Université d’Ottawa
Faculté de génie

University of Ottawa
Faculty of Engineering

uOttawa

L’Université canadienne
Canada’s university

ELG5901 Electrical Engineering Project

Final Report

Student name_ Salwa Youssef Attia Attia ________________ ID_ 300389878 _____

Student name_ Fatma Mohamed Basal ________________ ID_ 300389394 _____

Student name_ Mayar Mohamed Abdellatif ________________ ID_ 300389363 _____

Student name_ Manar Abdallah Ibrahim ________________ ID_ 300389404 _____

Graduate Program

Semester to Register _ Fall, 2023 ____________

Project Title _ Design Deception Solutions ____________________________

Table of Contents:

Table of Figures:

Table of Tables:

Acronyms

KVM: Kernel-based Virtual Machine.

SDN: Software-Defined Networking.

IP: Internet Protocol.

HTTP: Hypertext Transfer Protocol.

SSH: Secure Shell.

TCP: Transmission Control Protocol.

OVS: Open vSwitch.

SIEM: Security Information and Event Management.

NIDS: Network Intrusion Detection System.

IDS: Intrusion Detection System.

WAF: web application firewall.

ELK: Elasticsearch, Logstash, and Kibana.

DVWA: Damn Vulnerable Web Application.

OWASP: Open Worldwide Application Security Project.

1. Introduction

1.1 Problem Definition

Traditional cybersecurity measures often fail to detect and respond to attacks effectively. Attackers exploit vulnerabilities in real assets, evading detection and causing harm before being noticed. Manual decoy setup and monitoring are resource intensive and lack real-time insights. As a result, organizations lack a proactive means to swiftly identify attackers, assess their tactics, and protect their assets. Organizations require a comprehensive and automated solution that not only creates realistic decoys but also tracks interactions with them in real-time. This solution should immediately alert security teams when attackers engage with these decoys, enabling identification of potential threats and the gathering information about attackers' tactics and IP addresses. The project's general approach involves deploying automated decoys and honeypots that mimic real assets. The project aims to integrate SDN to dynamically alter network configurations, making the environment more challenging for attackers. Throughout the project, the team gained hands-on experience in developing deception solutions, deploying honeypots, and utilizing SDN for cybersecurity enhancement. The collaboration with industry experts provide insights into emerging threats, ethical considerations, and strategic planning for long-term cybersecurity resilience.

1.2 Background

This project is influenced by studying academic papers and industry resources. The academic Paper "Enhancing Cyber Defense with Autonomous Agents Managing Dynamic Cyber Deception" by Chiang et al. highlights the pivotal role of autonomous agents in orchestrating dynamic deception strategies. This paper underscores the significance of proactive engagement and dynamic adaptation in thwarting cyber threats. Another academic paper, "Automatic Honeypot Generation and Network Deception" by Stephen Hudak Jr., offers insights into the practical implementation of deception techniques. This research delves into automatic honeypot creation and network deception, laying the foundation for the project's proactive defense approach. In the industry domain, resources like the “MITRE ATT&CK Based Evaluation on In-Network Deception Technology for Modernized Electrical Substation Systems” and “Implementer's Guide to Deception Technologies” from SANS Institute provide tangible frameworks and guidelines for implementing effective cybersecurity measures. These references emphasize the importance of threat intelligence and adaptive strategies. To execute the project, tools like "Artillery" honeypot and "Open vSwitch" in SDN will be explored. Additionally, KVM will be used for automatic generation of honeypots to simulate diverse computer types and configurations. Also, will use ModSecurity WAF to create rules to monitor the traffic and ELK stack to visualize and analyze the logs to make decisions This technology enriches the project's realism and practicality.

1.3 Project Context

This project aims to enhance cybersecurity measures via the deployment of automated decoys and honeypots that mimic real assets, coupled with software-defined networking (SDN) capabilities for dynamic network reconfigurations. For the successful completion of our project, we collaborate with EG|CERT Team, Dr. Ahmed Hamdy As our Egypt Mentor and Prof. Ali Abbas As the uOttawa Supporter. also, we interact with various external systems, utilize third-party tools and APIs such as:

