



Hive Sentinel: The Hacker's Confine
Final Report FYP-1

Submitted by

Mariyam	B20F0165CS004
Muhammad Qaseem Ul Hassan	B20F0279CS013

Program: Computer Science FALL-2020

Semester: 7th

Submitted to

Dr. Abdul Waheed Khan

Pak Austria Fachhochschule Institute of Applied Sciences
and Technology, Mang Haripur

January 24, 2024

Abstract

In today's digital world, where businesses and individuals rely heavily on technology, cybersecurity is crucial. While firewalls and antivirus software offer basic protection, the changing tactics of cyber threats require a proactive approach. The "Hive Sentinel" project, a comprehensive solution addressing the critical issue of monitoring and managing honeypots to enhance cybersecurity. These specialized tools, strategically placed across networks, divert potential attackers, and provide invaluable insights into emerging threats. Influenced by the Deception 2.0 strategies, the project ensures proactive threat detection through features such as automated honeypot deployment, continuous health monitoring, real-time threat detection, and snapshot restoration. Moreover, the project aims to map the data to the standardized MITRE ATT&CK framework to get further insights into threat analysis.

Table of Contents

CHAPTER 1: Introduction.....	4
1.1 Introduction.....	5
1.2 Problem Statement.....	5
1.3 Scope.....	6
CHAPTER 2: Literature Review.....	8
2.1 Literature review	9
2.2 Key Findings:.....	10
CHAPTER 3: Software Requirement Specifications (SRS).....	11
3.1 Introduction.....	12
3.3 Objectives	12
3.4 Framework and Methodology.....	12
3.4.1 Strategies for Identifying Threats from Generated Logs Through Sysmon.....	14
3.4.1.1 Scenario 1: Sysmon Rule for Identifying DVWA attacks	14
3.5 Functional Requirements	16
3.6 Hardware Specification.....	17
3.7 Proposed Tools and Platforms	18
3.8 Project Timeline (Gantt chart)	20
CHAPTER 4: Software Design Document (SDD)	21
4.1 Introduction.....	22
4.2 Project Modules	22
4.3 Block Flow Diagram.....	23
4.4 Sequence Diagrams.....	24
4.6 Interface Design	30
4.7 References.....	35

CHAPTER 1

Introduction

1.1 Introduction

In today's interconnected world, where businesses and individuals heavily depend on digital systems, the importance of cybersecurity cannot be overstated. As our reliance on technology grows, so does the sophistication and frequency of digital threats and cyber-attacks. Traditional security measures, such as firewalls, which act as digital barriers to control incoming and outgoing network traffic, and antivirus software, designed to detect and eliminate malicious software, form the foundation of defence against common cyber threats.

In this dynamic and evolving digital environment, actively identifying, and addressing potential threats have become essential. Firewalls and antivirus tools are essential components, but they may not be sufficient on their own to tackle the ever-evolving tactics of cyber threats. Cybersecurity professionals recognize the value of a proactive approach to identify and understand the tactics, techniques, and procedures (TTPs) employed by malicious actors.

Honeypots, as a specialized cybersecurity tool, play a crucial role in this proactive defence strategy. By strategically placing decoy systems or resources across a network, organizations can lure potential attackers away from valuable assets and confidential information. These fake systems mimic genuine systems and services, inviting attackers to engage with them.

The primary objective of deploying honeypots is not just to divert and monitor unauthorized access attempts but also to gain insights into the attackers' methods and motivations. Analysis of the data collected from honeypots helps cybersecurity professionals better understand emerging threats, vulnerabilities, and attack patterns. This intelligence is invaluable for enhancing overall cybersecurity posture by enabling organizations to fine-tune their defences, update security protocols, and develop countermeasures against evolving cyber threats.

The information gathered from honeypots contributes to threat intelligence databases, enabling the global cybersecurity community to collaborate and share knowledge about emerging cyber threats. This collective intelligence helps organizations stay ahead of cyber adversaries and fortify their defences against potential future attacks. In essence, while traditional cybersecurity measures form a crucial baseline, honeypots add a proactive and adaptive dimension, enhancing the overall resilience of digital systems against an ever-evolving threat landscape.

1.2 Problem Statement

The lack of a dedicated health monitoring system for honeypots can lead to significant consequences for an organization's cybersecurity. Firstly, there is a risk of a false sense of security. If a honey pot is not functioning properly, it may fail to detect and capture malicious activities effectively. This can create a misleading belief that the network is adequately protected, leaving critical assets vulnerable to real threats. Secondly, the organization may face limited or inaccurate threat intelligence.

A compromised or malfunctioning honey pot may provide unreliable or incomplete information about attacker tactics and tools. This hampers the organization's ability to gather accurate threat intelligence, hindering their capacity to make informed decisions and take appropriate security measures. Furthermore, a vulnerable honey pot can introduce additional risk and exposure. Attackers can exploit the compromised honey pot as an entry point to infiltrate the network or gain insights into the organization's infrastructure. This not only

undermines the effectiveness of the honey pot but also puts other systems and valuable assets at risk.

1.3 Scope

Among the various security mechanisms discussed—firewalls, antivirus, and honeypots—this project, “Hive Sentinel: The Hacker's Confine” specifically focuses on automating honeypot deployment and health monitoring. The basic modules of project are discussed below as illustrated in Figure 1.1.

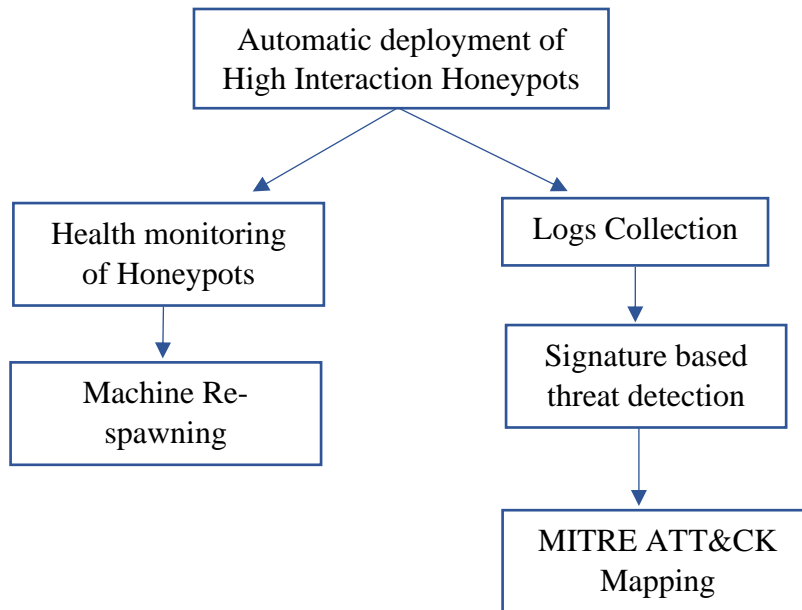


Figure 1.1: Basic Modules of Hive Sentinel

1) Automatic Deployment of Honeypots:

Hive Sentinel places a significant emphasis on automation to enhance the efficiency and effectiveness of cybersecurity measures. The project involves the automated deployment of high interaction honeypots as part of the deception platform 2.0. These decoy systems are strategically placed to lure potential attackers, providing valuable insights into their tactics and methodologies.

2) Logs Collection:

The project incorporates the essential process of log collection in honeypots, playing a vital role in detecting and analyzing malicious activities, offering insights into attack patterns, tools, and techniques for promptly identifying and containing security threats.

3) Signature Based Threat Detection:

The project integrates signature-based threat detection in honeypots, using predefined patterns or signatures of known threats to identify malicious activities.

4) MITRE ATT&Ck Mapping:

The project involves the strategic practice of mapping generated honeypot logs to MITRE ATT&CK, providing a structured framework for understanding and categorizing cyber threats, enhancing overall threat detection capabilities.

5) Health Monitoring:

The project involves a comprehensive approach to health monitoring, incorporating the collection of critical parameters such as CPU time and memory usage. This integrated health monitoring system enables the system to promptly detect any irregularities or potential issues, utilizing the extracted health monitoring parameters from each honeypot. This vigilance not only contributes to the early detection of threats but also supports the ongoing improvement of honeypot defenses.

6) Machine re-spawning:

An innovative feature of the project is the machine re-spawning functionality, which operates in accordance with health monitoring guidelines. This capability ensures that compromised or malfunctioning honeypots can be restore, maintaining a resilient and continuously effective deception platform.

CHAPTER 2

Literature Review

2.1 Literature review

Deception 1.0 initiated the use of basic decoys and honeypots, introducing the concept of misdirection in cybersecurity. Deception 2.0 addresses the deficiencies of Deception 1.0 by offering more advanced features. This progression signifies a sophisticated understanding of attacker behaviour and a commitment to staying ahead in the ever-evolving landscape of cybersecurity. A detailed overview of Deception 1.0 and Deception 2.0 is discussed below.

