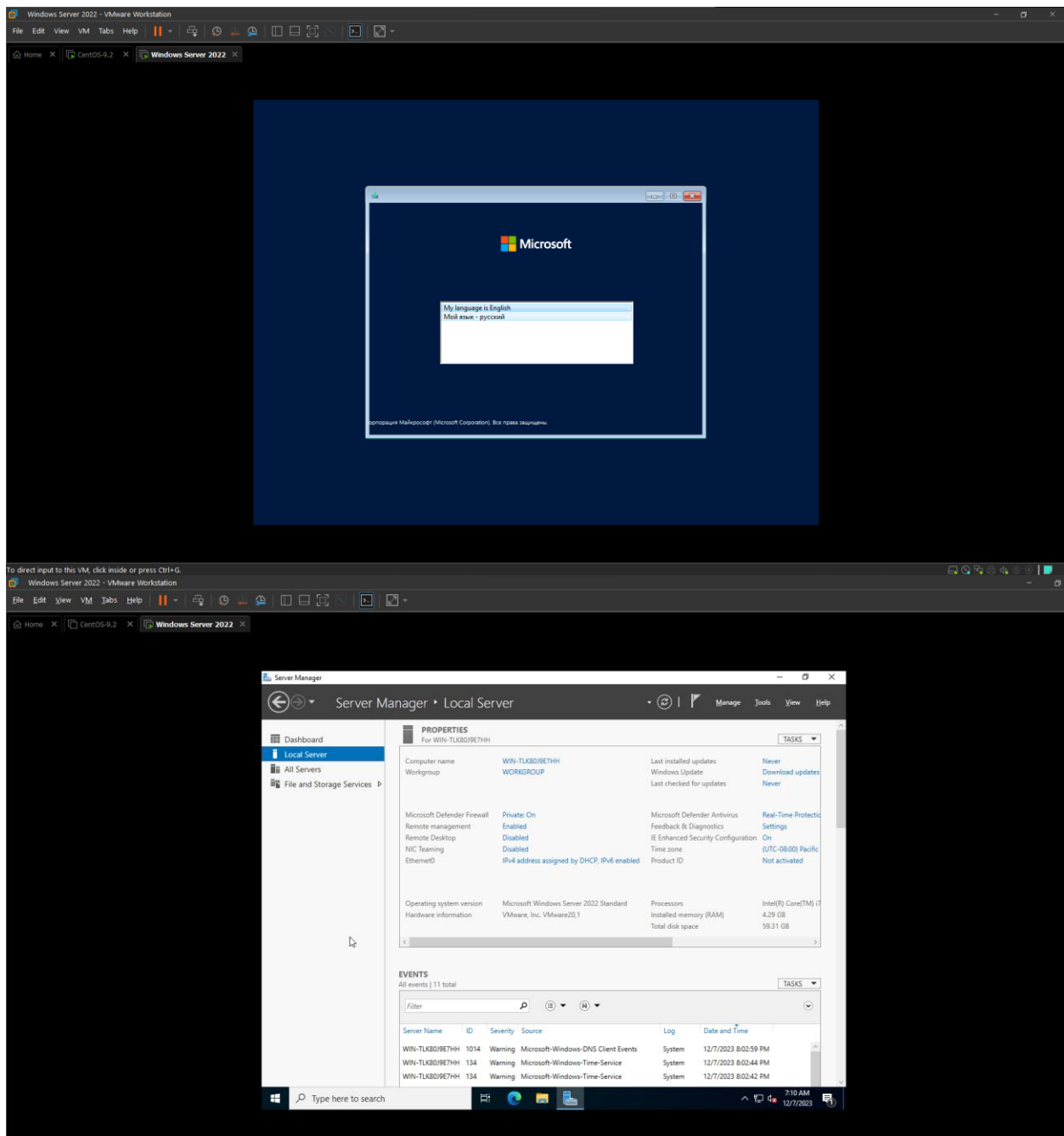
Closed

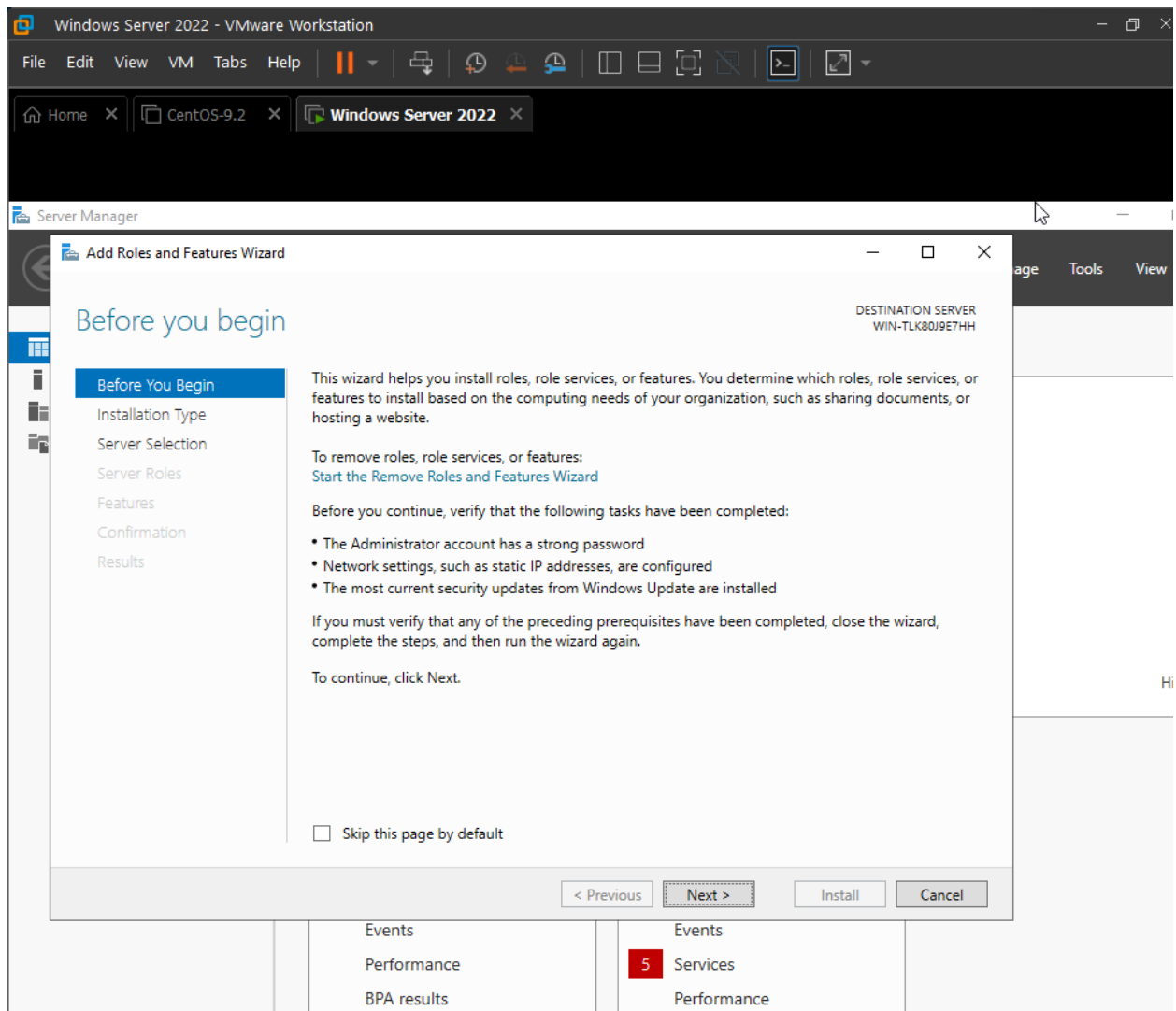
Un'Acked

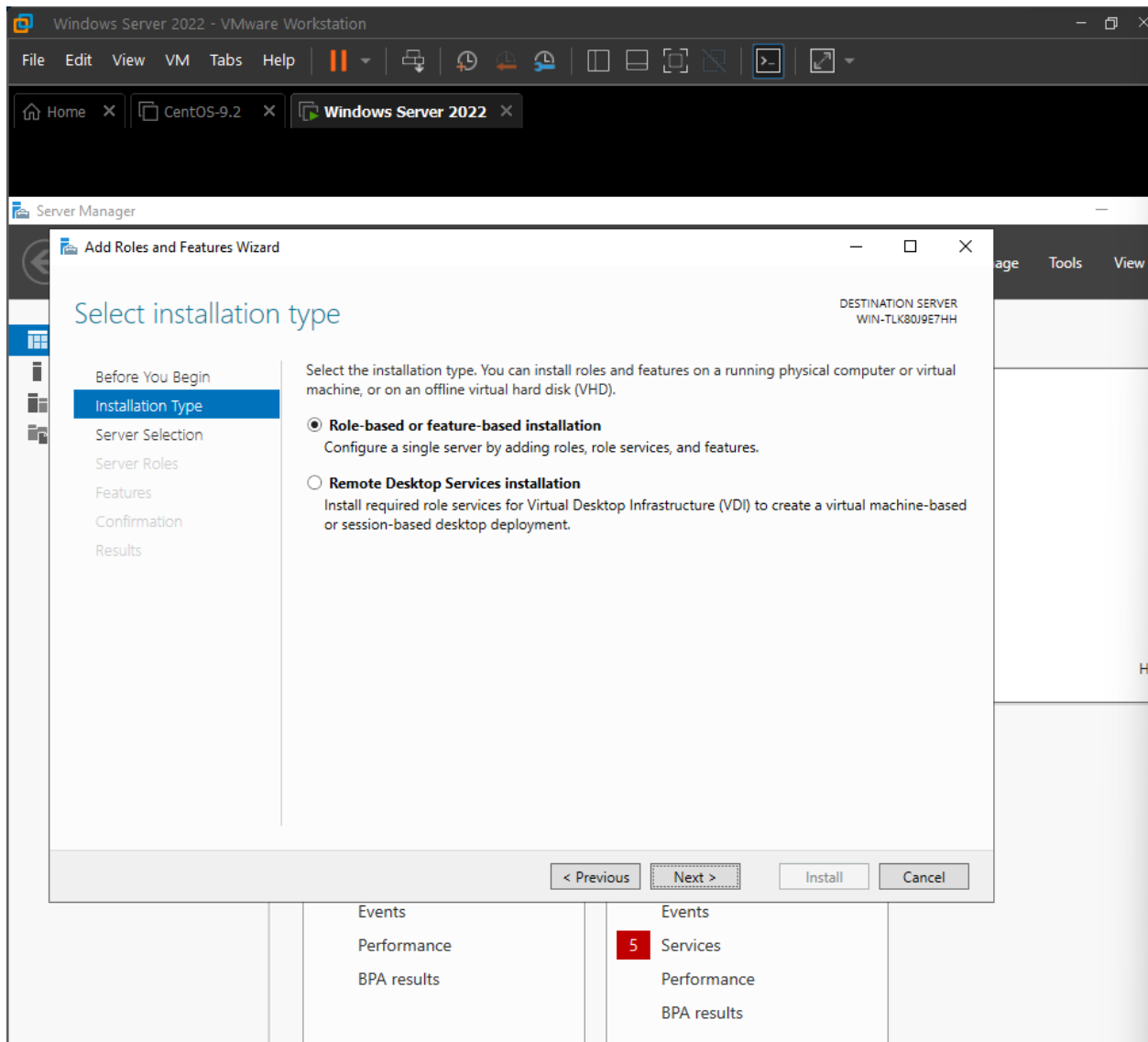
Not seen

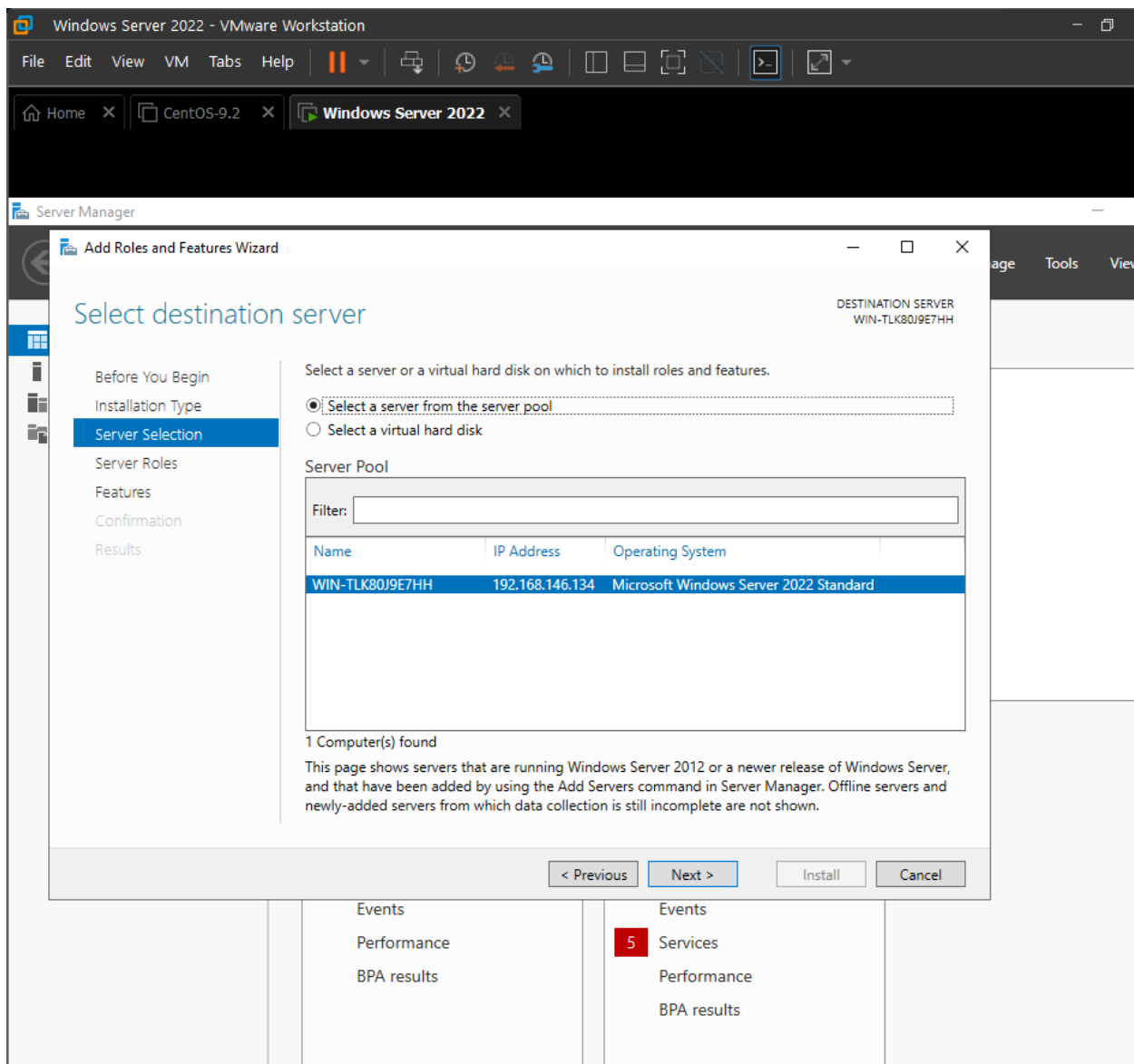
Assigned to me

<input type="checkbox"/>	#12	P5	User Creation by a Non-Whitelisted User Detected	1	FYP	OPEN	Ack Close	Sep 4, 2024 10:54 PM (GMT+05:00)
<input type="checkbox"/>	#11	P3	Password Spraying Attempt Detected	4	FYP	OPEN	Ack Close	Sep 3, 2024 11:45 AM (GMT+05:00)
<input type="checkbox"/>	#10	P3	AD Attack - Brute Force detected over Active Directory	2	FYP	OPEN	Ack Close	Sep 3, 2024 10:55 AM (GMT+05:00)
<input type="checkbox"/>	#8	P4	User Deletion by a Non-Whitelisted User Detected	4	FYP	OPEN	Ack Close	Sep 3, 2024 10:37 AM (GMT+05:00)
<input type="checkbox"/>	#7	P3	Malware Prevention Alert			OPEN		