- **Network Infrastructure:** We need to interact with the network infrastructure to deploy and monitor the honeypots and SDN-based cyber deception solution. This includes the routers, switches, firewalls, and servers that hosts our decoys and dynamically alter network configurations.
- **Artillery Honeypot:** This is a powerful tool for simulating vulnerable services commonly found in real-world environments. It provides the means to engage potential attackers and gather crucial insights.
- **Open vSwitch:** This open-source software-defined networking (SDN) solution is essential for implementing dynamic network configuration changes.
- **KVM:** Used for the automatic generation of honeypots to simulate diverse computer types and configurations.
- **Intrusion Detection System (IDS):** The IDS is integrated with the honeypots and central database for data storage. The IDS trigger real-time alerts upon detecting suspicious activity, providing relevant information such as attacker IP addresses and interaction logs.
- **ModSecurity WAF:** It provides security for web applications by monitoring and blocking malicious activities. It also integrates with web servers to inspect and filter HTTP traffic, and we used it to create rules and policies that define how web requests and responses should be handled. These rules can be customized to detect and block specific patterns associated with known attacks or suspicious behavior.
- **ELK Stack:** ELK is used for log analysis and monitoring, it is also used for business intelligence, security information and event management (SIEM), and any scenario where the ability to analyze and visualize large volumes of data is essential.

2. Design Overview

The proposed centralized control system aims to enhance cybersecurity posture by monitoring and logging network activities. The system seamlessly integrated with a SIEM solution, the ELK (Elasticsearch, Logstash, Kibana) stack, for comprehensive analysis of logged data. Additionally, the system is configured to detect malicious activities and, based on alerts generated by the SIEM, redirect suspicious traffic to a honeypot, thus minimizing potential risks to the main server.

2.0.1 Key Components and Architectural View

- **Log Collection and Forwarding:** Agents deployed on network servers to forward logs to the centralized monitoring system.
- **Centralized Monitoring System:** The system aggregates logs from various sources to provide a comprehensive overview of network activities. Additionally, it normalizes these logs for consistency and ease of analysis, enhancing the efficiency of monitoring and facilitating meaningful insights into the network's functioning.
- **SIEM Integration:** Direct communication between the Centralized Monitoring System and the ELK stack for log analysis and visualization.
- **Network Monitoring:** NIDS sensors deployed throughout the network enable real-time monitoring of network traffic, actively scanning for potential attacks.
- **Web Application Security:** implemented to safeguard web applications, with regularly updated rules aimed at detecting and preventing common web-based attacks ensuring that the web applications are shielded from evolving threats against potential vulnerabilities and unauthorized access attempts.
- **Alert Generation:** Real-time alerts trigger automated responses.
- **Automated Traffic Redirection:** Integration with network devices for redirecting suspicious traffic to the honeypot.
- **Honeypot Infrastructure:** Emulates various services to attract and engage potential attackers.

The centralized control system acts as the nerve center for monitoring network activities, collecting logs, and integrating with a SIEM solution for advanced analysis. Upon detection of suspicious activities, the system triggers automated responses, redirecting the traffic towards the honeypot. This proactive approach allows for the containment of potential threats and provides valuable insights into evolving attack vectors.

2.0.2 Technical Specifications

1. **Log Collection and Forwarding:**
   - Log Agents: Deployed on network devices and servers and support for common log formats such as Syslog.
   - Centralized Log Server: Utilizes Logstash for log processing and normalization and Elasticsearch for efficient indexing and storage.
2. **SIEM Integration (ELK Stack):**
   - Elasticsearch for indexing and storage.
   - Logstash for log parsing and enrichment.
   - Kibana for real-time visualization and analysis.
3. **Automated Response:**
   - Alerting Rules**:** Configurable rules based on known threat indicators and abnormal behavior.
   - Traffic Redirection: Integration with SDN Controller for automated traffic redirection to the honeypot.