Deception 1.0 pioneered the use of honeypots and decoys to divert and monitor attackers, providing a foundational layer of defence. It provided valuable insights into attackers' methodologies but fell short in adaptability and automation. The demerits included static setups requiring manual intervention, limited scalability, and vulnerability to sophisticated attacks due to its lack of dynamic response mechanisms. While it helped detect threats, its effectiveness weakened against evolving cyber threats due to its static nature and reliance on easily recognizable decoys. [1]

Deception 2.0 marks a significant evolution from its predecessor, Deception 1.0, by introducing a range of advancements in the field of cyber threat mitigation. A notable improvement lies in its seamless integration of both low and high-interaction methods, offering a fluid and adaptable approach termed Hybrid Interactivity/Fluid Deception.

Unlike Deception 1.0, the new version takes customization to the next level by tailoring low-interaction services to evade emulations, thus enhancing the overall deception strategy. Scalability is efficiently addressed through a dynamic DevOps model, overcoming deployment scale challenges by automating deployment and maintenance processes, ensuring ongoing freshness, and operational efficiency.

Security measures receive a substantial boost with the introduction of deception farms and projection points, strategically placed throughout the network as virtual traps. Deception farms act as decoy fields, confusing attackers by blending real assets with fake ones, reducing the risk of security breaches. Projection points, key locations for decoys, intensify the overall deception strategy, creating a dynamic defence that disrupts attackers and makes compromising the digital environment tougher.

Moreover, the expanded deception range goes beyond traditional honeypots, allowing the system to mimic a diverse array of enterprise resources. This expansion is complemented by ecosystem integration, as Deception 2.0 actively engages with the broader security ecosystem, promoting collaboration for more effective threat detection and response.

The implementation of intrinsic threat analysis further sets Deception 2.0 apart, ensuring continuous awareness and proactive responses to emerging threats. This dynamic and sophisticated component establishes its status as a powerful tool in modern threat mitigation strategies. In comparison, Deception 1.0 may be seen as more static, lacking the integrative and adaptive features that characterize the advancements introduced in Deception 2.0. [2]

2.2 Key Findings:

Based on the literature review, it has been established that a high-interaction honeypot must satisfy the following conditions to effectively engage with threats:

- 1. Comprehensive Deception Setup:**

The honeypot must feature multiple decoys, breadcrumbs, and baits strategically installed to identify exploits used and lateral movement paths.

- 2. Interconnected Decoys for Lateral Movement:**

Lateral movement within the network should lead the attacker to other decoys, enhancing the honeypot's effectiveness.

- 3. Invisible Collection Mechanism:**

The collection mechanism must remain invisible to the attacker, ensuring covert monitoring of their activities.

- 4. Network Traffic Analysis:**

Capture and correlation of network traffic with the attacker's actions, aiding in identifying command and control centers, exfiltration activities, etc.

- 5. Memory Dump Analysis:**

Capability to review memory dumps to identify any files not saved to disk, providing a deeper understanding of the attacker's activities.

- 6. Tracking Downloaded or Dropped Files:**

Ability to track downloaded or dropped files, enabling the analysis of potentially malicious content.

CHAPTER 3

Software Requirements Specifications (SRS)

3.1 Introduction

In software development, the Software Requirements Specification (SRS) document is like a guide, giving a detailed plan for the entire project. It's crucial for everyone involved—clients, developers, and testers—to understand what the software is meant to do.

The SRS document is essential because it acts as a roadmap for the development team throughout the project. It helps communication between clients and developers and serves as a reference point for the project's different stages. A well-done SRS reduces misunderstandings, lowers project risks, and sets the stage for successful software.

This chapter encompasses critical components fundamental to the project's success, providing insights into objectives, framework and methodology, functional requirements, hardware specifications, proposed tools, platforms, and the anticipated project timeline.

3.3 Objectives

Hive Sentinel is strategically designed to focus on advancing the capabilities of Deception 2.0 by delivering a range of advanced functionalities. Following objectives will enable the accomplishment of main aim of the project:

1. To provide automated deployment of high interaction honeypots for the deception platform 2.0.
2. To offer health monitoring for high interaction honeypots on the deception platform 2.0.
3. To provide machine re-spawning in accordance with health monitoring guidelines.
4. To enable hackers' intelligence gathering from high interaction honeypots.
5. To present a dashboard that displays gathered intelligence information through visual charts and statistics.

3.4 Framework and Methodology

To efficiently execute the process of creating Hive Sentinel, a robust combination of frameworks and tools is essential. Figure 3.1 presents the actions and activities that are necessary for achieving an effective deployment of a honeypot system. A sequence of activities is illustrated in Figure 3.1 and discussed below.

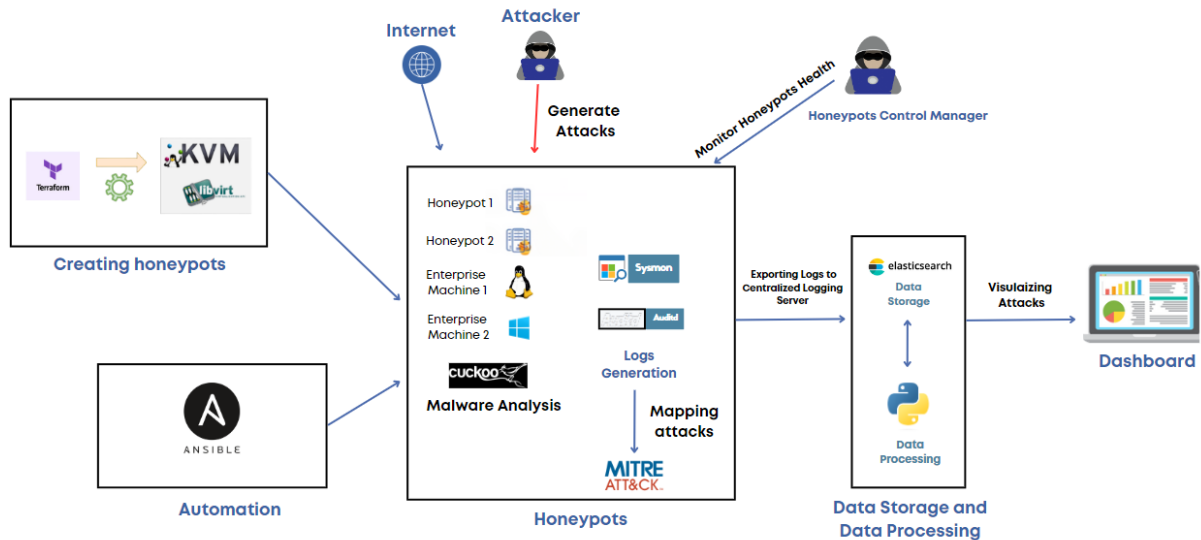


Figure 3.1: Project Architecture Diagram

As Illustrated in Figure 3.1 , the project begins with virtualization; Kernel-based Virtual Machine (KVM) facilitates hardware virtualization on Linux, allowing the concurrent operation of multiple virtual machines on a single host. For automation, Terraform streamlines infrastructure provisioning and management through code, helping to automatically create virtual machines on KVM. Ansible simplifies configuration, deployment, and task automation, proving particularly useful for automating the deployment of vulnerable services on honeypots.

Cuckoo Sandbox automates the analysis of suspicious files, contributing to the security posture. Process monitoring involves implementing Sysmon on Windows and Auditd on Linux, enhancing system security through detailed logging of process executions and network connections.

Security infrastructure integration is achieved through the incorporation of MITRE ATT&CK with generated logs from Sysmon and Auditd, providing a comprehensive framework for understanding cyber adversary tactics and techniques. After successfully generating logs and mapping them with the MITRE ATT&CK framework, export the logs to a centralized Elasticsearch database. Elasticsearch efficiently stores and indexes large datasets, enabling centralized monitoring.

Process the logs stored in Elasticsearch through Python to ensure they are in the same format, as they come from different honeypot machines running Windows and Linux. Visualization is facilitated by developing a dynamic React.js dashboard with a robust Python and Flask backend to monitor health parameters of honeypots, create machines, analyze MITRE ATT&CK techniques, control management of honeypots.

This integrated approach ensures a comprehensive cybersecurity framework, seamlessly combining virtualization, automation, monitoring, database management, security integration, and visualization, thereby ensuring an efficient honeypot system.

3.4.1 Strategies for Identifying Threats from Generated Logs Through Sysmon

System Monitor (Sysmon) is a Windows-based device driver and system service which keeps track of all system activities even after a system reboot and logs it to the Windows event log. It augments the regular Windows logs with higher-level monitoring of events. It gives detailed information on the development of processes, network connections, and changes to the time taken to create files in a repository. Sysmon logs will be collected and exported to a centralized database for analysis, aiming to detect anomalous and potentially malicious activities.