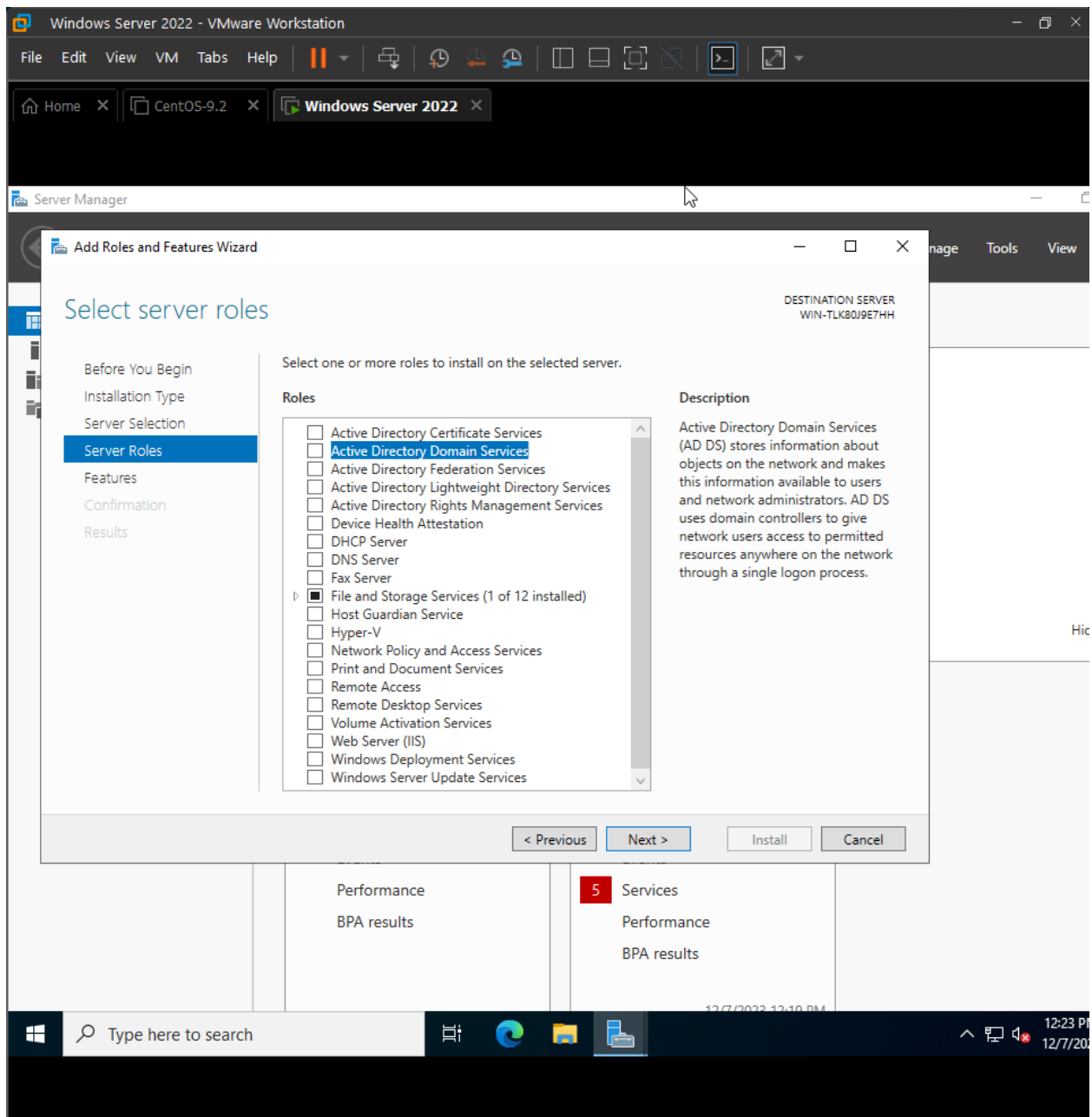


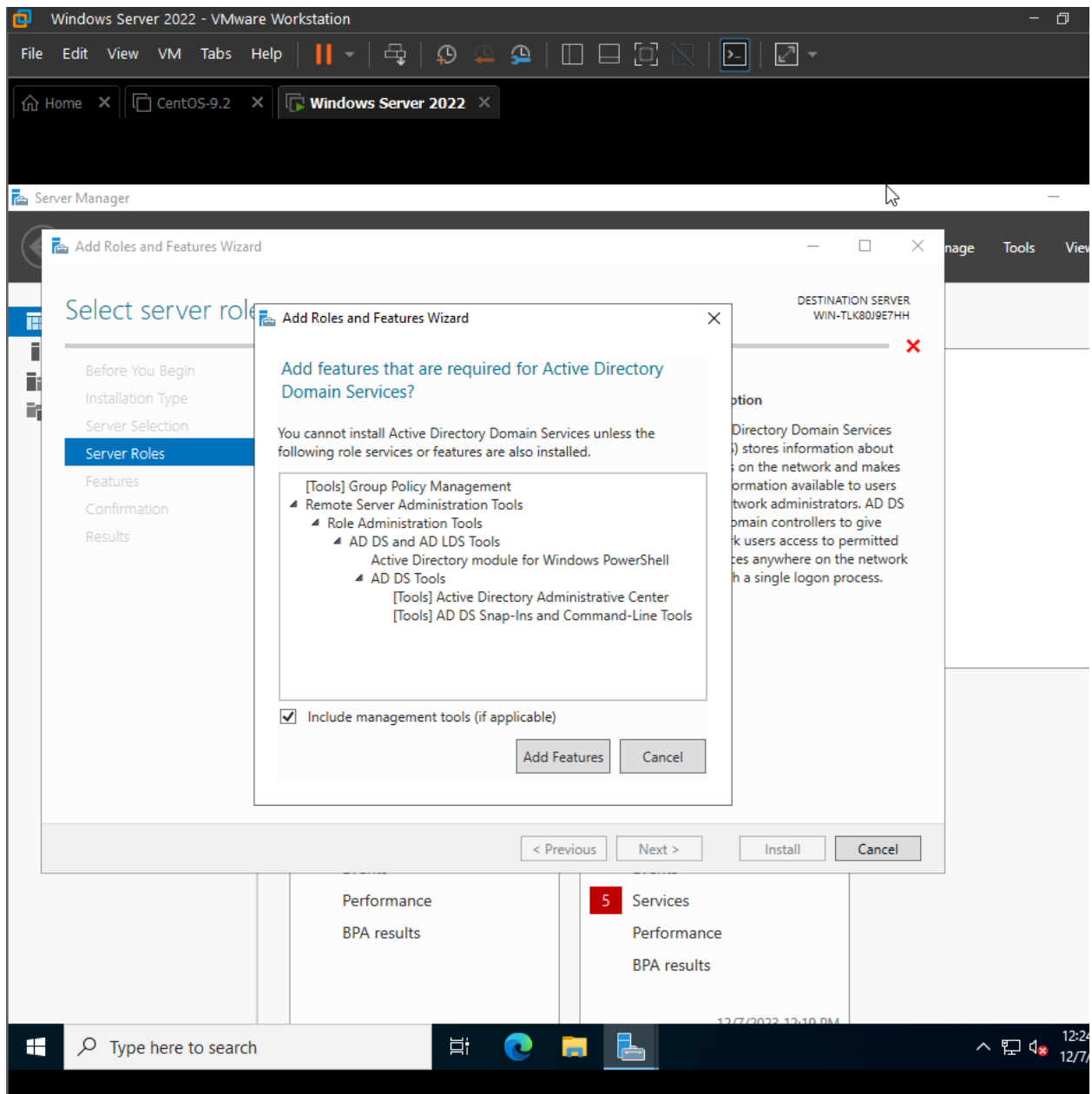


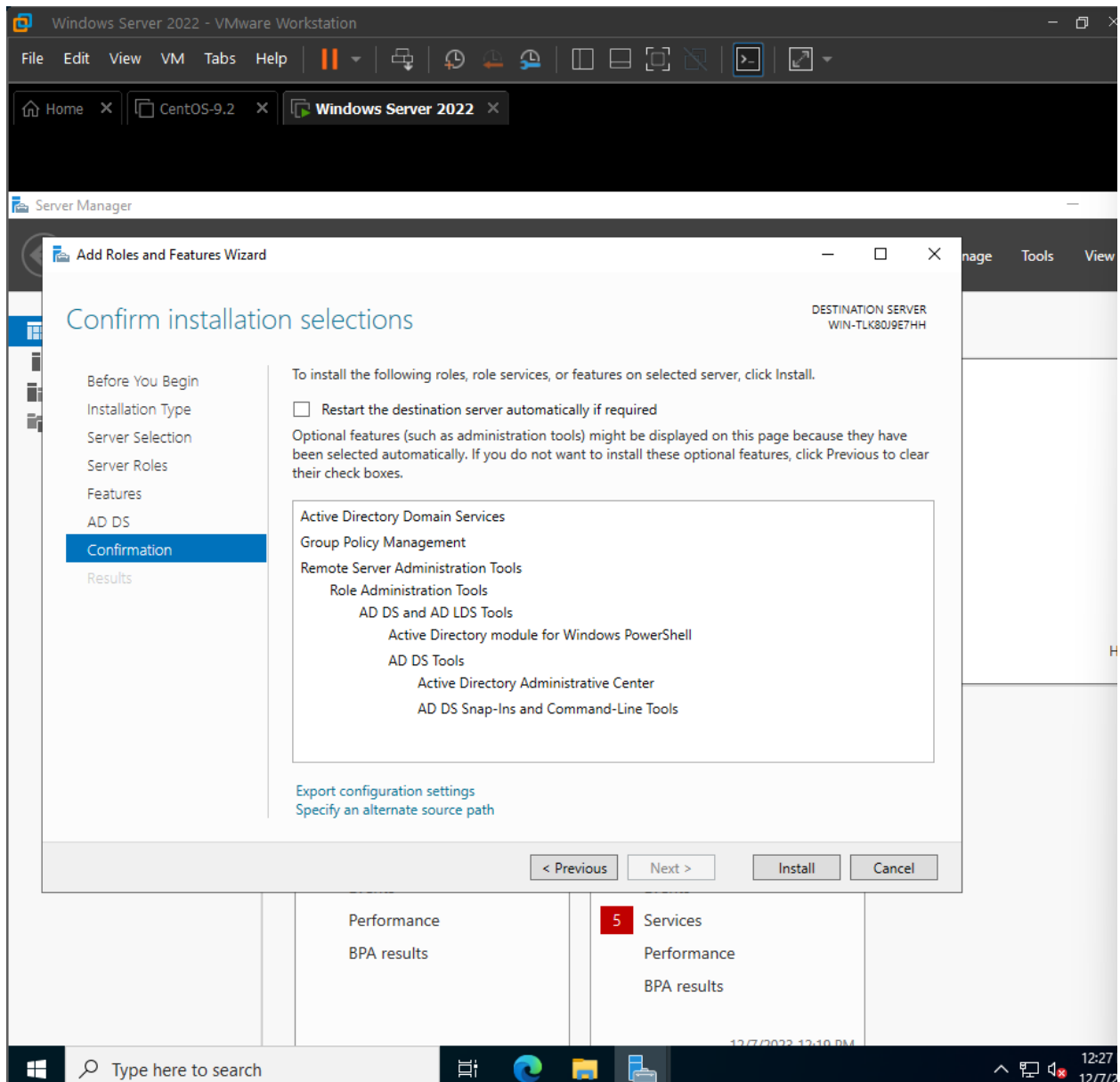


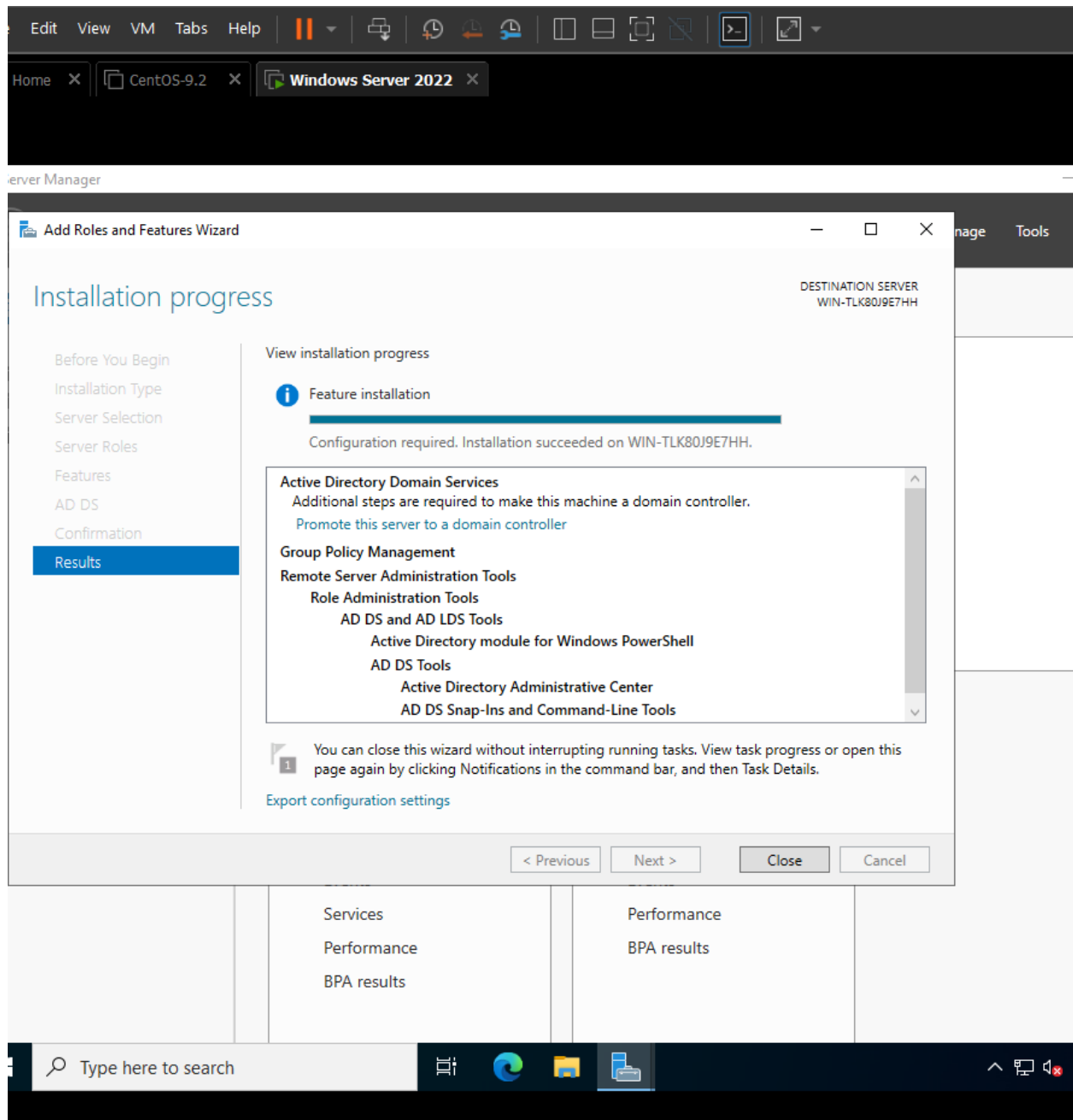


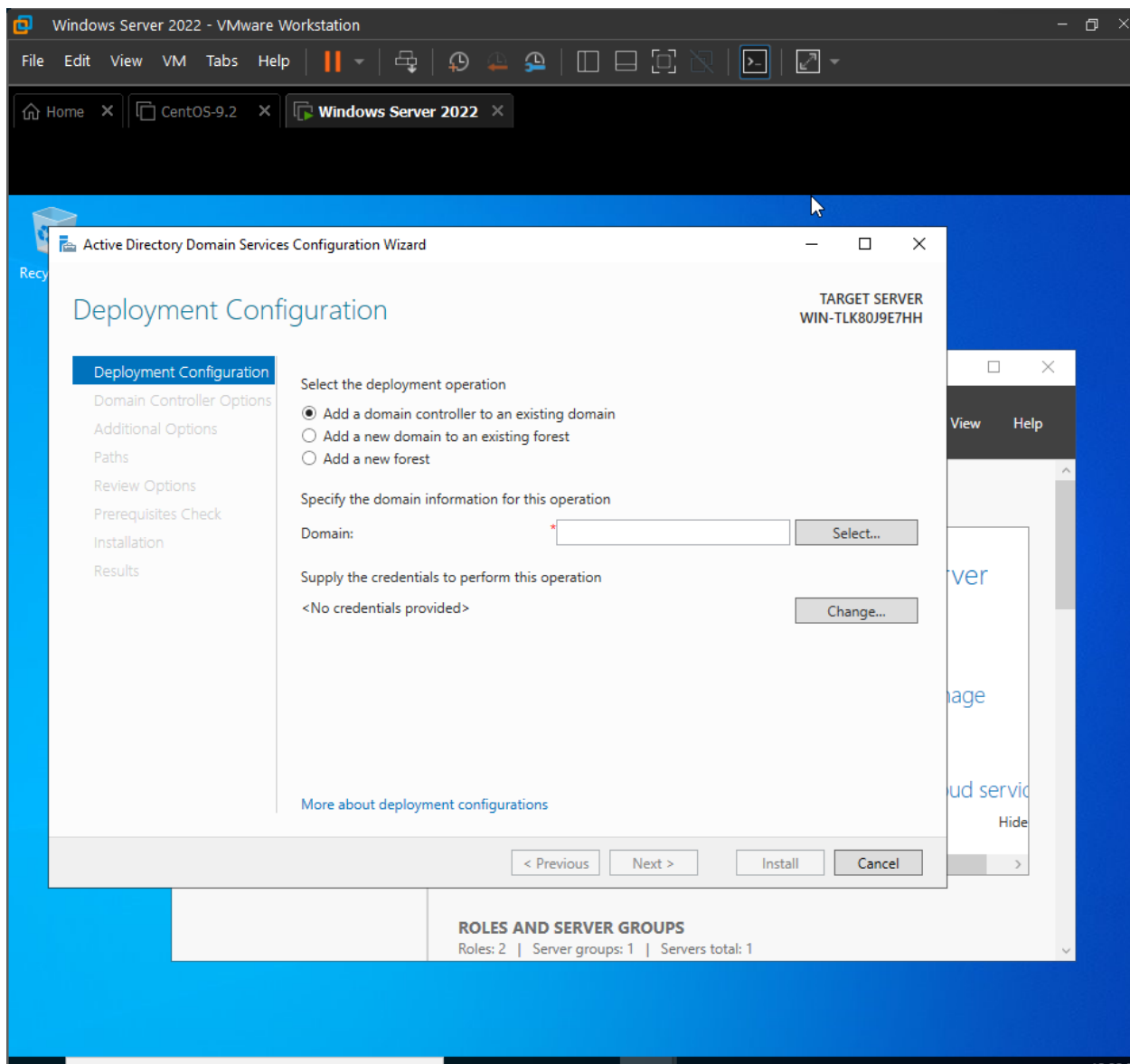


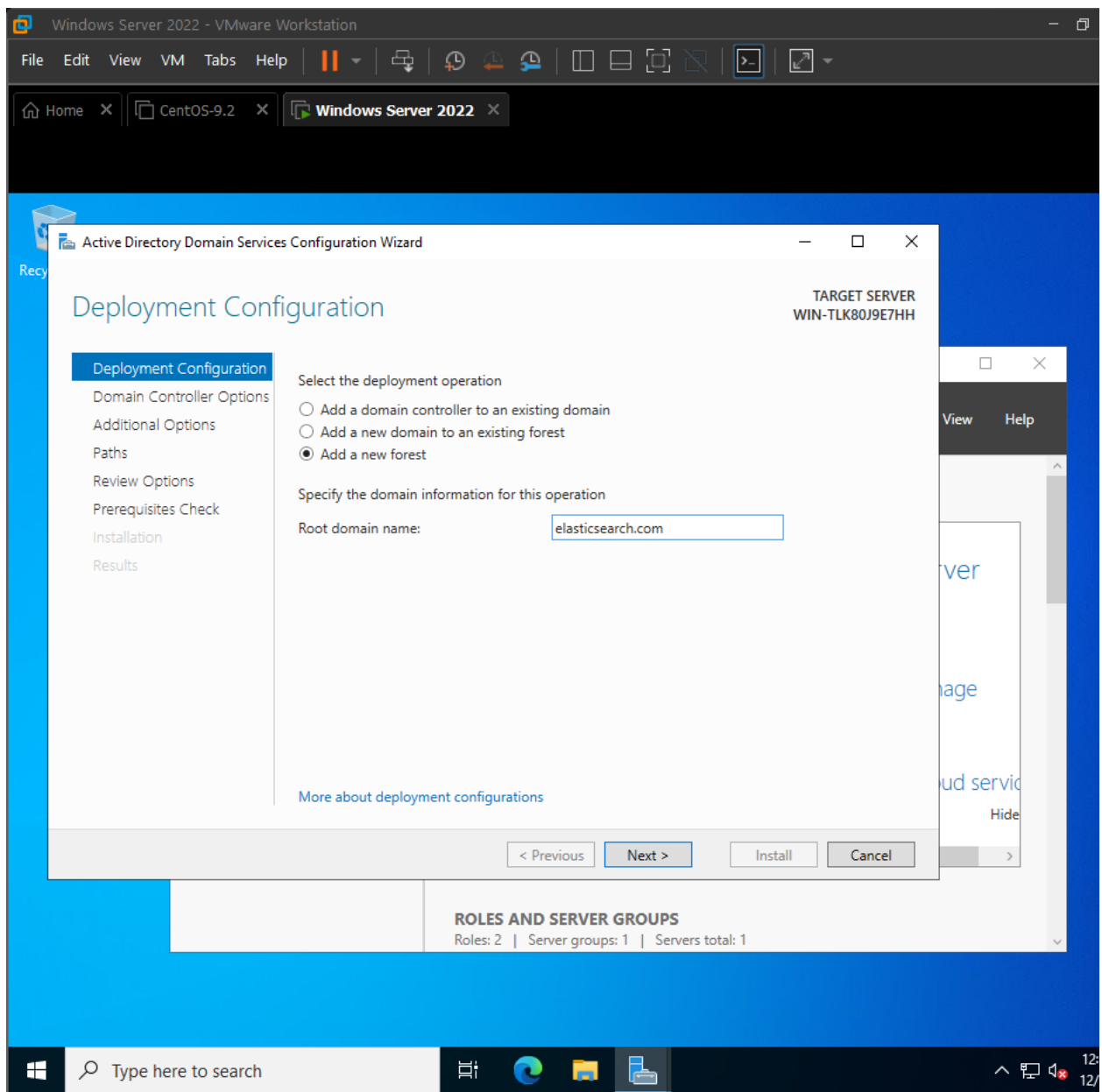


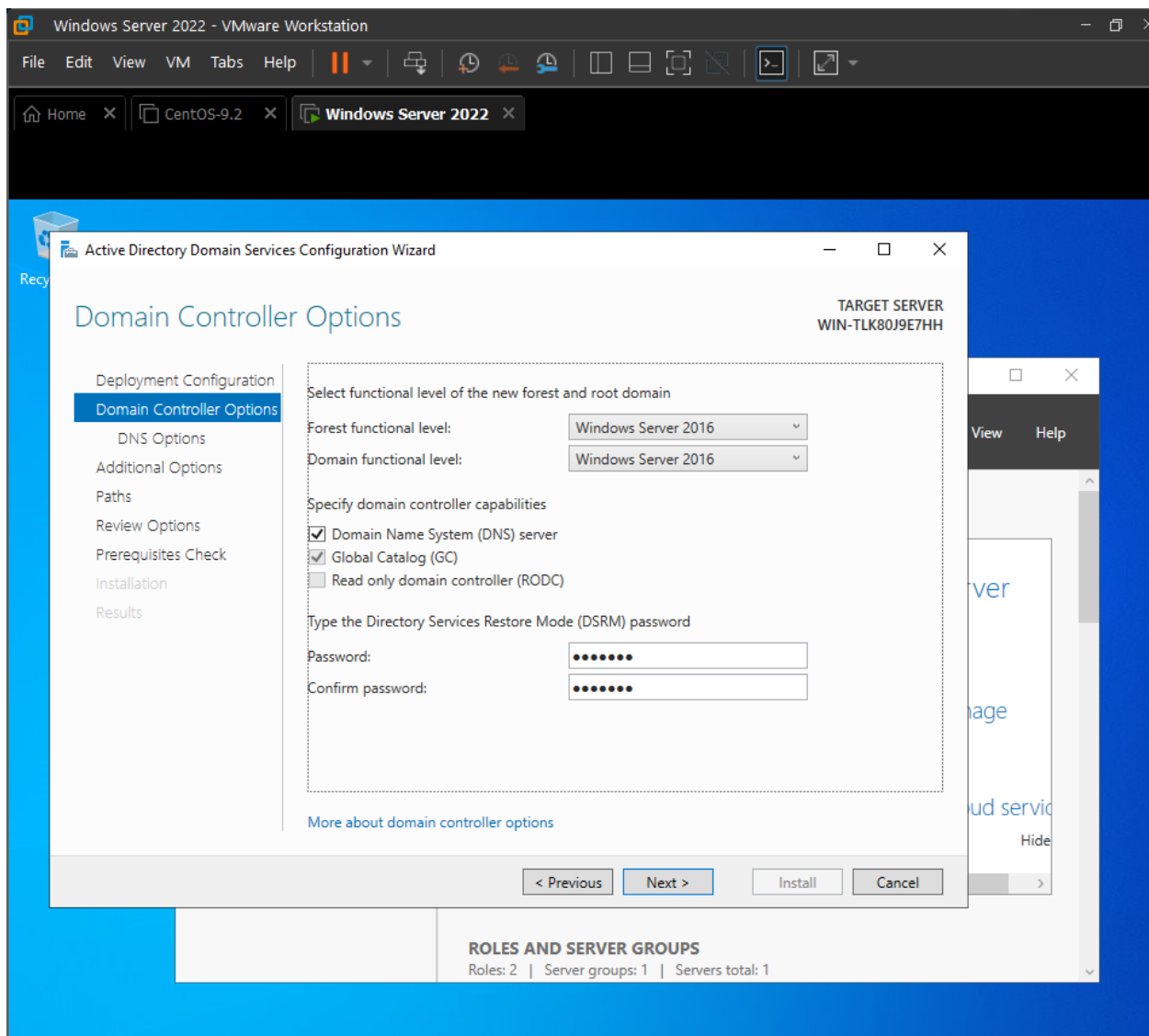


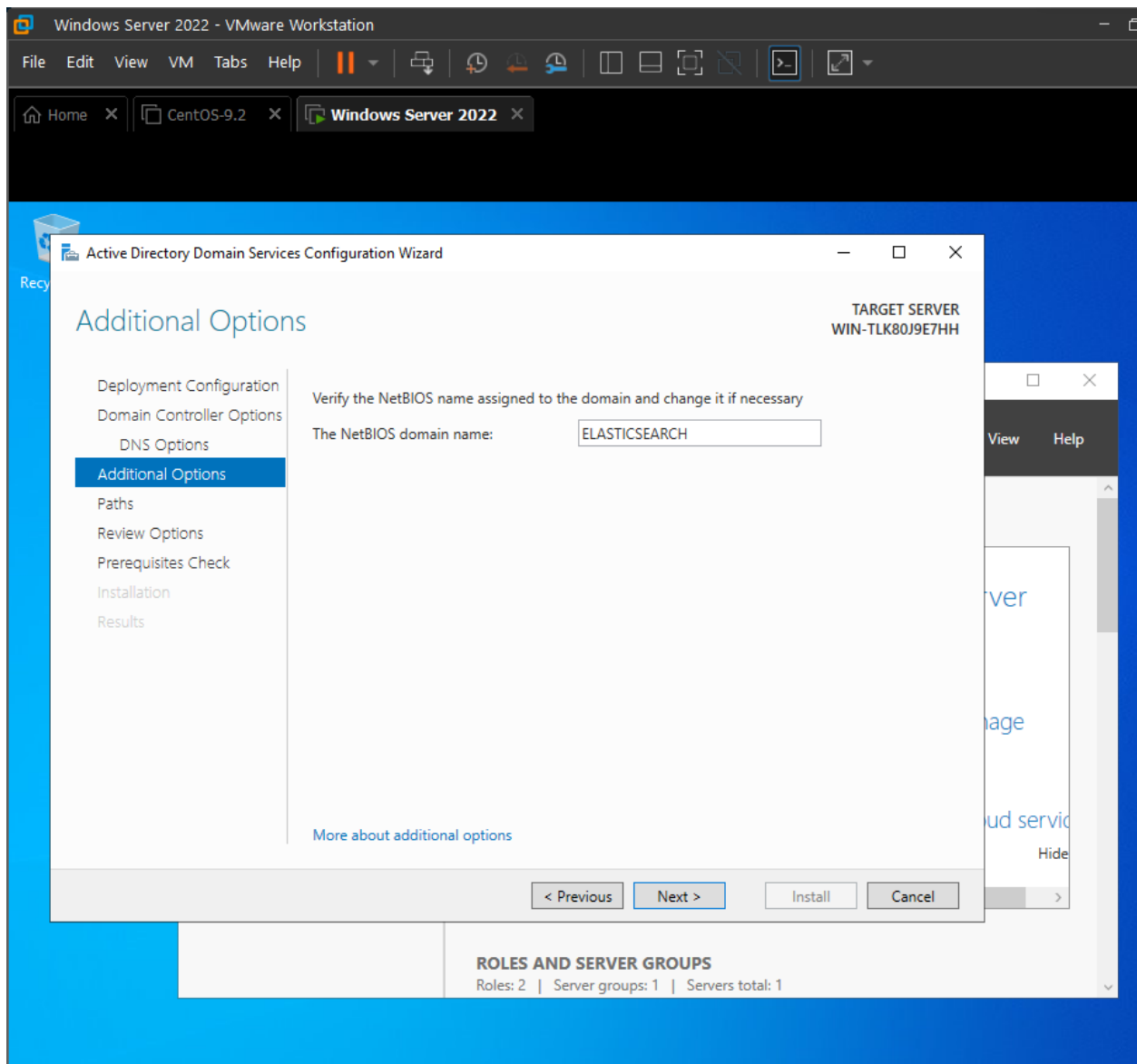




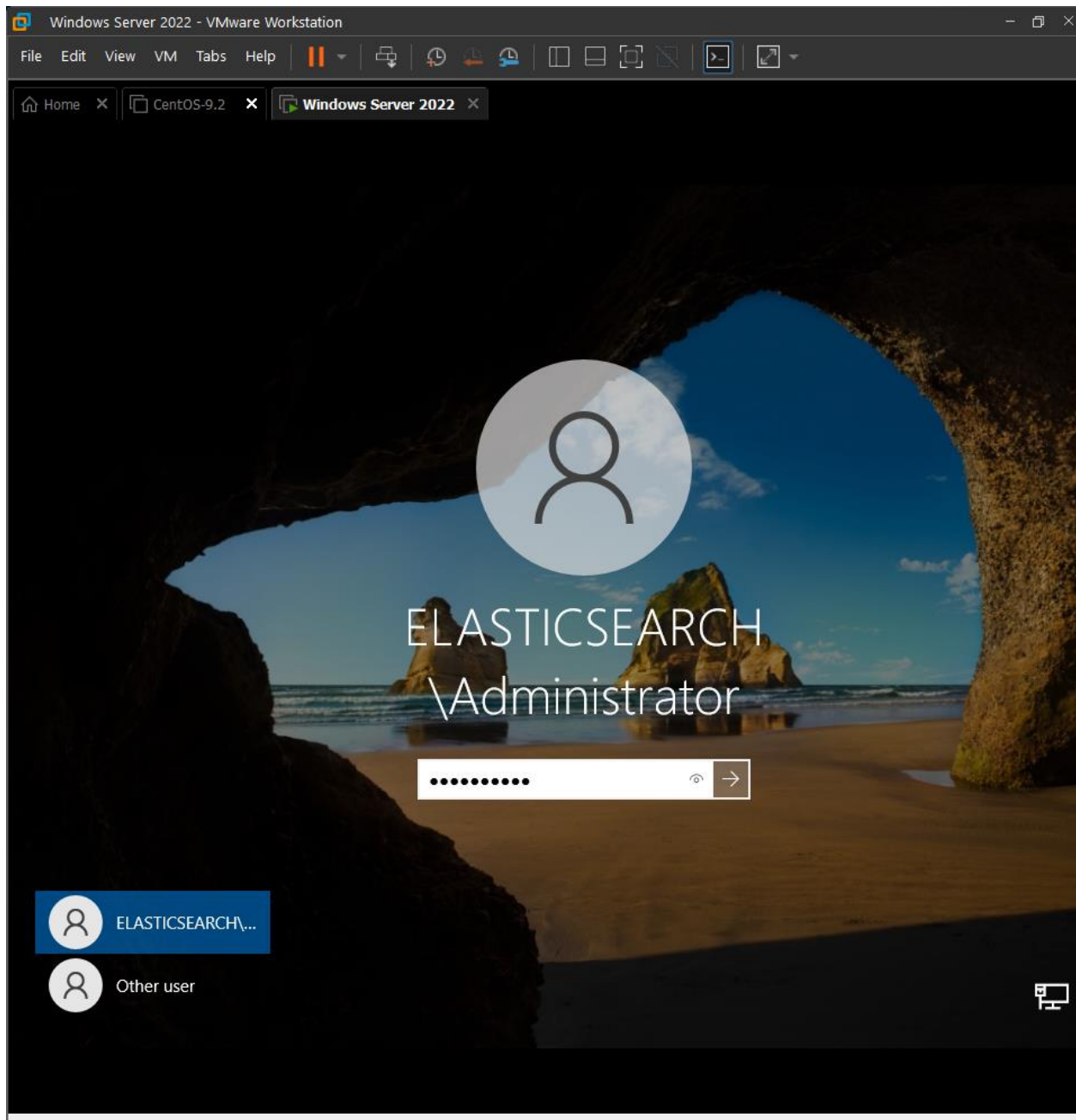


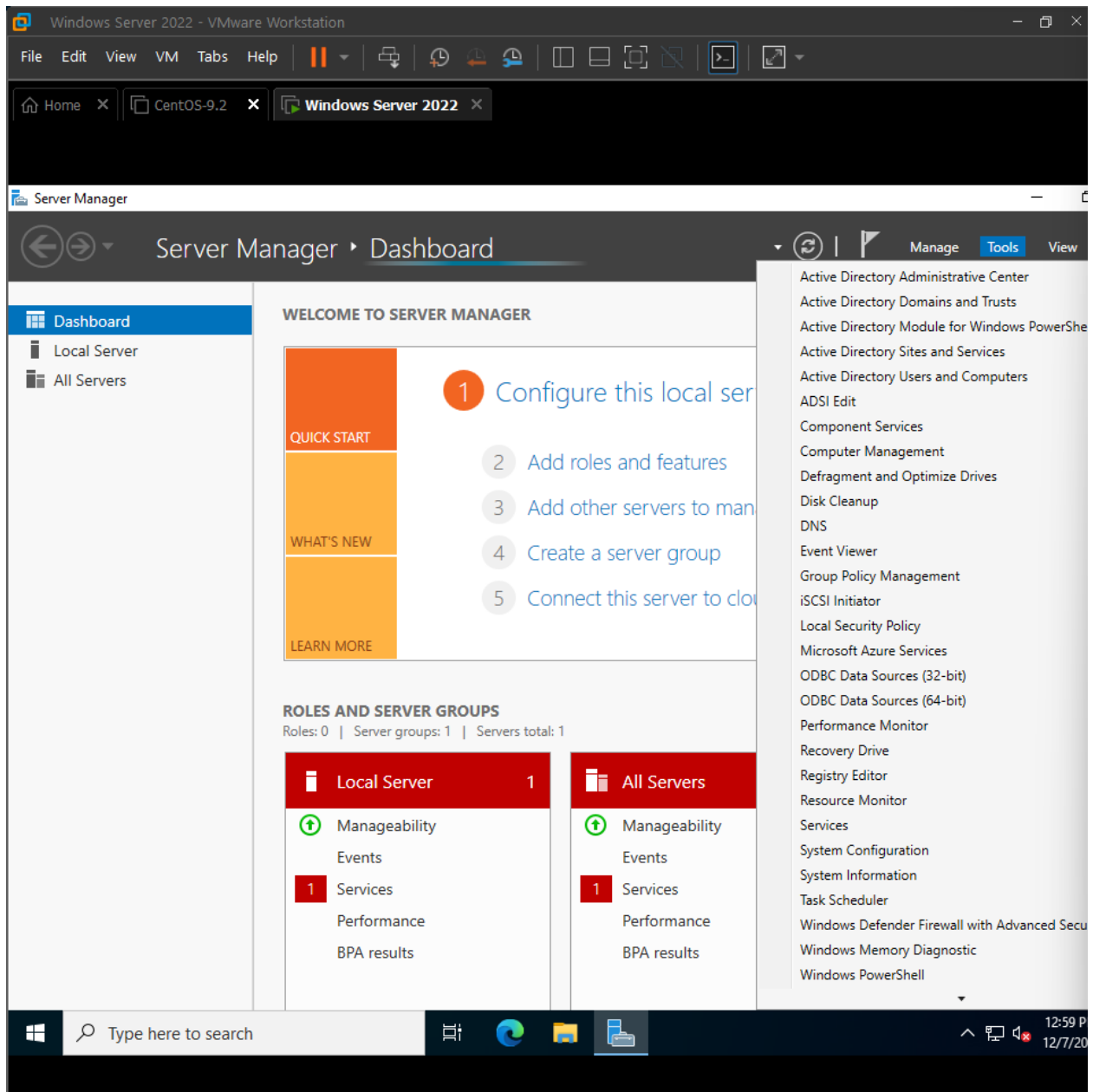


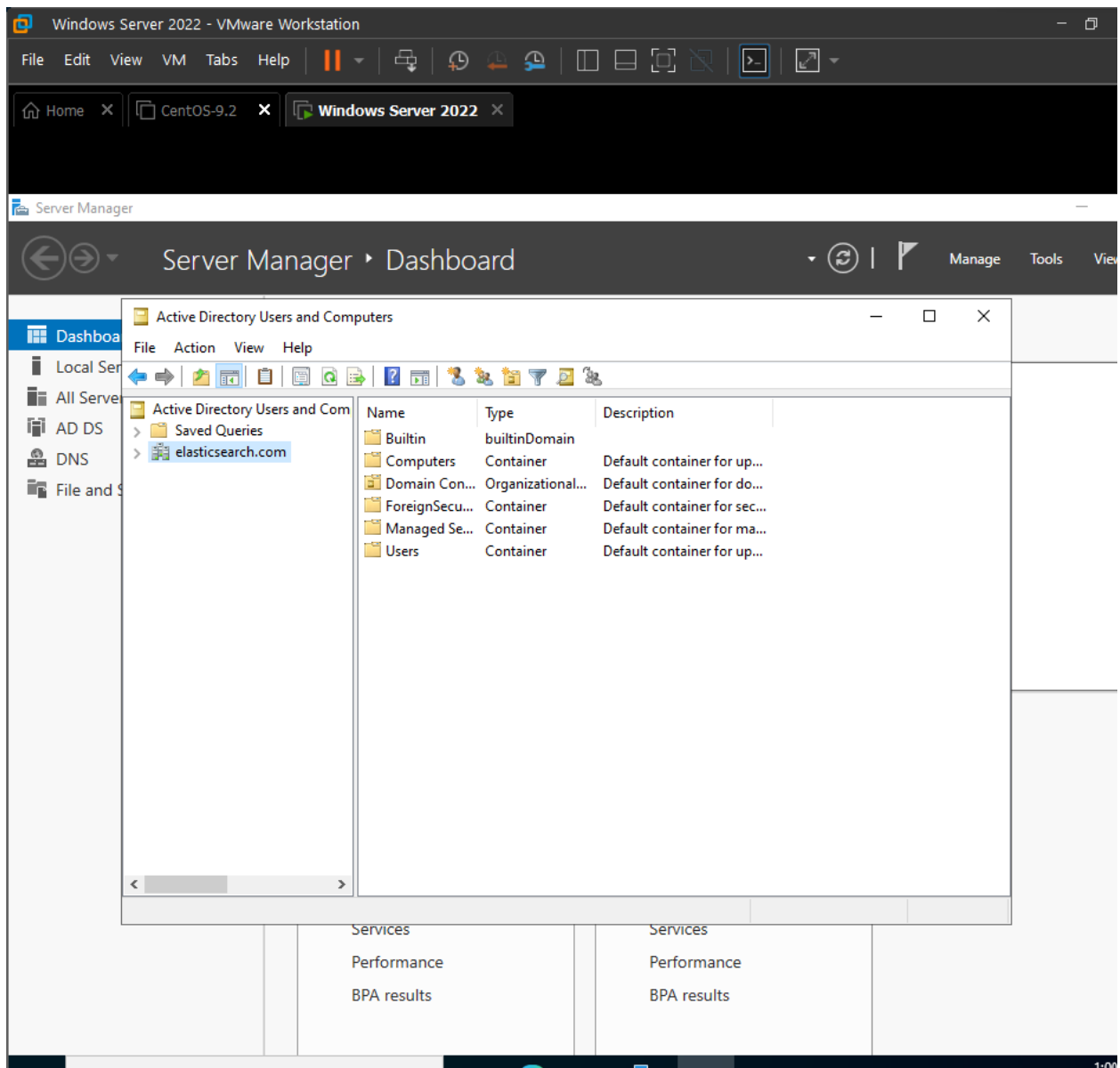