2.0.3 Architectural View Diagram

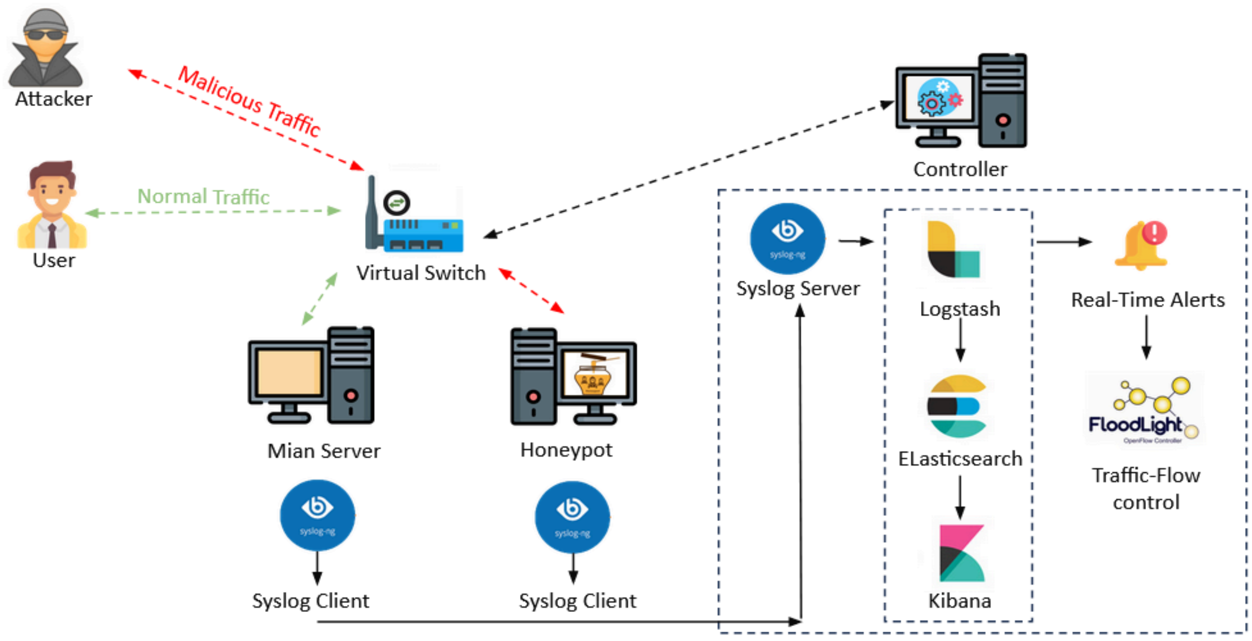


Figure 1 : Architecture View Diagram

This design envisions a robust, centralized control system that seamlessly integrates with SIEM technology, providing organizations with enhanced visibility into network activities and a proactive approach to mitigating potential threats through traffic redirection to the honeypot.

2.1 Requirements

The requirements for the centralized network monitoring and deception system have been carefully crafted to address the needs and expectations of stakeholders. These requirements are structured to ensure that the end users can effectively utilize the system to solve their engineering problems related to network security.

- **Network Monitoring:** The system shall capture and analyze network traffic in real time.
- **Logging and SIEM Integration:** The system shall log network activities, generating detailed logs for analysis. In addition, the system shall integrate seamlessly with the SIEM solution for log parsing, storage, and retrieval.
- **Real-Time Alerting Engine:** An alerting engine shall be implemented to analyze SIEM alerts.
- **User Interface:** A user-friendly interface shall be provided for system administrators to monitor network activities, configure settings, and view alerts.
- **Performance:** The system shall handle network traffic monitoring without significant impact on overall network performance. Also, alerts and redirections should occur in near real-time to ensure timely responses.
- **Scalability and Reliability:** The architecture shall be scalable to accommodate the growth of network traffic and the addition of new network segments.

2.2 Detailed Design

The detailed design of the network monitoring and deception system involves the decomposition of the overall system into key subsystems and components. These components interact seamlessly to achieve the project's objectives:

2.2.1 System Infrastructure

KVM Virtualization Environment:

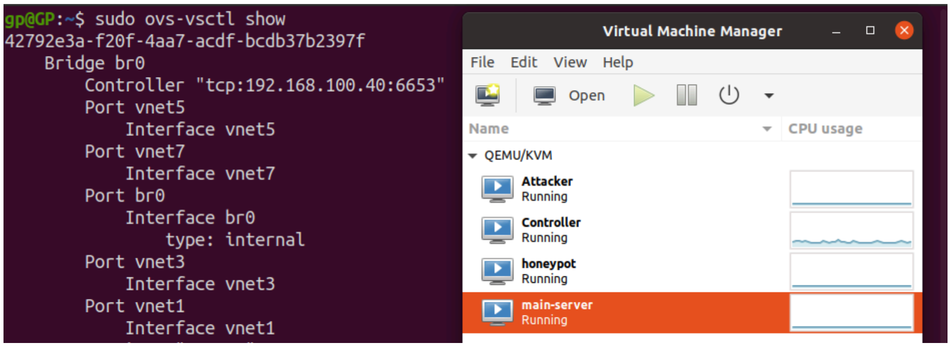
This system is utilized for the creation of virtual machines, encompassing essential components such as:

- **The main server**
    - o Implementing a web server for hosting web applications intended for testing and analysis purposes. It is configured to provide a controlled environment, ensuring a secure setting for conducting comprehensive security assessments on hosted web applications.
    - o SSH Server is implemented to enable secure remote access ensuring that the system.
- **The Honeypot**
    - o Deploying Artillery Honeypot simulating vulnerable services and listening on ports that are used in attacks (22 for SSH attacks and 80 for HTTP attacks).