Sysmon has the following capabilities:

- 1) Logs process creation with complete command line
- 2) Logs file hashes (MD5, IMPASH, SHA256)
- 3) Process GUID (Global User ID) to correlate process events where IDs are reused.
- 4) Session GUID to keep track of events and logs generated during a logon session
- 5) Also logs network connection events, including Source and Destination IP/Port.
- 6) Capable of logging modifications to file timestamps, a common technique utilized by malware for defence evasion purposes.
- 7) Filtering rules to include or exclude certain events

Sysmon logs are stored under its own channel under “Applications and Services Logs” in the standard Windows Event Viewer. Sysmon monitors and logs events based on Event ID, where each event is assigned a unique ID. The configuration file for Sysmon, typically named "sysmonconfig.xml," plays a crucial role in determining which events are monitored and logged. This XML-based configuration file allows users to customize Sysmon's behaviour by specifying rules and filters for different types of events. By specifying customized rules in Sysmon configuration file we can create signature-based detection.

3.4.1.1 Scenario 1: Sysmon Rule for Identifying DVWA attacks

The Damn Vulnerable Web Application (DVWA) is a deliberately vulnerable web application designed for educational and testing purposes. It contains various vulnerabilities that users can exploit to understand and learn about common web application security issues.

Sysmon Configuration rule:

```

    <!-- DVWA Attack Logs -->
    <DestinationPort name="DVWA Attack Port" condition="is">80</DestinationPort>
  </NetworkConnect>

  <!-- Process Creation -->
  <ProcessCreate onmatch="include">
    <ImageName condition="contains">cmd.exe</ImageName>
    <ImageName condition="contains">powershell.exe</ImageName>
  </ProcessCreate>
</RuleGroup>

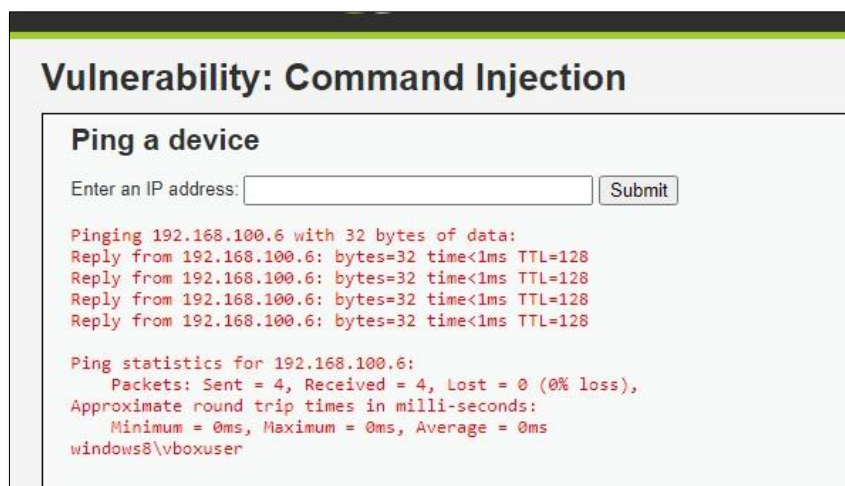
```

Figure 3.2: Sysmon Configuration rule for DVWA

As illustrated in Figure 3.2, Sysmon rule is implemented to bind to port 80, the port on which DVWA is running. This rule is designed to trigger Sysmon to generate logs whenever events occur on port 80, specifically capturing commands executed through the command line (cmd) or PowerShell.

Command Injection Attack on DVWA:

A command injection attack is a type of security exploit where an attacker injects malicious commands into a command interpreter or shell, often through a vulnerable web application. This attack occurs when an application passes not filtered user input to a system shell, allowing the attacker to execute arbitrary commands on the underlying system.

*Figure 3.3: Command Injection attack on DVWA*

Executing command: 'ping 192.168.100.6 && whoami' through input field as illustrated in Figure 3.3. This command will ping 192.168.100.6 and find the current user of device through cmd.

Sysmon generated log for command injection attack according to rule in figure 3.2:

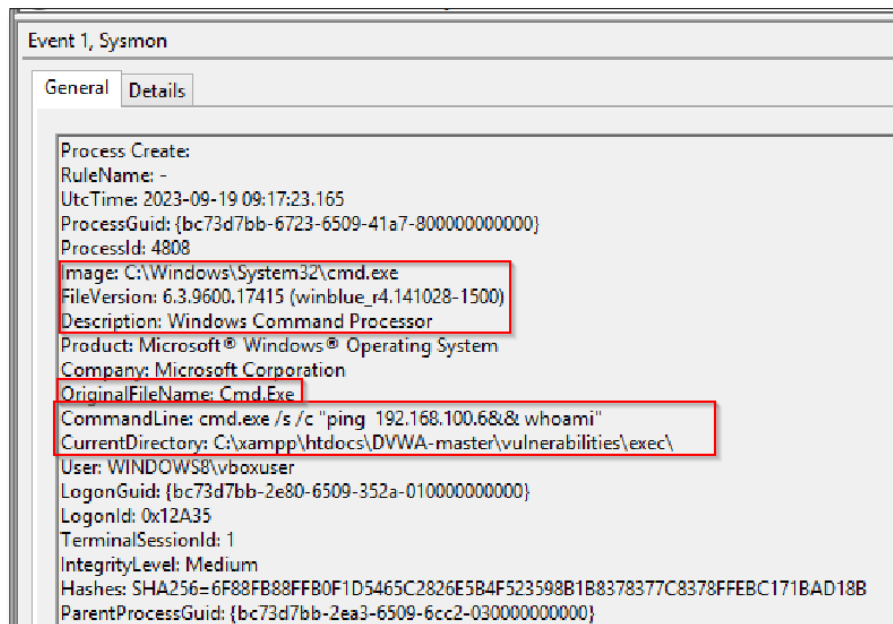


Figure 3.4: Sysmon log for DVWA attack

3.5 Functional Requirements

The project must meet the following functional requirements to establish an effective honeypot system that will efficiently contribute to proactive cybersecurity defense mechanisms.

1. A dashboard providing the following functionalities:
 - Automatically create virtual machines and deploy vulnerable services on them.
 - Monitor health parameters of honeypots and facilitate machine re-spawning.
 - Analyze MITRE ATT&CK techniques.
 - Control the management of honeypots.
2. Continuous monitoring of system resources, network connectivity, and application services of honeypots.
3. Real-time detection and alerting of suspicious or malicious activities targeting the honeypots.
4. Generation of detailed logs, reports, and analysis of honeypot activities for threat intelligence purposes.
5. Mapping of generated logs with MITRE ATT&CK to get insights into hacker tactics, techniques, and procedures.
6. Monitor and organize security events in one central place.
7. Control management system to monitor health of honeypots.
8. Snapshot restoration capabilities to reset honeypots to a clean state after attacks or specific penetration levels.

3.6 Hardware Specification

1. Server: Dell PowerEdge R740

In the Hive Sentinel project, a server is essential for creating multiple virtual machines, including honeypots and real machines. The server has the capability to run 24/7, aligning with the project's requirement for continuous honeypot operation to protect the real system. The Dell PowerEdge R740 server fulfills the project needs.

The Dell PowerEdge R740 is a high-performance server known for its reliability and scalability. Its robust architecture and management capabilities make it an ideal choice for handling complex tasks like control management and health monitoring in honeypot projects. It offers features for remote management, ensuring efficient control and monitoring of honeypots spread across different environments.

2. RAM: 64 GB

The 64 GB RAM capacity is crucial for this project as it allows for seamless multitasking and data handling. In a honeypot environment, where multiple instances might run simultaneously, this ample RAM ensures that the system can process and analyze a significant amount of data without experiencing performance bottlenecks. It supports concurrent processes for real-time monitoring and analysis of activities within the honeypots.

3. CPU Cores: 8

In this project, the utilization of 8 CPU cores is essential to leverage substantial processing power. These cores enable efficient parallel processing, allowing for quick analysis of incoming data, rapid responses to potential threats, and the ability to run multiple monitoring and analysis tools concurrently. This capability is vital for ensuring timely threat detection and response in a dynamic environment.

4. Storage: 1 TB

In this project, a 1 TB storage capacity is essential for storing logs, captured data, and information collected from honeypots. In a security monitoring scenario, where vast amounts of data are generated regularly, having sufficient storage is critical for retaining historical data for forensic analysis, identifying patterns, and investigating security incidents. It provides the necessary space to archive logs and maintain a comprehensive record of honeypot activities.

These specifications collectively form the foundation for an efficient and robust infrastructure, capable of handling the diverse demands of control management and health monitoring within a honeypot project. They ensure that the system remains responsive, capable of handling extensive data processing, and equipped to support the project's objectives of security analysis and threat detection.

3.7 Proposed Tools and Platforms

1. Virtualization

- **KVM:**

KVM (Kernel-based Virtual Machine) is a Linux kernel module that enables hardware virtualization. It allows running multiple virtual machines (VMs) on a single physical host providing a flexible and efficient platform for virtualization.