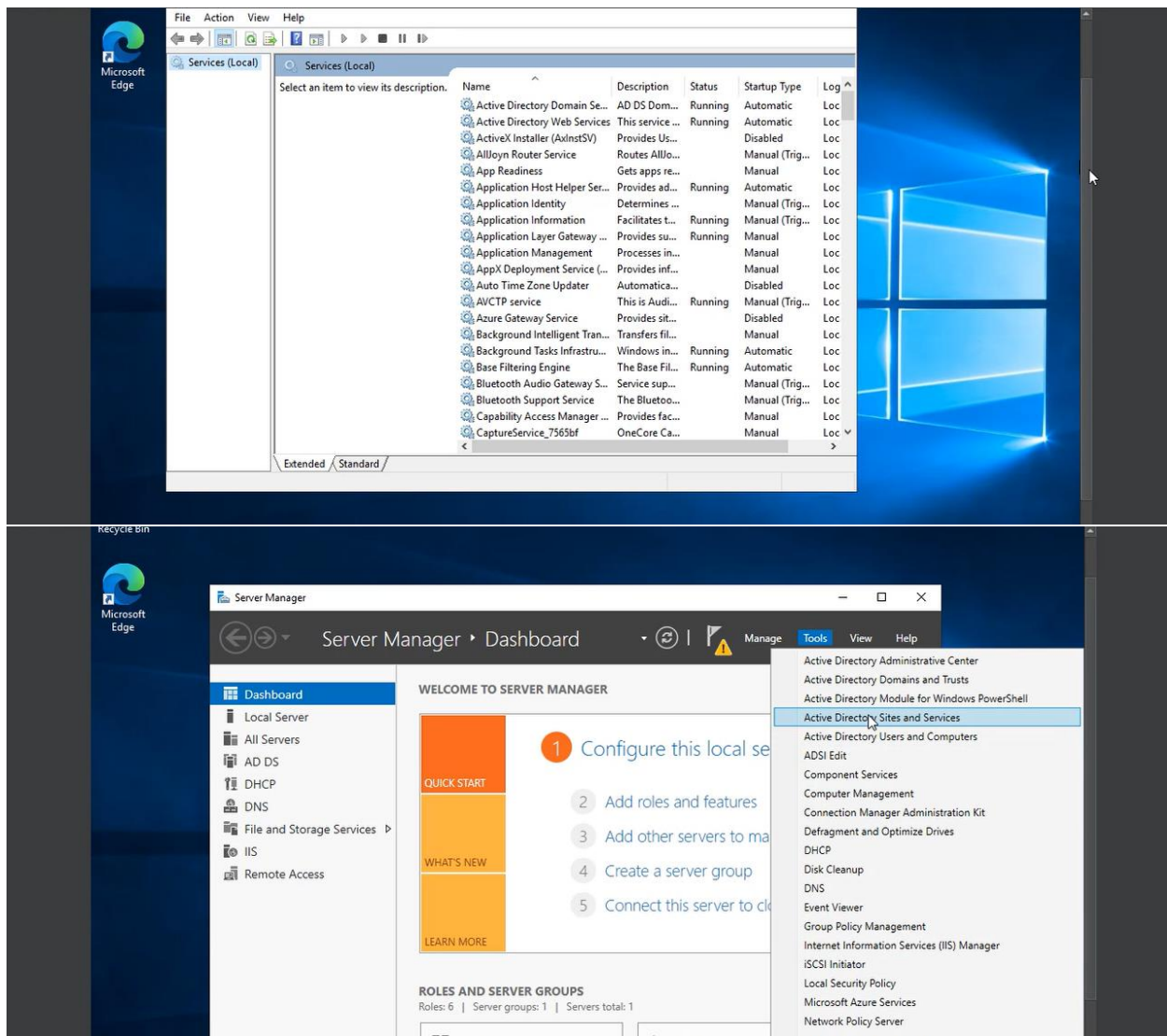












## 5. Reuse and relationships to other products:

The Performance requirements can vary based on the specific software and technologies used. Generally, it is recommended to have a multi-core processor, a minimum of 8 GB RAM, sufficient storage space, a stable network connection, and an operating system compatible with the chosen SOAR platform for optimum performance. It's crucial to have internet access and advanced technology which supports the system. Safety requirements typically apply to systems where physical safety and health are major concerns. However, in the context of cybersecurity and a SOAR system, the focus is primarily on data and information security, hence there are no safety requirements. Security requirements include access control, data encryption, authentication, incident data privacy, audit logging, threat intelligence integration security, secure communication, vulnerability management, malware protection, backup and

recovery, compliance, user training, incident response planning, and third-party security. These measures safeguard the system against cyber threats and ensure data integrity and confidentiality. Software quality attributes include security, reliability, scalability, usability, performance, maintainability, interoperability, compliance, data integrity, and availability. These attributes collectively ensure the system's effectiveness, reliability, and alignment with security needs and user expectations.