- **The Centralized Controller**
  - o The centralized controller is responsible for orchestrating and managing the overall system. Within the KVM environment, this virtual machine oversees the deployment of decoys, adjusts network flow rules, and interfaces with SIEM components.
- **The Attacker**
  - o The attacker virtual machine represents a simulated adversary within the system. It interacts with the decoys and mimics potential threat behaviors, contributing to the testing and validation of the overall security infrastructure.

Open vSwitch (OVS):

Open vSwitch Enables the creation of a virtual network to interconnect the virtual machines facilitating flexible traffic control and routing within the virtualized environment. We have deployed a virtual switch “br0” attached to our machines to enhance the overall control and adaptability of the system, allowing for efficient data flow and communication patterns using “OpenFlow Protocol” optimizing the overall performance and connectivity of the interconnected virtual machines.



```
gp@GP:~$ sudo ovs-vsctl show
42792e3a-f20f-4aa7-acdf-bcdb37b2397f
    Bridge br0
        Controller "tcp:192.168.100.40:6653"
        Port vnet5
            Interface vnet5
        Port vnet7
            Interface vnet7
        Port br0
            Interface br0
                type: internal
        Port vnet3
            Interface vnet3
        Port vnet1
            Interface vnet1
```

Attacker
Running

Controller
Running

honeypot
Running

main-server
Running

Figure 2 : OVS and VMs

2.2.2 The Main-server

The main server is a crucial component within the network monitoring and deception system. It hosts various services and vulnerable websites designed for testing and analysis purposes.

Deployed Services:

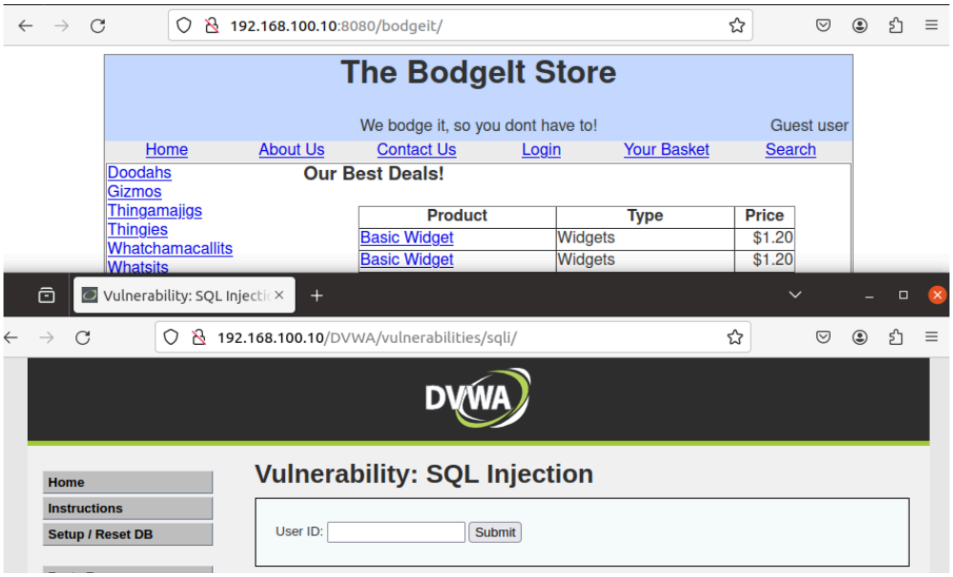
- **Web Server**

The web server on the main server hosts two intentionally vulnerable websites, providing a controlled environment for security testing:

- o Bodgeit: A deliberately insecure web application for practicing penetration testing and enhancing security skills.
- o DVWA (Damn Vulnerable Web Application): An intentionally vulnerable web application designed for security testing and training.

- **SSH Server**

The SSH server is deployed to facilitate secure remote access for system administration and monitoring, hardening it against brute forcing and password attacks.



Product	Type	Price
Basic Widget	Widgets	$1.20
Basic Widget	Widgets	$1.20

Figure 3 : The Deployed websites

2.2.3 Network Security Components

NIDS (Network Intrusion Detection System):

The system employs Snort for monitoring network traffic by mirroring the network traffic to the controller machine. Positioned strategically within the virtual network, Snort analyzes and detects potential security threats effectively. To fortify the network against various threats, the system configures rules specifically targeting:

- Denial of Service (DoS) attacks
- SSH brute forcing attempts.
- unauthorized access to databases and configurations.