2. Automation

- **Terraform:**

Terraform is an open-source tool for automating the provisioning and management of infrastructure through code. Using a simple declarative language, users define their desired infrastructure state, and Terraform handles the coordination of resources across various cloud providers or on-premises environments. It enables efficient and repeatable infrastructure deployment, making it easier to scale, modify, and maintain complex systems with minimal manual intervention. Used to automate the creation of virtual machines.

- **Ansible:**

Ansible is an open-source automation tool that simplifies configuration management, application deployment, and task automation. It uses YAML syntax for playbooks, defining tasks and configurations in a human-readable format. Ansible does not require agents on managed hosts, making it agentless and easy to use for automating IT processes. Used to automate the deployment of vulnerable services on honeypots.

3. Process Monitoring

- **Sysmon and Auditd:**

Sysmon (Implemented on Windows) and Auditd (Implemented on Linux) enhance system security by logging detailed information like process executions and network connections, aiding in threat detection. Generated logs provide valuable forensic data for post-incident analysis, enabling proactive monitoring. This comprehensive logging ensures visibility and accountability, crucial for maintaining a secure and compliant computing environment. Implement Sysmon and Auditd to gain insights into process activities within the honeypot environment through logs generation.

- **Cuckoo Sandbox:**

Cuckoo Sandbox is an open-source malware analysis system that automates the execution of suspicious files in a controlled environment. It provides detailed reports on malware behavior, helping analysts understand and respond to potential threats. Cuckoo Sandbox supports various analysis techniques, making it a valuable tool for cybersecurity professionals in malware research and incident response.

4. Database:

- **Elastic Search:**

Elasticsearch is a distributed, open-source search and analytics engine known for its scalability and real-time search capabilities. It efficiently stores and indexes large volumes of data, accommodating complex queries and full-text searches. Widely utilized in applications such as log analytics, monitoring, and enterprise search, Elasticsearch enables centralized monitoring. By integrating a monitoring system with Elasticsearch, data from diverse machines are collected and combined in a centralized repository, enhancing the overall efficiency of analysis and insights.

5. Security Infrastructure Integration:

- **MITRE ATT&CK:**

MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) serves as a knowledge base, providing a comprehensive framework to comprehend the tactics and techniques employed by cyber adversaries. It categorizes and describes diverse attack methods, empowering cybersecurity professionals to strengthen threat detection, analysis, and response strategies. Integrating generated logs with MITRE ATT&CK provides valuable insights into hacker tactics, techniques, and procedures, enhancing the understanding of potential security threats.

6. Visualization:

- Develop a dashboard with React.js for a dynamic front end. Utilize Python and Flask for a robust backend, handling data requests and facilitating seamless communication.

7. Attack Emulation:

- **Atomic Red Team and Caldera:**

Atomic Red Team, an open-source framework for security testing, offers a library of small, modular tests to evaluate and enhance detection capabilities. Caldera, a threat emulation platform, leverages Atomic Red Team's tests to create a controlled environment for assessing how well honeypots detect and respond to cybersecurity threats. Both Atomic Red Team and Caldera are frameworks that assist security teams in testing their detection capabilities by simulating real-world attacks.

3.8 Project Timeline (Gantt chart)

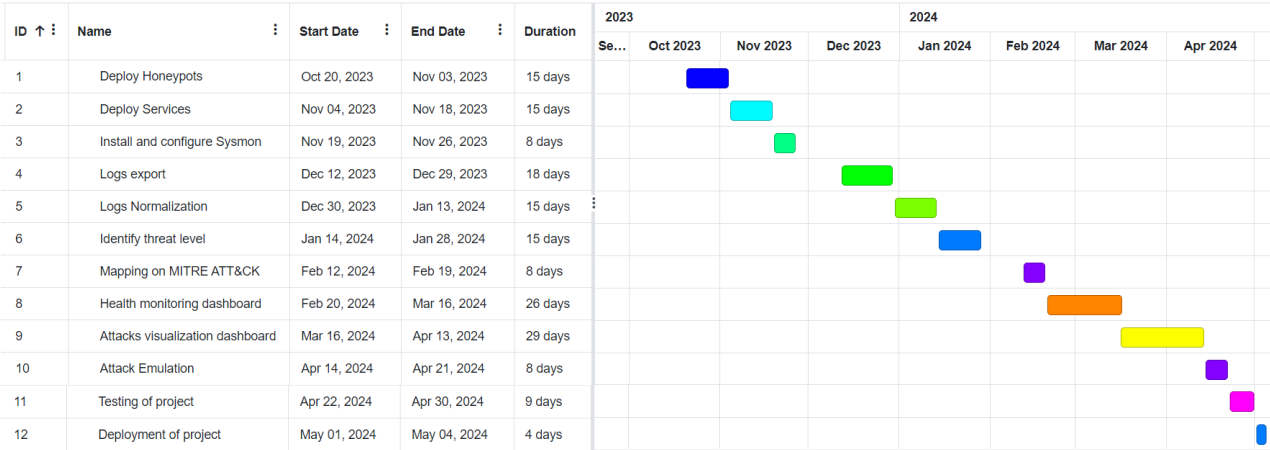


Figure 3.5: Gantt chart

CHAPTER 4

Software Design Document (SDD)

4.1 Introduction

The Software Design Document (SDD) serves as a comprehensive guide outlining the architecture, functionality, and structure of a software system. It is a crucial artifact in the software development life cycle, providing a roadmap for developers, designers, and stakeholders to understand the details of the proposed system. By examining the design principles, modules, and interactions, the SDD aims to provide documentation aid in software development, detailing how the software should be built.

The purpose of the design document for "Hive Sentinel: The Hacker's Confine" entails a comprehensive overview covering technical and functional aspects to guide effective development. This chapter includes both textual and graphical details, such as the Block Flow diagram, Sequence diagrams, Database design, and Mock-ups of the web application. It outlines the high-level architecture of the system, defining main components, their relationships, and interactions, establishing the foundation for the entire project.

4.2 Project Modules

Project is divided into following modules for efficient development:

1. Create Machines

- This module focuses on automating the process of creating machines through Terraform and deploying services on the created machines through Ansible from dashboard.

2. Logs Generation and Storage

- This module focuses on generating logs on honeypots in response to event detected through Sysmon and Auditd.
- Exporting logs on centralized database Elastic search.
- Display logs on dashboard.

3. MITRE ATT&CK Mapping

- This module focuses on mapping generated logs with MITRE ATT&CK Techniques.
- Display MITRE ATT&CK mapping on dashboard.

4. Identify Threat

- This module focuses on identifying threat from generated logs with the help of signature-based detection and displaying on dashboard.
- Signature-based detection include comparing Sysmon logs with pre-defined attack list of python and identifying threats.

5. Health Monitoring

- This module focuses on fetching health monitoring parameters from honeypots
- Display health of honeypots on dashboard
- Honeypot Re-spawning based on machine health

4.3 Block Flow Diagram

A block flow diagram for the project, "Hive Sentinel: The Hacker's Confines," offers a simplified representation of the overall structure and key components illustrated in Figure 4.1. Labelled blocks signify crucial stages, while arrows illustrate the flow or sequence of actions. This high-level overview aids in visualizing and comprehending the project's structure and functionality.