## 6. Design and Tradeoffs:

My project is CLI based so there is no design and tradeoffs.

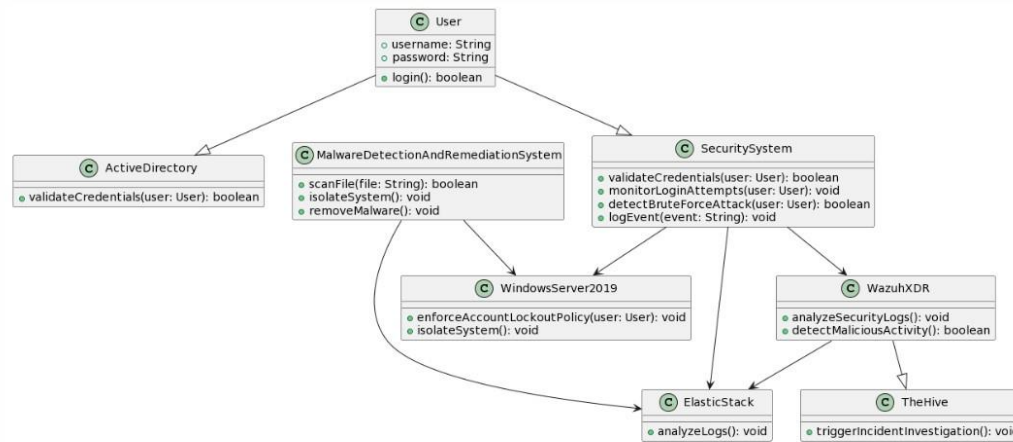
## 7. Pseudocode for components:

### Elastic stack agent installation:

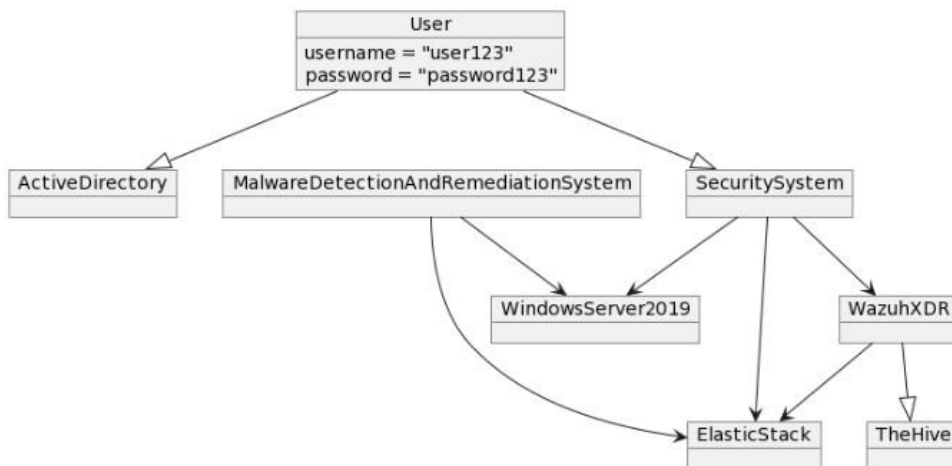
- Download Elastic Agent:  
Invoke-WebRequest -Uri [https://artifacts.elastic.co/downloads/beats/elastic-agent/elastic-agent-8.14.0-windows-x86\\_64.zip](https://artifacts.elastic.co/downloads/beats/elastic-agent/elastic-agent-8.14.0-windows-x86_64.zip) -OutFile elastic-agent-8.14.0-windows-x86\_64.zip
- Unarchive:  
Expand-Archive .\elastic-agent-8.14.0-windows-x86\_64.zip -DestinationPath .  
cd elastic-agent-8.14.0-windows-x86\_64
- Installation:  
.\elastic-agent.exe install --  
url=https://58d85fed0ad14960957fc75eb0a4225c.fleet.uscentral1.gcp.cloud.es.io:443 --  
enrollment-  
token=UGZFNC1ZOEJwVWpyXzJneldZTWk6bGI1LWhpaE9ScEdta1dYX25BbzBVdw=  
=  
=
- To Check Logs Enablement over machine run this command over CMD in admin mode:auditpol.exe /get /category:\*
- User Creation: Whwn you delete the user via command (net userusername password /add)
- User Deletion:Whwn you delete the user via command (net user username /delete)

## 8. Appendices:

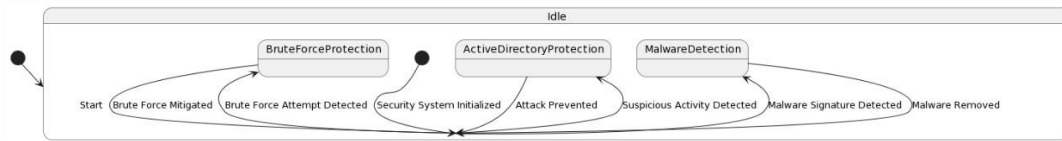
### 8.1. Class Diagram:



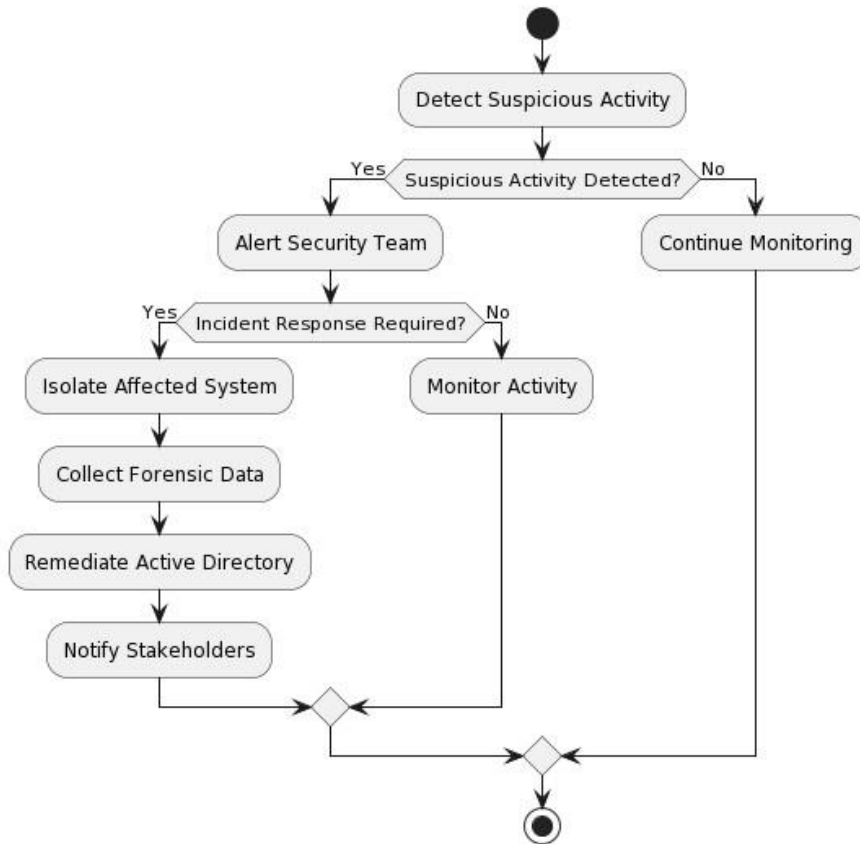
### 8.2. Object Diagram:



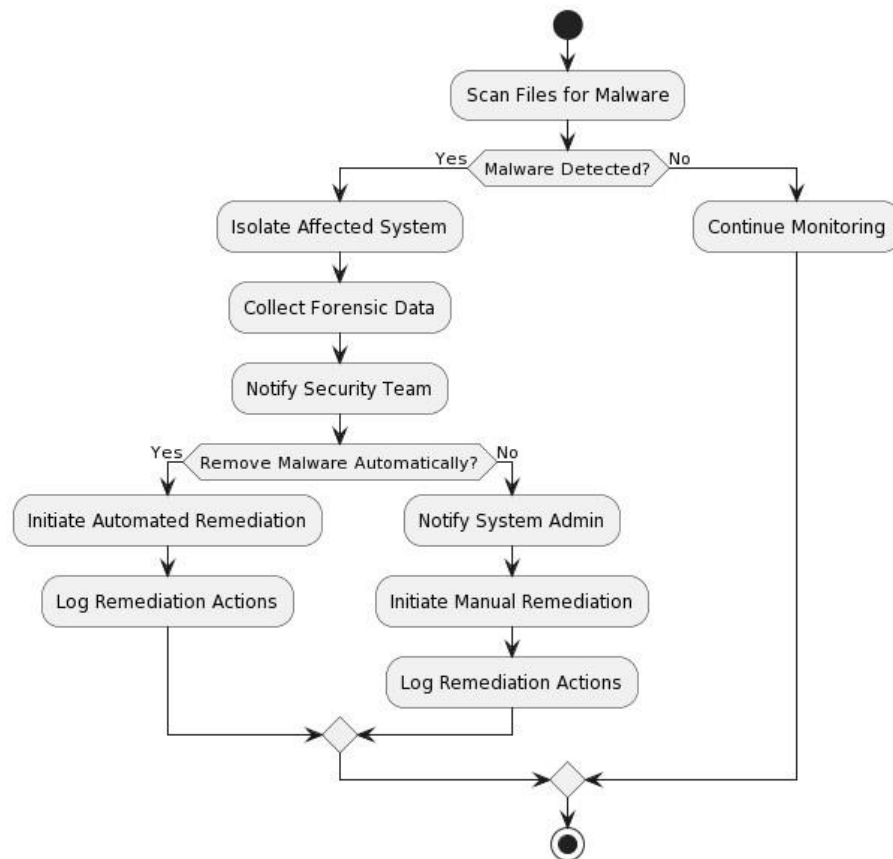
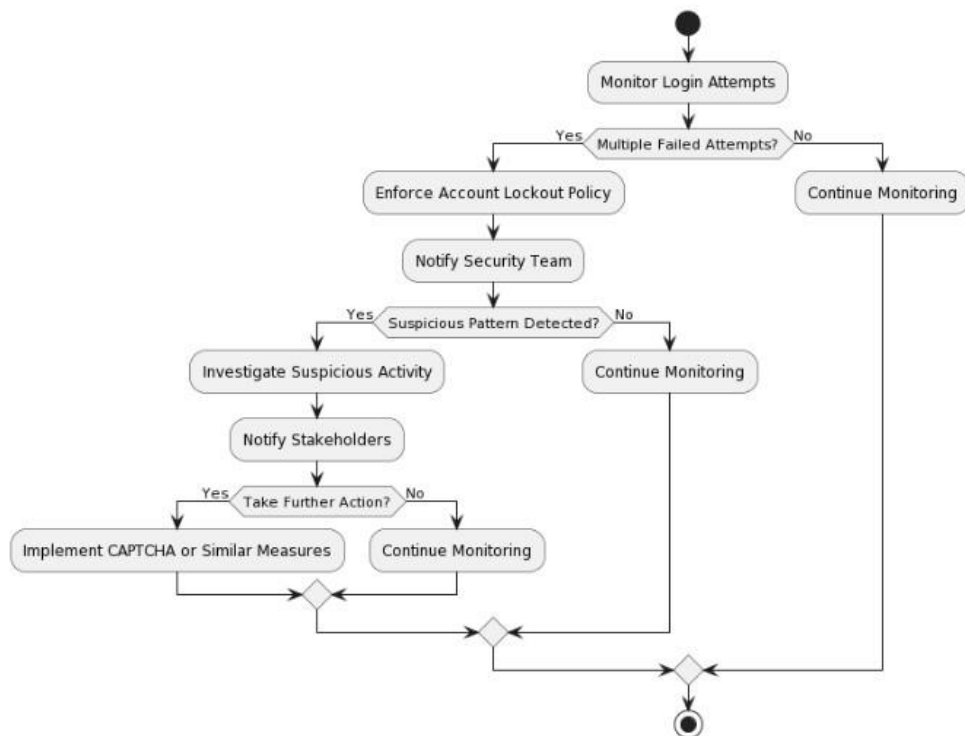
### 8.3. State chart Diagram:



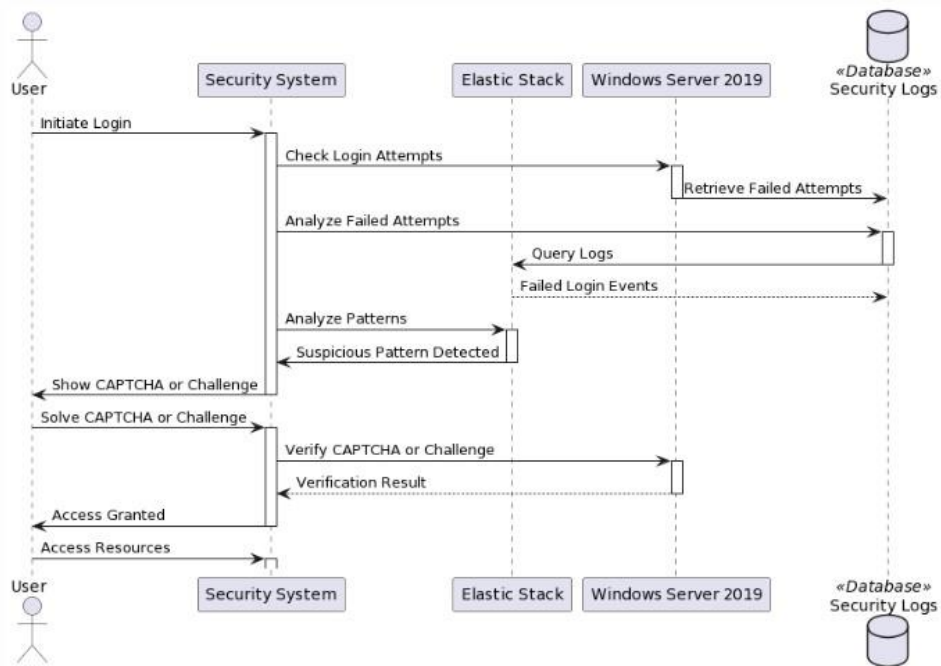
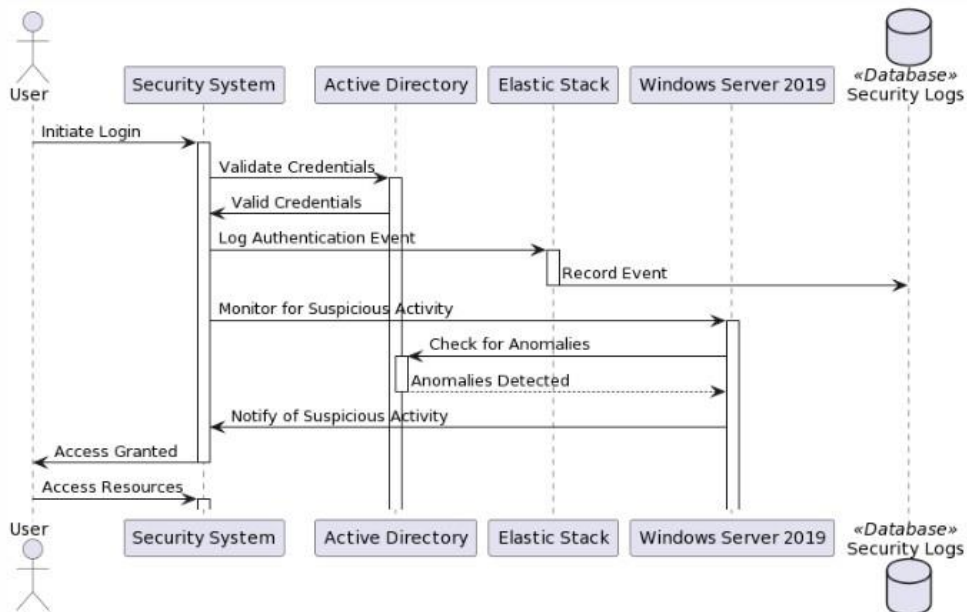
## 8.4. Activity Diagram:

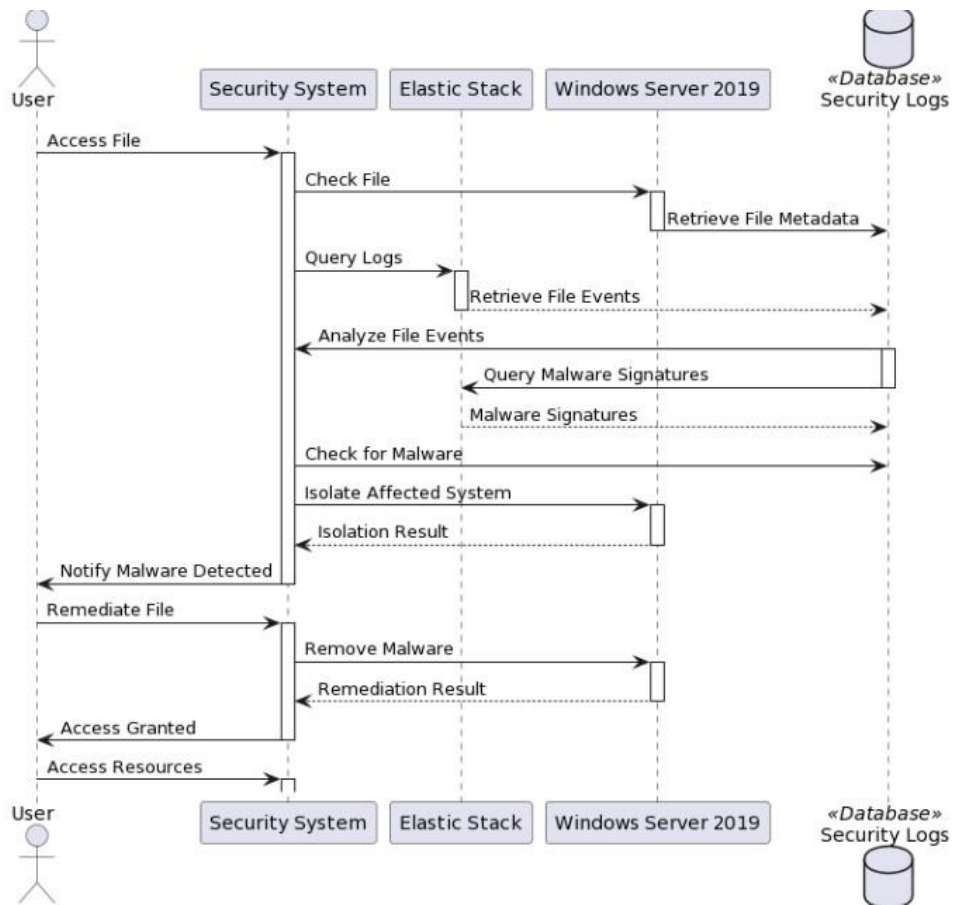




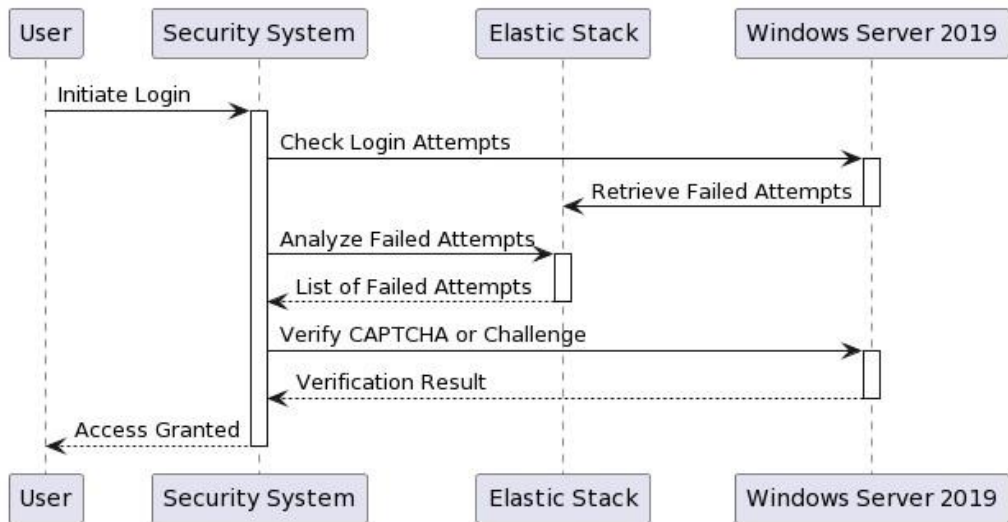
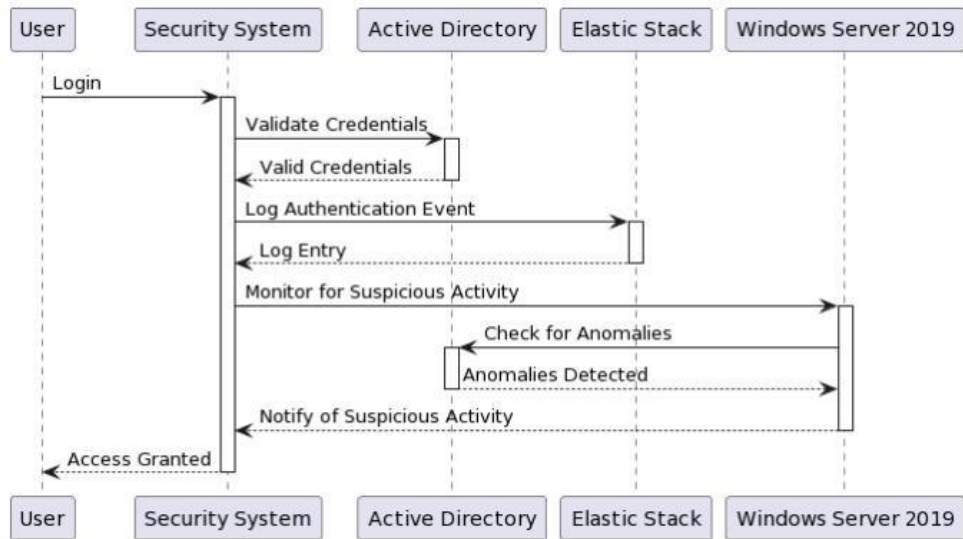


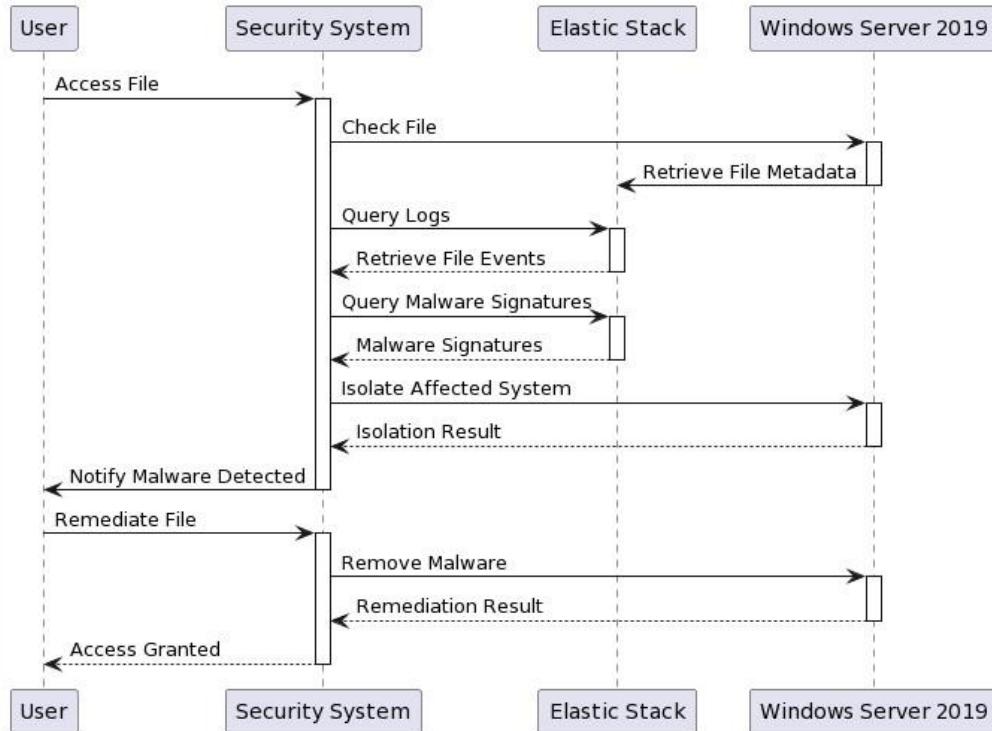
## 8.5. Sequence Diagram:



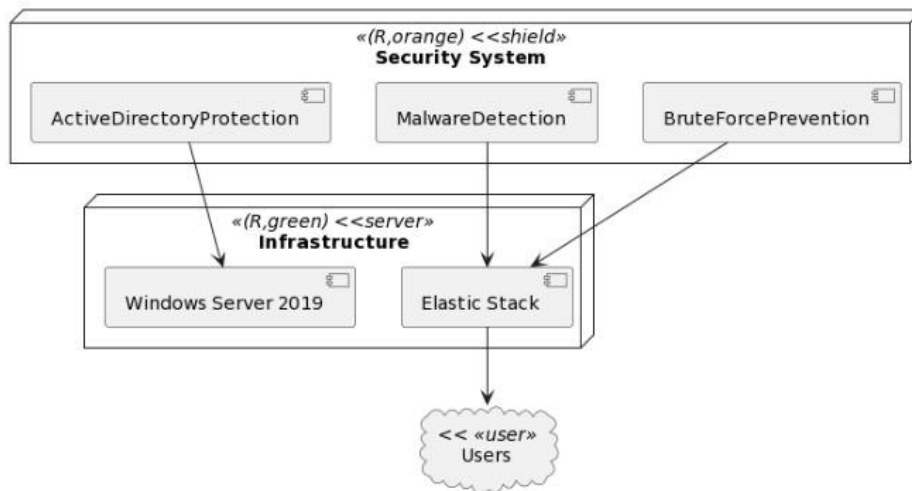


## 8.6. Collaboration Diagram:

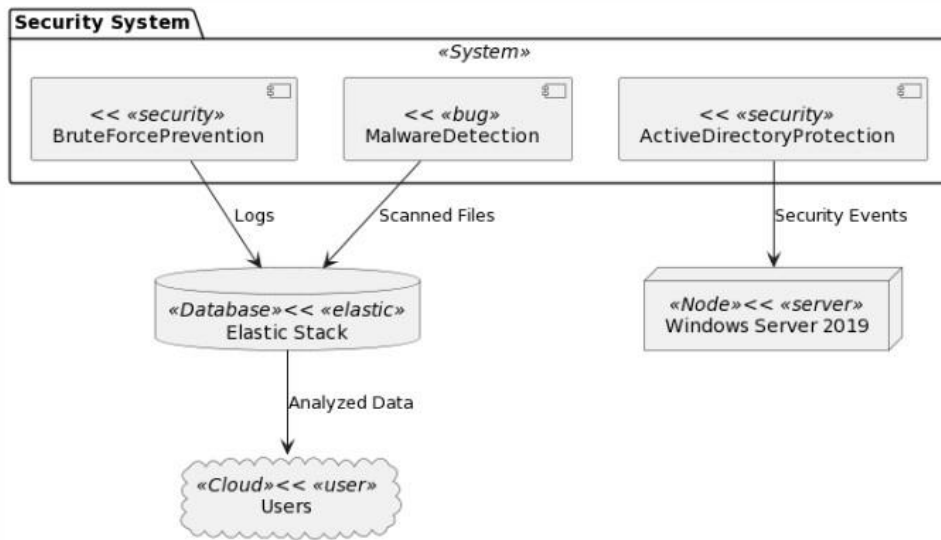




## 8.7. Deployment diagram:



## 8.8. System Block Diagram:



## **Test Cases:**

<b>Test Case Name</b>		Prevent Active Directory Attacks	<b>Test Case Description</b>		Secure Active Directory against potential threats		
<b>Created By</b>		Ali Iqbal	<b>Version</b>		1.1	<b>Date Tested</b>	
<b>S #</b>	<b>Prerequisites:</b>						
1	User is required to login and provide credentials						
<b>Step #</b>	<b>Step Details</b>		<b>Expected Results</b>	<b>Actual Results</b>		<b>Pass / Fail / Not</b>	
1	Provide valid credentials for an existing user.		User is logged in successfully.	User logged in successfully.		Pass	
2	Provide invalid credentials for an existing user.		User login fails with appropriate error.	User login failed as expected.		Pass	
3	Attempt multiple failed logins.		User account is locked after failed attempts.	User account locked as expected.		Pass	
4	Log in with valid credentials. 2. Check Elastic Stack for security logs.		Security logs capture successful login.	Security logs show successful login.		Pass	
5	Attempt multiple failed logins. 2. Check Elastic Stack for security logs.		Security logs capture failed login attempts.	Security logs show failed login attempts.		Pass	
6	Attempt multiple failed logins. 2. Monitor Elastic EDR for alerts.		Wazuh detects brute force attack.	Elastic EDR alerts indicate brute force attack.		Pass	
7	Wait for account lockout. 2. Attempt login after lockout period.		User account is unlocked, login is possible.	User account unlocked, login successful.		Pass	
8	Trigger suspicious login event. 2. Check Opsgenie for incident creation.		TheHive creates incident for suspicious login.	Incident created in Opsgenie as expected.		Pass	
9	Perform Active Directory actions. 2. Monitor real-time events in Elastic Stack and Elastic EDR.		Real-time events captured in Elastic Stack, Elastic EDR.	Events captured in Elastic Stack, Elastic EDR as expected.		Pass	
10	Trigger incident in Opsgenie. 2. Monitor for response actions (e.g., IP blocking).		Opsgenie initiates response actions as configured.	Response actions initiated by Opsgenie as expected.		Pass	



<b>Test Case Name</b>	Prevent Brute Force Attacks	<b>Test Case Description</b>	Implement measures to protect against unauthorized access		
<b>Created By</b>	Ali Iqbal	<b>Version</b>	1.1	<b>Date Tested</b>	
<b>S #</b>	<b>Prerequisites:</b>				
1	User is required to login and provide credentials				
<b>Step #</b>	<b>Step Details</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass / Fail / Not</b>	
1	Attempt multiple incorrect logins.	User account is locked out.	Account is locked after 5 failed logins.	Pass	
2	Check Elastic Stack for logs related to failed logins.	Logs capture failed login attempts.	Elastic Stack shows failed login events.	Pass	
3	Monitor Elastic EDR for alerts after multiple failed logins.	Elastic EDR detects the brute force attack.	Elastic EDR generates alerts for failed logins.	Pass	
4	Check Windows Server logs after failed logins for automated lockout enforcement.	Windows Server enforces account lockout.	Lockout policy is enforced after 5 failed logins.	Pass	
5	Monitor Elastic Stack and Elastic EDR for real-time detection of multiple failed logins.	Real-time monitoring captures attempts.	Real-time monitoring shows failed logins.	Pass	
6	Trigger failed logins, check Opsgenie for incident creation and automated response actions.	Opsgenie creates an incident; initiates response actions.	Incident created; response actions initiated.	Pass	
7	Monitor Elastic EDR for IP blocking after detecting a brute force attack.	Elastic EDR blocks the IP associated with attack.	IP address is blocked by Elastic EDR.	Pass	
8	Wait for account lockout and verify automatic unlock after the lockout period.	User account is automatically unlocked.	Account unlocks after 15 minutes.	Pass	
9	Simulate brute force attack; check Opsgenie for incident related to failed logins.	Opsgenie creates incident for attack.	Incident created for multiple failed logins.	Pass	
10	Trigger brute force attack; check Elastic Stack for correlated events.	Elastic Stack correlates events for attack.	Correlation in Elastic Stack for brute force attack.	Pass	

Test Case Name		Malware Detection and Remediation	Test Case Description		Detect and remediate malware on the system.		
Created By		Umair khan	Version		1.1	Date Tested	
S #	Prerequisites:						
1	User accesses a file or resource.						
Step #	Step Details		Expected Results	Actual Results		Pass / Fail / Not	
1	1. Introduce a known malicious file into the system. 2. Monitor Elastic Stack and Elastic EDR for detection.		Elastic EDR detects the malicious file, and Elastic Stack shows alerts.	Malicious file detected by Elastic EDR; Alerts displayed in Elastic Stack.		Pass	
2	1. Introduce a file with suspicious behavior. 2. Monitor real-time file activities using Elastic EDR.		Elastic EDR captures real-time file activities.	Real-time monitoring shows file activities.		Pass	
3	1. Trigger a malware event. 2. Check Opsgenie for the creation of an incident related to the malware.		Opsgenie creates an incident for the malware event.	Incident created in Opsgenie for the triggered malware event.		Pass	
4	1. Simulate a malware event. 2. Check Opsgenie for automated response actions (e.g., isolation, file quarantine).		Opsgenie initiates automated response actions.	Response actions initiated by Opsgenie for the simulated malware.		Pass	
5	1. Trigger multiple malware events. 2. Check Elastic Stack for correlation of events related to the malware.		Elastic Stack correlates and displays events related to malware.	Correlation in Elastic Stack for multiple triggered malware events.		Pass	
6	1. Introduce a known malicious file. 2. Check Windows Server logs and Elastic EDR for quarantine actions.		Windows Server logs show quarantine actions.	Quarantine actions recorded in Windows Server logs and Elastic EDR.		Pass	
7	1. Introduce a new malware variant. 2. Ensure that Elastic EDR has the latest malware signatures.		Elastic EDR updates its signatures and detects the new malware variant.	Elastic EDR updates signatures and detects the introduced malware.		Pass	
8	1. Trigger a malware event. 2. Monitor for IP blocking initiated by Elastic EDR in response to the malware.		Elastic EDR blocks the IP address associated with the malware event.	IP address is blocked by Elastic EDR in response to the triggered malware.		Pass	
9	1. Create an incident in TheHive. 2. Collaborate with analysts to investigate and remediate the malware event.		Analysts collaborate in Opsgenie to investigate and remediate.	Collaboration and investigation activities recorded in Opsgenie.		Pass	
10	1. Trigger a malware event. 2. Check Windows Server logs for remediation actions (e.g., file deletion, process termination).		Windows Server logs show remediation actions.	Remediation actions executed on Windows Server in response to malware.		Pass	

## Advisor Form:



SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE & TECHNOLOGY KARACHI CAMPUS

### Form IV: Student Log Form

Title: Security Process Fusion Through Automation

Supervisor: Dr. Hsraim Mansoor Batch/Sec: 2020 Group #: BSS

Reg. # (Group members): Ali Iqbal (201286), Umair Khan (201242)

Sr.	Task Assigned	Due	Task Completed (S)	Date (S)/Sign.
1	Proposal Submission		Done <del>yes</del>	
2	SRS <del>prot</del> making		Done	
3	Installation of elastic and <sup>make</sup> <del>integrity</del> <sup>integrity</sup>		Done	}
4	Configuration of elastic and <sup>integrity</sup>		Done	
5	Active <del>directory</del> and <sup>integrity</sup> <del>server</del>		Done	
6	Elastic EDR Integration		Done	}



SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE &  
TECHNOLOGY KARACHI CAMPUS

7	Malicious File execution and detection POC	Done	
8	Elastic EDR detection	Done	
9	Windows Security Logs enhancement and detection	Done	
10	Performing Remediation Via Elastic Live Response	Done	
11	Live Response Console POC	Done	
12	SDS half Completed	Done	
13	SDS Completed	Done	
14			
15			

Supervisor's Authentication (Completed report):

FYP Coordinator Authentication:

Dated: \_\_\_\_\_

Dated: \_\_\_\_\_



SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE &  
TECHNOLOGY KARACHI CAMPUS

**Form IV: Student Log Form**

Title: Security Policy through Fusion through automation

Supervisor: Dr. Hussain Mansoor Batch/Sec: 2020 Group #: BSCS

Reg. # (Group members): Ali Iqbal (2019255) Usair Khan (20192432)

Sr.	Task Assigned	Due	Task Completed (S)	Date (S)/Sign.
1	Deployment of <sup>management</sup> incident		Done	
2	Integration of <sup>management</sup> incident		Done	}
3	Deployment of AD		Done	
4	Integration of AD with ELK		Done	
5	DHCP configuration		Done	
6	AD Configuration		Done	}