This proactive approach to rule configuration enhances the system's ability to identify and respond to specific security risks, contributing to a robust defense mechanism against a range of potential cyber threats within the virtualized environment.

WAF (Web Application Firewall):

ModSecurity WAF is installed on the main server to safeguard web applications. Functioning as a critical security layer, the WAF actively monitors and filters HTTP traffic within the virtualized environment. Specifically, it is configured to secure against OWASP (Open Web Application Security Project) Top 10 vulnerabilities, offering protection against a range of common web application security risks.

Syslog-ng:

The logging system is configured for centralized logging, gathering logs from various sources including the main server, Snort, and ModSecurity.

The data is transmitted on port 514 using the TCP protocol, ensuring secure and reliable communication. Additionally, to enhance fault tolerance, the logging system maintains a local copy of the logs in the event of a server failure. This dual approach of centralization for efficient log management and local backup for resilience contributes to a robust and reliable logging infrastructure, ensuring the preservation and accessibility of critical log data even under adverse conditions.

It also plays a vital role in aggregating and forwarding logs to the ELK stack for centralized storage and analysis. Syslog-ng enhances the system's visibility into network activities, providing valuable data for threat detection and incident response.

ELK Stack:

The ELK stack, composed of Elasticsearch, Logstash, and Kibana, serves as the SIEM solution for the network monitoring and deception system.

- **Logstash:**
    - Processes and enriches raw log data, ensuring compatibility and normalization before forwarding it to Elasticsearch.
- **Elasticsearch:**
    - Stores and indexes logs received from Syslog-ng, creating a centralized repository for log data.
- **Kibana:**
    - Provides a user-friendly interface for real-time visualization and analysis of log data creating insights and alerts based on the collected logs.

Floodlight SDN Controller:

The Floodlight SDN controller is a crucial element in the system, managing the virtual network created by Open vSwitch. It controls traffic flow based on alerts generated by the SIEM (ELK stack) and dynamically adjusts network flow rules. This ensures the redirection of suspicious traffic to the honeypot using a python script and Floodlight REST API, optimizing the system's response to potential security incidents.

2.2.4 Dynamic Flow Rule Adjustment Script

Python Script for SDN Flow Rules:

A Python script is developed to dynamically adjust flow rules within the Floodlight SDN controller. The script is designed to interact with the Floodlight SDN controller's REST API, enabling responding to alerts and updating the network flow rules in real-time allowing for immediate changes in the network's behavior. Key components and functionalities of the script include:

- **StaticFlowPusher Class:**
  - Represents the functionality for interacting with the Floodlight controller's REST API and provides methods for getting, setting, and removing flow rules.

- **REST API Interaction:**
  - Uses the “httplib” library to establish HTTP connections with the Floodlight controller.
  - Implements RESTful calls (GET, POST, DELETE) to retrieve existing flow rules, add new rules, or remove existing rules.

- **Dynamic Flow Rule Definition:**
  - Contains parameters such as switch identifier, rule name, priority, input port, and actions.
  - Defines flow rules dynamically based on information extracted from alerts assigning them a higher priority than the regular flow rules.

2.2.5 Interaction Between Subsystems and APIs

- **Log Flow:**
  - Syslog-ng collects logs from the main server, NIDS (Snort), and WAF (ModSecurity) through standard syslog protocols.
  - These logs are forwarded to the ELK stack for centralized storage and analysis.
- **SIEM Integration:**
  - ELK stack processes and correlates logs to generate real-time alerts.
  - Floodlight SDN controller interacts with ELK to receive alerts for traffic flow adjustments.
- **Traffic Redirection:**
  - Floodlight, based on ELK alerts, dynamically adjusts flow rules to redirect suspicious traffic to the honeypot.
- **SDN Controller and Virtual Network:**
  - Floodlight controls the virtual network created by Open vSwitch Adjusting traffic flow rules to isolate and analyze potentially malicious traffic.
- **Dynamic Flow Rule Adjustment:**
  - Python script extracts malicious IPs from ELK alerts.
  - Uses Floodlight's REST API to push updated flow rules for immediate network adaptation.