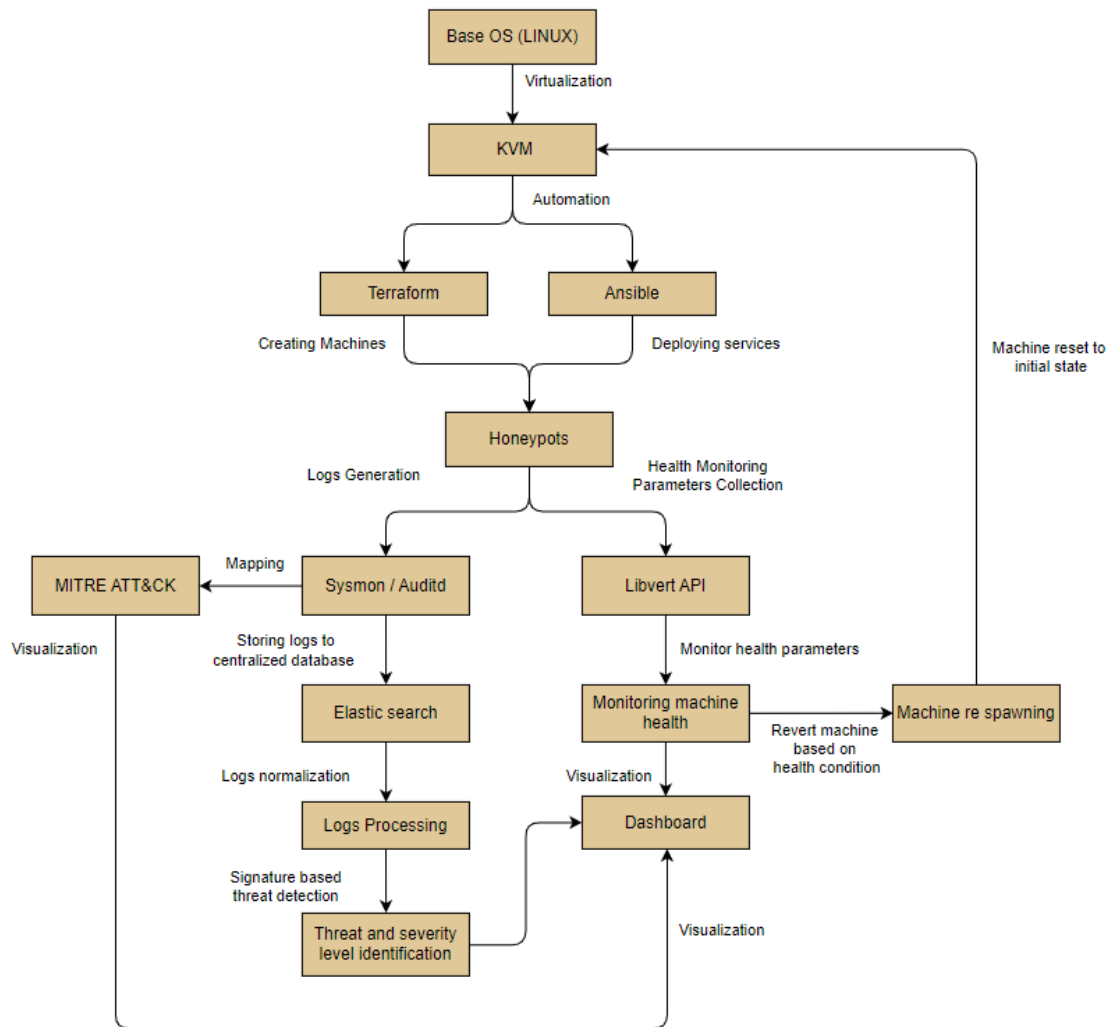


Figure 4.1: Block Flow Diagram

As depicted in Figure 4.1, the initial phase involves establishing the foundational environment by installing the Linux operating system and implementing KVM for virtualization. The infrastructure deployment follows a systematic approach using Terraform and Ansible. Terraform is employed to define the infrastructure as code and deploy virtual machines and services on KVM, while Ansible is used for deploying services on virtual machines.

Moving on to the log generation path, Sysmon/Auditd configurations are implemented on the virtual machines to generate logs, which are then mapped to MITRE ATT&CK for visualization. The logs are pushed to Elasticsearch for storage, normalized for consistency, processed to identify relevant information, and subjected to signature-based threat detection

with severity levels identified. The resulting data is displayed on a dashboard that includes MITRE ATT&CK mapping, threat information, and severity levels.

In parallel, the health monitoring path utilizes Libvirt API for fetching health monitoring parameters from machines, displaying them on the dashboard. Thresholds are set for health parameters, and if breached, a snapshot is applied, and machines are respawned within the KVM environment. The KVM state is then updated to reflect the changes. This comprehensive approach ensures a robust and secure virtualized environment with effective log generation, threat detection, and health monitoring.

4.4 Sequence Diagrams

Sequence diagrams for the project, "Hive Sentinel: The Hacker's Confines," illustrates the interactions and chronological order of messages exchanged among objects or components within a system. They provide a dynamic view of how various entities collaborate to accomplish a specific task or process.

4.4.1 Login

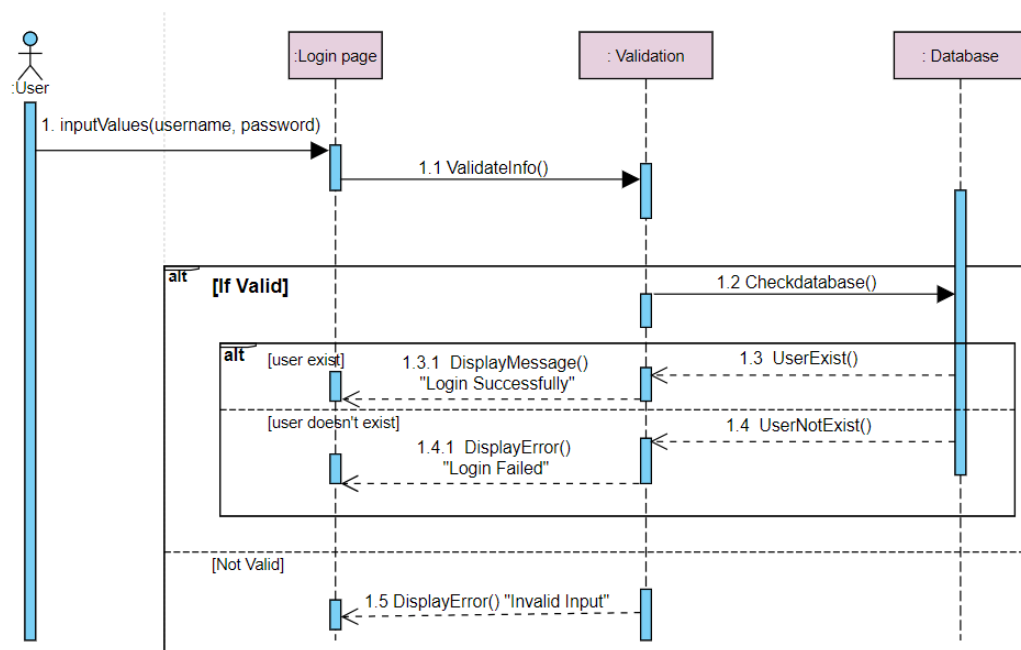


Figure 4.2. Sequence diagram of login page

Sequence of actions as illustrated in Figure 4.2 are described below:

1. User will enter the required fields username and password in Login page. After clicking Login button, the Login page will check the validations.
2. If the user doesn't fill all the required data in form or submit wrong information that is against the validation rules, error message will be displayed "invalid input" and system will remain on login page.

3. After correctly entering the Login credentials, it will be checked in database for authentication.
4. If the user exist in database following message will be displayed on screen “Login Successfully” and will move to Overview page.
5. If the user doesn't exist in database following message will be displayed “Login Failed”.

4.4.2 Create Machines

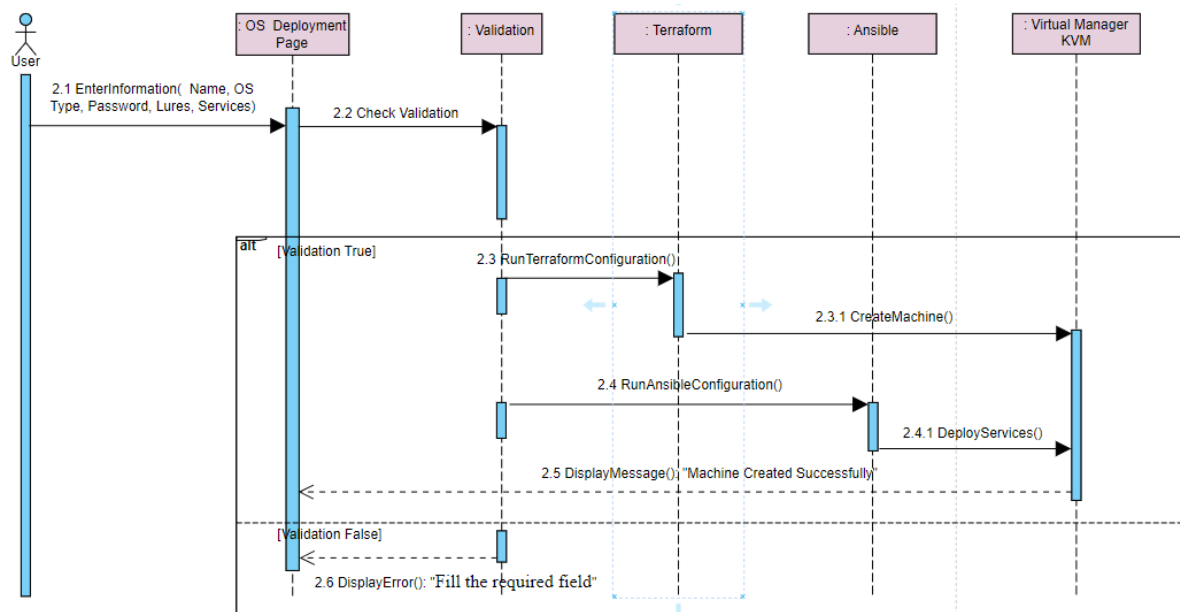


Figure 4.3: Sequence diagram of OS Deployment page

Sequence of actions as illustrated in Figure 4.3 are described below:

1. User will enter the required fields Name of machine, OS type, Password, Services and Lures in OS Deployment page. After clicking create button, the OS Deployment page will check the validations.
2. If the user doesn't fill all the required data in form, system will display error message “Fill the required field” and will remain on OS deployment page.
3. After correctly entering all the required fields, Terraform configuration will run to create machine on KVM.
4. After successfully machine is created, Ansible configuration will run to deploy services on machine.

4.4.3 View Deployed Machines

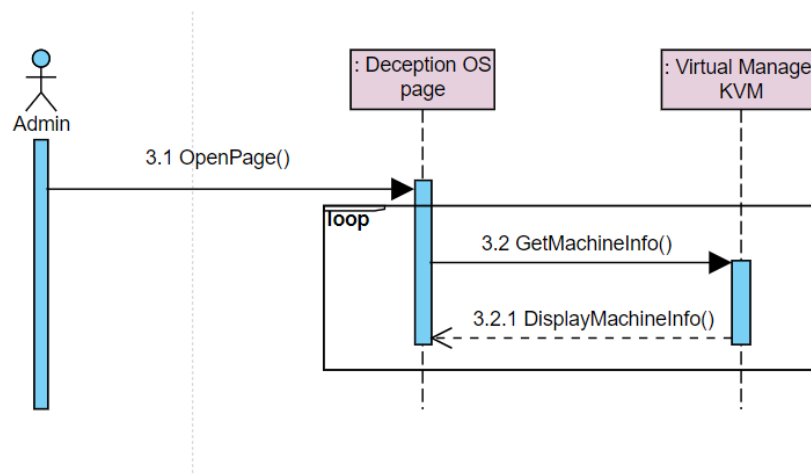


Figure 4.4: Sequence diagram of Deception OS page

Sequence of actions as illustrated in Figure 4.4 are described below:

1. User will open Deception OS page.
2. All the information regarding deployed machines will be fetched from Virtual Manager and displayed on page.

4.4.4 Logs Analysis

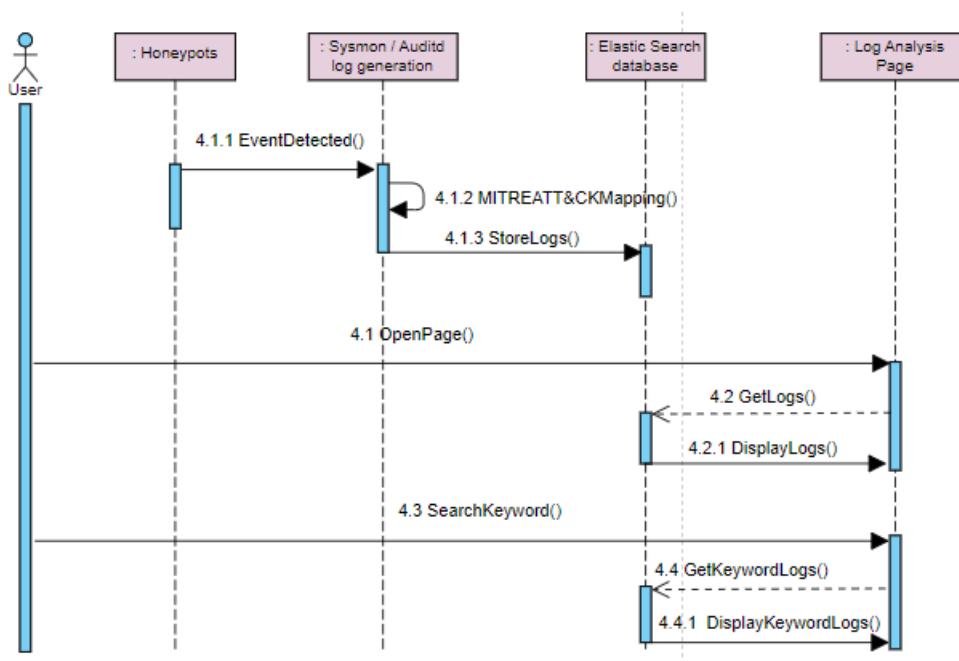


Figure 4.5: Sequence diagram of Log Analysis page

Sequence of actions as illustrated in Figure 4.5 are described below:

1. User will open Log Analysis page.
2. Logs are generated by Sysmon or Auditd on the honeypot with MITRE ATT&CK mapping, and then these logs are exported or pushed to the Elastic search database. Subsequently, they can be fetched from the Elastic search database and displayed on the dashboard.

4.4.5 MITRE ATT&CK Mapping

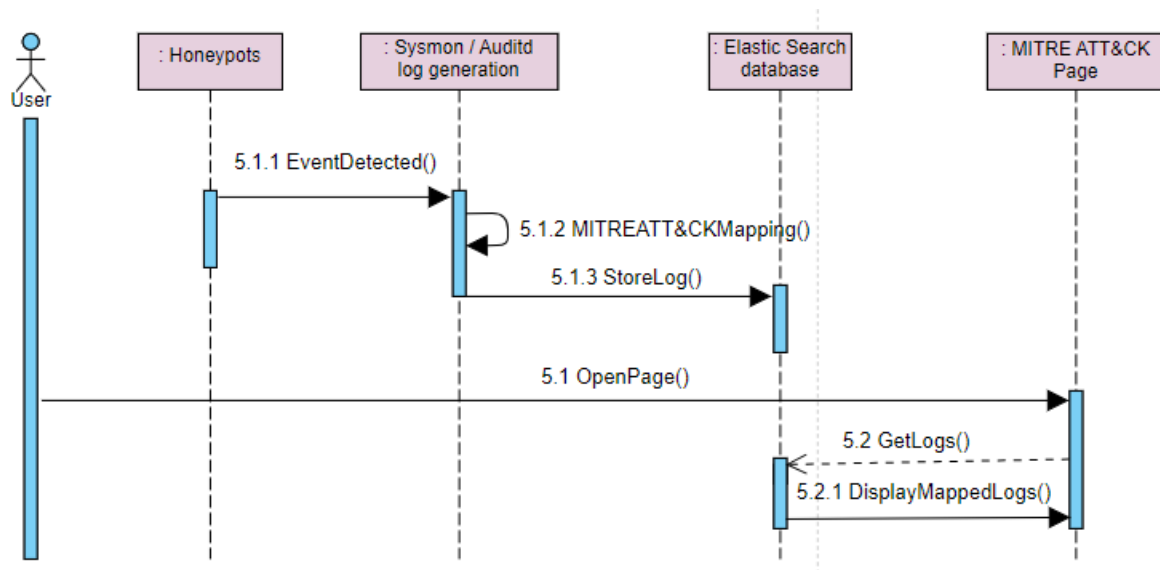


Figure 4.6: Sequence diagram of MITRE ATT&CK page

Sequence of actions as illustrated in Figure 3 are described below:

1. User will open MITTRE ATT&CK page.
2. Generated logs with MITRE ATT&CK mapping will be fetched from Elastic search database and will be displayed on dashboard.

4.4.6 Health Monitoring

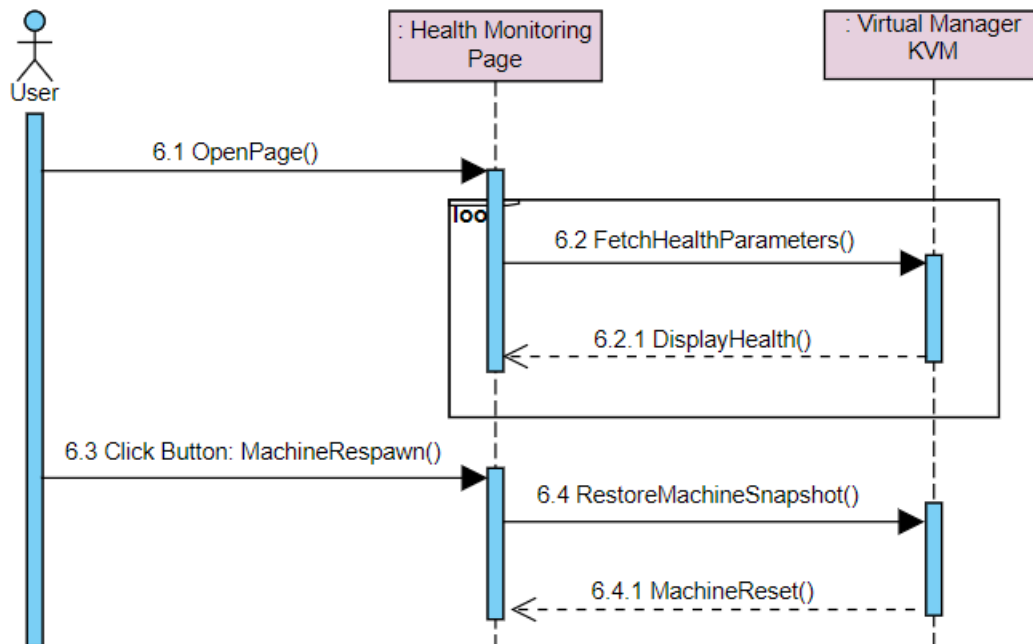


Figure 4.7: Sequence diagram of Health Monitoring page

Sequence of actions as illustrated in Figure 4.7 are described below:

1. User will open Health Monitoring page.
2. Health Monitoring parameters such as CPU and memory usage of each machine in Virtual Manager will be fetched through Libvirt API and displayed on dashboard.
3. Thresholds are set for health parameters, and if breached, a snapshot is applied, and machines are respawned within the KVM environment to its initial state.