2.2.6 Trade-Off Studies and Evaluation of Alternatives

In the design phase, critical decisions were made regarding the selection of key components, and trade-off studies were conducted to evaluate alternatives for virtualization platforms, SDN controllers, and SIEM solutions. These decisions were driven by considerations of performance, scalability, compatibility, and the overall suitability of each option for the network monitoring and deception system.

The following assessments detail the rationale behind the chosen alternatives in each category:

Virtualization Platform:

	KVM	VMware	VirtualBox
Performance	provides better control over resource utilization and allocation.	slower performance when running servers	has limitations in resource management, making it less suitable for resource-intensive applications.
Integration	primarily used in Linux environments and is well-integrated with the Linux kernel.	Requires additional conversion utility for more VM types; VMware VSphere and Cloud Air.	offers compatibility with various operating systems, including Windows, macOS, and Linux
Cost	Free Open source	require a paid license	Free, under the GNU General Public License

Table 1 : Virtualization Platform

For the project's specific requirements, we chose KVM because of its unparalleled performance, seamless integration with Linux environments, and cost-effectiveness as an open-source solution.

Honeypot Selection:

	Artillery	Honeyd	Dionaea
Emulation of Vulnerabilities	Functions as a honeypot, monitoring tool, and alerting system.	Creates virtual hosts with configurable arbitrary services and TCP personalities	Can emulate vulnerabilities in services like HTTP, MySQL, and SMB
Configurability	Adapts to evolving security needs, including hardening monitoring and detection of insecure configurations.	Highly configurable through a simple configuration file, enabling simulation of various services.	Configuration may involve specifying services and adapting TCP personalities.

Table 2 : Honeypot Selection

The decision to choose Artillery for the project is based on its versatility in emulating vulnerabilities and its role as a honeypot, monitoring tool, and alerting system. While Artillery excels in alerting when an attacker connects, it's important to note that it doesn't facilitate direct interactions with the attacker. The focus on alerting aligns with the project's emphasis on proactive threat detection and response, contributing to a robust network monitoring and deception system.

NIDS and WAF Selection:

	Snort	Suricata
Performance and Efficiency	Snort is effective on lower-speed networks but may be less scalable on high-speed networks.	High throughput and scalability but resource-intensive
Integration	Flexible deployment, integration with other security solutions	Flexible deployment, integration with other security solutions
Features and Capabilities	Widely adopted, comprehensive rule coverage, active community support	limited community ruleset compared to Snort.

Table 3: NIDS and WAF Selection

The decision to choose Snort as the preferred Intrusion Detection System (IDS) aligns with the project's specific needs, emphasizing effective performance across varying network speeds, flexible deployment, and a comprehensive ruleset with active community support. While Suricata may be suitable for larger enterprises with demands for high throughput and scalability, Snort meets the current requirements of our project effectively. The focus on Snort ensures an optimal balance between performance and the project's immediate needs without compromising on security capabilities.

SIEM Solution Selection:

	ELK	Splunk
Performance	Elasticsearch is known for speed and scalability in searching and retrieving data. Efficient for full-text search and complex queries.	Known for powerful search capabilities, especially with complex queries and large datasets.
Scalability	Designed for horizontal scaling. Elasticsearch clusters can easily scale by adding more nodes to handle indexing and search workload.	Scaling horizontally can be challenging, but clustering options are available to distribute the workload.

Index Management Overhead	Requires attention to optimize index management for performance; Inefficient management may impact system efficiency.	Splunk handles index management transparently; Less manual intervention required.
Data Ingestion	Logstash is used for shipping data; Beats (Filebeat, Heartbeat, Packetbeat, etc.) are lightweight agents for specific data types.	Supports various formats (XML, JSON, CSV, etc.) and provides multiple ingestion options including forwarders and streaming connectors.
Data Visualization	Kibana is used for data visualization with pre-built visualizations; Focuses on data discovery and exploration.	User-friendly interface with pre-built dashboards; Focuses on search-based analytics.

Table 4 : SIEM Solution Selection

We chose ELK (Elasticsearch, Logstash, Kibana) for our project, prioritizing its proven speed, scalability, and efficient full-text search capabilities, despite potential index management overhead.

2.3 Implementation

In our project, we've implemented a comprehensive security infrastructure that leverages Open vSwitch within the SDN framework. The primary objective is to enhance the security of virtual network devices by proactively addressing various cyber threats. Key components of our security framework include:

1. **ModSecurity Integration:**
   - We've integrated ModSecurity, a powerful Web Application Firewall (WAF) tool.
   - Proficient in countering OWASP Top 10 attacks, such as SQL injection and various cross-site scripting types.
   - Effectively safeguards against cross-site request forgery, authentication bypass, and command injection.
2. **Snort Rules for Defense:**
   - Snort rules have been deployed to fortify defenses against SSH and Denial of Service (DoS) attacks.
3. **Dynamic Threat Response:**
   - The SDN controller dynamically identifies an attacker's IP address when alerted by security tools.
   - Traffic is then rerouted to a honeypot for in-depth analysis.