4.5 Database Design

Hive Sentinel incorporates Elasticsearch database for its powerful search and analytics capabilities, making it an ideal choice for handling vast datasets. Its efficient storage and retrieval mechanisms, based on distributed indexing and sharding, enable quick and scalable data access. Elasticsearch excels at handling both structured and unstructured data in real-time, making it valuable for applications such as log analysis. Its ability to deliver fast and relevant search results, coupled with distributed architecture for resilience, makes Elasticsearch a popular choice for organizations seeking robust data exploration and retrieval solutions.

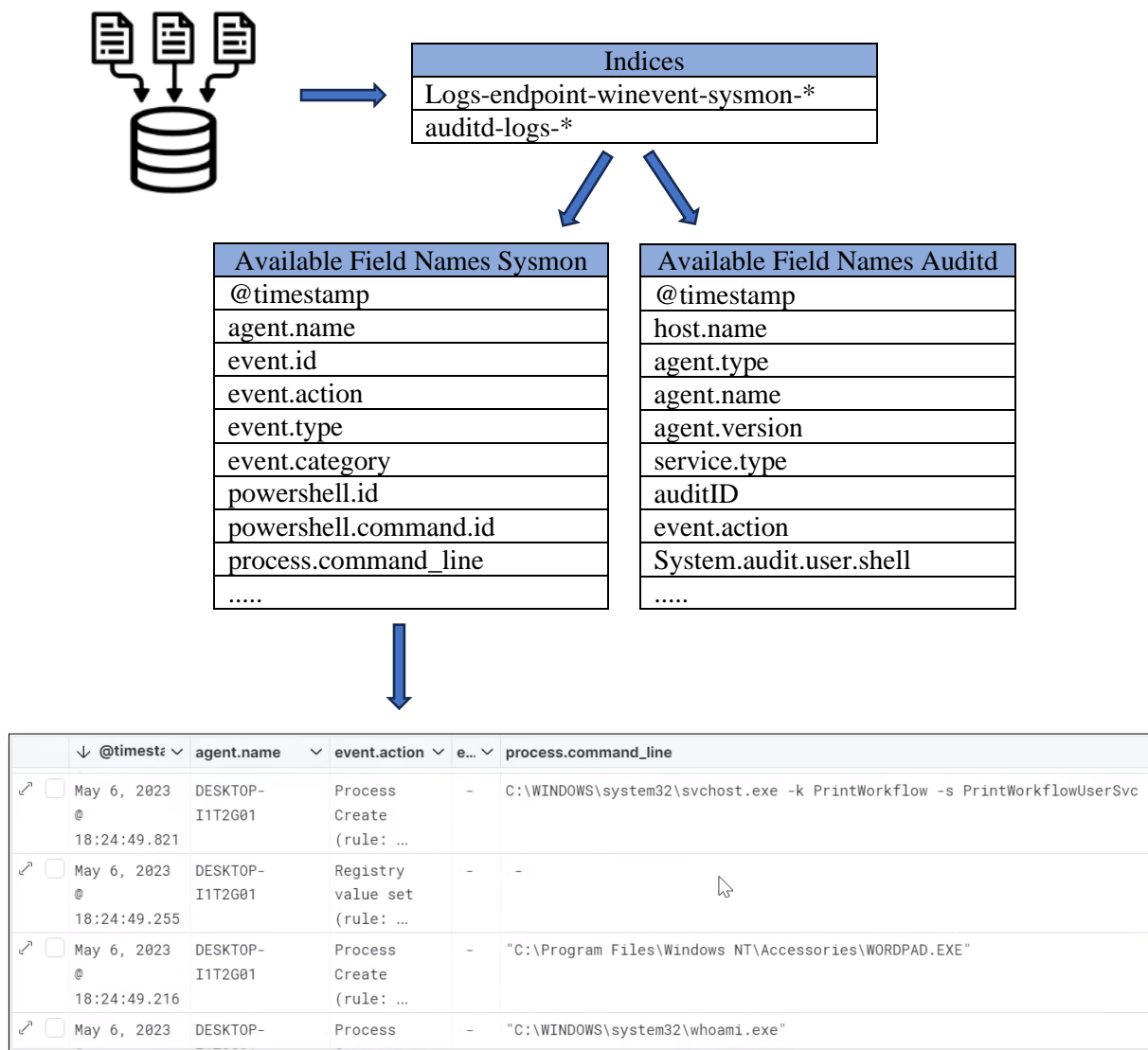
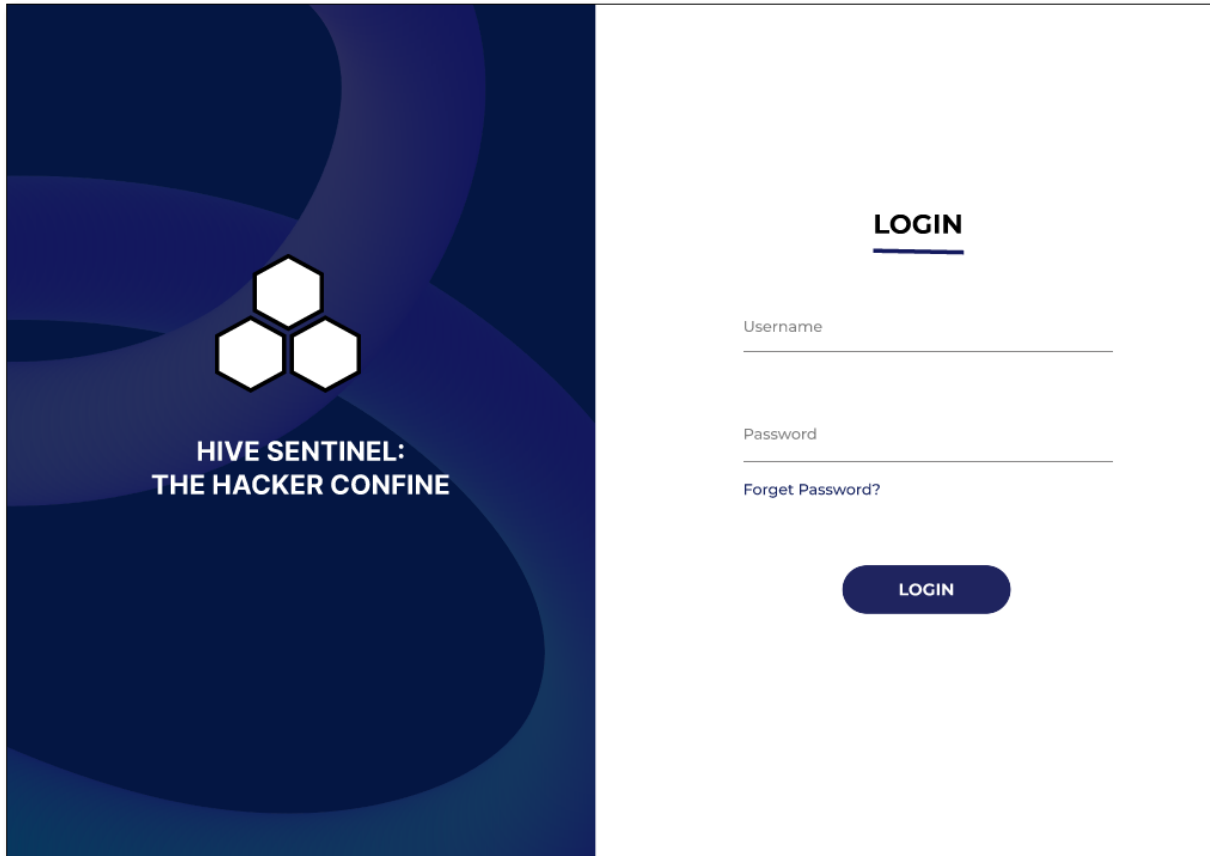


Figure 4.8: Database design

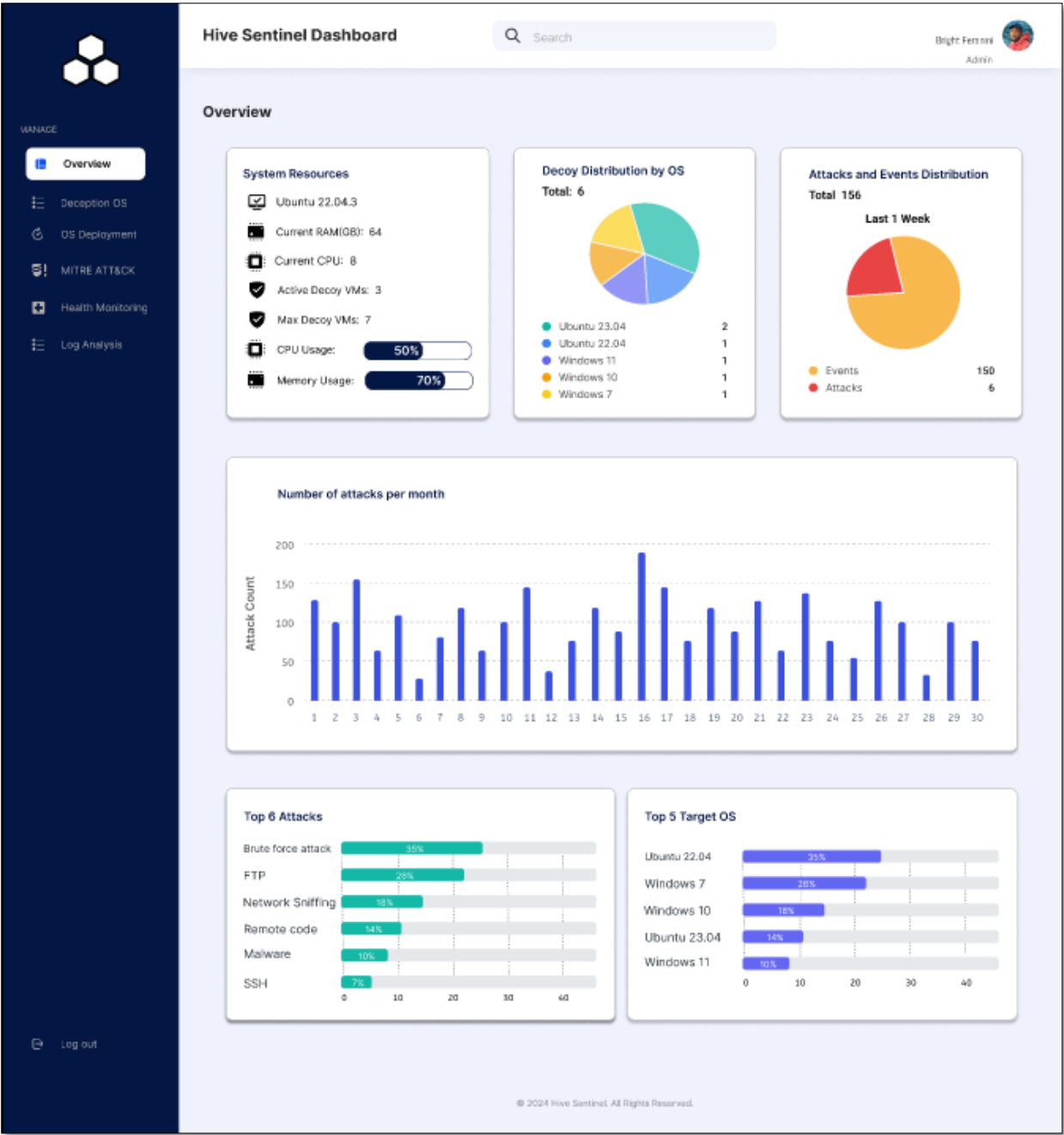
Elasticsearch will create different indices for incoming log files from Sysmon and Auditd from various honeypots. A list of available field names processed by Elasticsearch will be provided. We can then create a table view using the suggested field names based on our specific requirements.

4.6 Interface Design


4.6.1 Login Page



4.6.2 Overview



4.6.3 Deception OS



MANAGE

Overview

Deception OS

OS Deployment

MITRE ATT&CK

Health Monitoring

Log Analysis

Log out

Hive Sentinel Dashboard

Search


Bright Foranmi Admin

Deception OS

#	MACHINE NAME	STATUS	IP ADDRESS	OS TYPE	LURES	SERVICES
1	honeyp0t-1	Initialized	10.4.100.6	Ubuntu 23.10	SMB, FTP, MySQL	RDP
2	honeyp0t-2	Initialized	10.4.100.9	Ubuntu 22.4	SMB, RDP, MySQL	SMB
3	Machine-1	Initialized	10.4.100.7	Windows 11		RDP
4	honeyp0t-3	Initialized	10.4.100.10	Windows 10	SMB, MySQL	MySQL
5	honeyp0t-4	Initialized	10.4.100.11	Ubuntu 23.10	SMB, FTP, MySQL	RDP
6	honeyp0t-5	Initialized	10.4.100.13	Windows 7	SMB, RDP, MySQL	SMB

© 2024 Hive Sentinel. All Rights Reserved.

4.6.4 OS Deployment



MANAGE

Overview

Deception OS

OS Deployment

MITRE ATT&CK

Health Monitoring

Log Analysis

Log out

Hive Sentinel Dashboard

Search

Bright Foranmi Admin

OS Deployment

Name*

new profile

Please enter machine name.

OS Type*

Ubuntu 23.10

Password*

Enter Password

Lures*

FTP,SMB


Selected Services*

MySQL

Create Machines

Clear

4.6.5 MITTRE ATT&CK



MANAGE

Overview

Deception OS

OS Deployment

MITRE ATT&CK

Health Monitoring

Log Analysis

Log out

Hive Sentinel Dashboard

Search

Bright Feramli Admin

MITRE ATT&CK

2024-03-15 09:30:00

Attack Name

Deceptive Credential Usage

Severity

Very High

MITRE ATT&CK

T0812 T0859

Service

System

2024-03-15 09:30:00

Attack Name

Remote Code Execution

Severity

Very High

MITRE ATT&CK

T1193 T1566

Service

System

2024-03-15 09:30:00

Attack Name

Reconnaissance Scanning

Severity

Low

MITRE ATT&CK

T1046 T1133

Service

Web

2024-03-15 09:30:00

Attack Name

Network Sniffing

Severity

Moderate

MITRE ATT&CK

T1040 T1047

Service

Network

2024-03-15 09:30:00

Attack Name

Deceptive Credential Usage

Severity

Very High

MITRE ATT&CK

T0812 T0859

Service

System

2024-03-15 09:30:00

Attack Name

Brute Force Password Guessing

Severity

Moderate

MITRE ATT&CK

T1110 T1111

Service

MySQL

2024-03-15 09:30:00

Attack Name

Adware Installation

Severity

Low

MITRE ATT&CK


T1170 T1125

Service

System

© 2024 Hive Sentinel. All Rights Reserved.

4.6.6 Health Monitoring



MANAGE

Overview

Deception OS

OS Deployment

MITRE ATT&CK

Health Monitoring

Log Analysis

Log out

Hive Sentinel Dashboard

Search

Bright Feramli Admin

Health Monitoring

honeyp0t-1

Ubuntu 23.10

RAM(GB): 4

CPU Usage: 10%

Memory Usage: 60%

Revert Shutdown

honeyp0t-2

Ubuntu 22.10

RAM(GB): 4

CPU Usage: 90%

Memory Usage: 70%

Revert Shutdown

honeyp0t-3

Windows 11

RAM(GB): 4

CPU Usage: 30%

Memory Usage: 50%

Revert Shutdown

honeyp0t-4

Windows 11

RAM(GB): 4


CPU Usage: 40%

Memory Usage: 60%

Revert Shutdown

© 2024 Hive Sentinel. All Rights Reserved.

4.6.7 Log Analysis



MANAGE

Overview

Deception OS

OS Deployment

MITRE ATT&CK

Health Monitoring

Log Analysis

Log out

Hive Sentinel Dashboard

Search

Bright Ferammi Admin

Analytics

Event ID: 1

Date and Time: 2024-03-15 09:30:00

Process Name: C:\Windows\System32\svchost.exe

Parent Process: C:\Windows\System32\services.exe

User: NT AUTHORITY\SYSTEM

Event ID: 2

Date and Time: 2024-03-15 12:30:00

Target File: C:\Program Files\Example\important_file.txt

New Creation Time: 2024-03-15 12:30:00

Event ID: 3

Date and Time: 2024-03-15 10:15:00

Source IP: 192.168.1.100

Source Port: 53234

Destination IP: 8.8.8.8

Destination Port: 53

Protocol: UDP

Event ID: 4

Date and Time: 2024-03-15 14:10:00

Pipe Name: \\pipe\MyNamedPipe

Process ID: 1234

4.7 References

- [1] Team Acalvio. (2017). DECEPTION 1.0: HONEYPOTS ARE DEAD! In *Acalvio Deception Cyberdefense Manual* (pp. 8-15). Acalvio Technologies.
- [2] Team Acalvio. (2017). DECEPTION 2.0: LONG LIVE HONEYPOTS! In *Acalvio Deception Cyberdefense Manual* (pp. 24-33). Acalvio Technologies.