4. **Centralized Alert Management:**
   - All generated alerts from across the system are centralized and sent to the ELK stack (Elasticsearch, Logstash, Kibana).
   - Enables comprehensive monitoring and analysis, empowering quick responses to emerging threats.

Our implemented system includes an automated logging infrastructure aimed at improving incident response and providing users with comprehensive insights:

- **Automated Log Transmission:**
  - In the event of a cyber attack, logs are automatically transmitted to the centralized controller.
  - Facilitates quick and efficient access to relevant log data for prompt analysis and response.

The centralized controller serves as a command center, offering users the capability to access and control the entire system:

- **User-Friendly Interface:**
  - Through an intuitive interface, users can navigate seamlessly through logs.
  - Real-time visibility into network events and potential security threats is provided.

This centralized control mechanism not only streamlines the monitoring process but also offers a holistic view of the system's status, enhancing the overall security resilience of our virtual network environment.

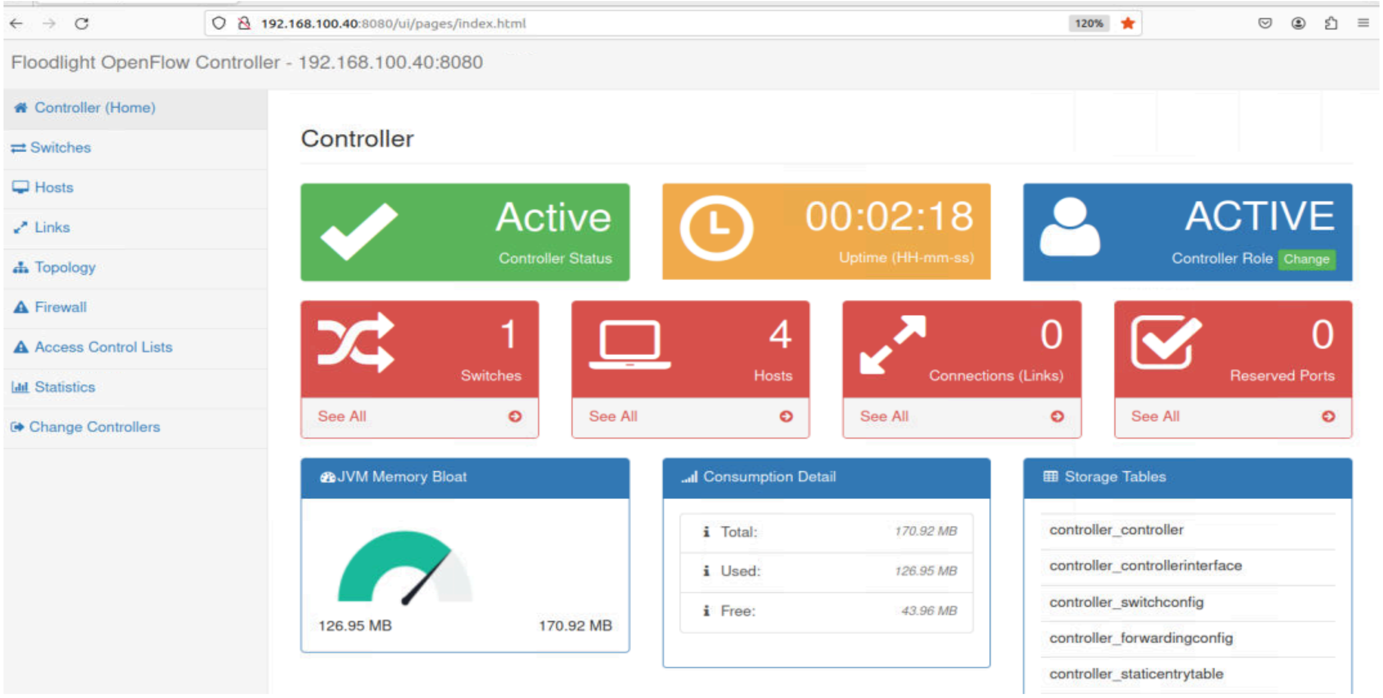
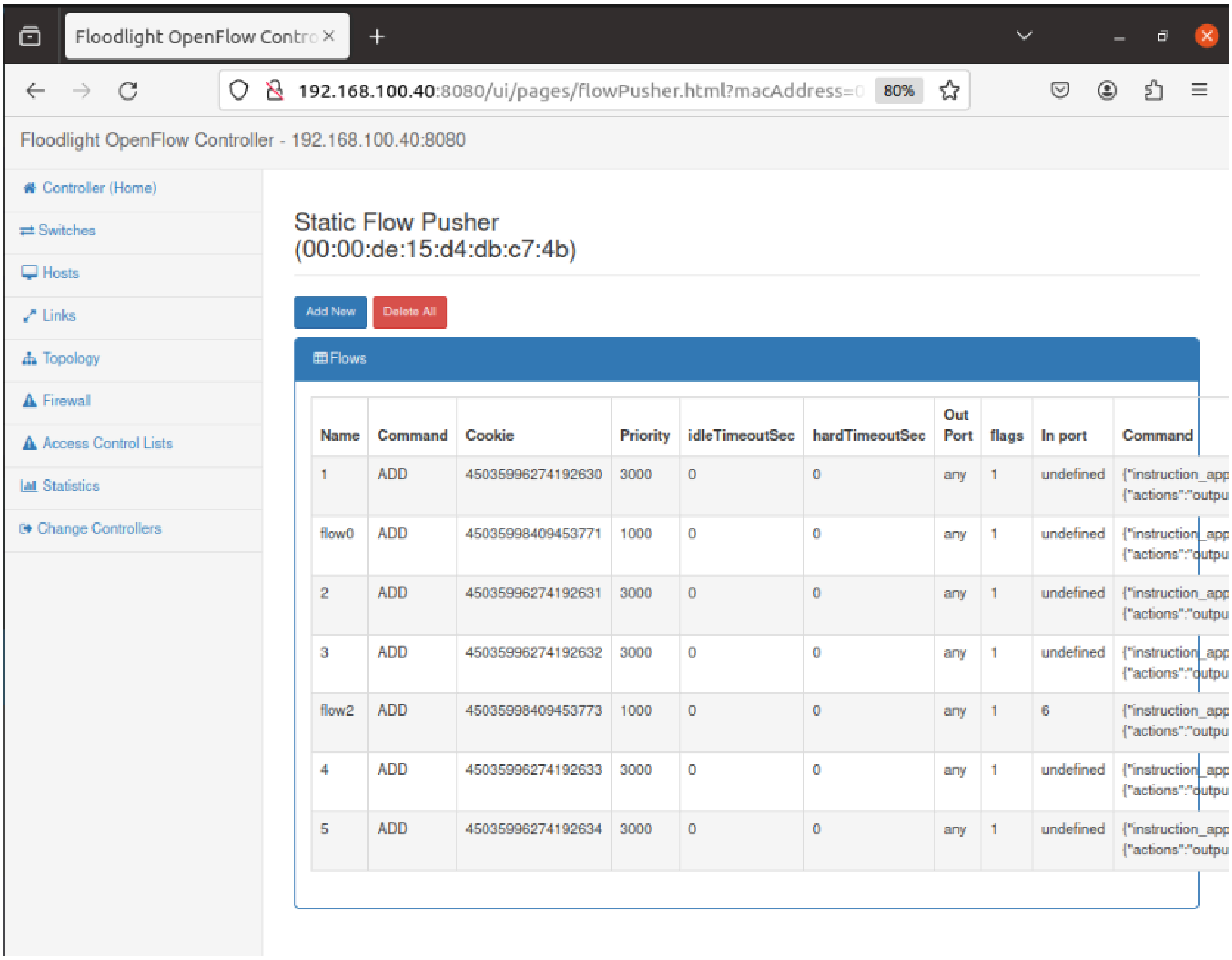


Figure 4 : Controller


Floodlight OpenFlow Contro
192.168.100.40:8080/ui/pages/FlowPusher.html?macAddress=0
80%
Floodlight OpenFlow Controller - 192.168.100.40:8080
Controller (Home)
Switches
Hosts
Links
Topology
Firewall
Access Control Lists
Statistics
Change Controllers
Add New
Delete All
Flows

Static Flow Pusher
(00:00:de:15:d4:db:c7:4b)

Name	Command	Cookie	Priority	idleTimeoutSec	hardTimeoutSec	Out Port	flags	In port	Command
1	ADD	45035996274192630	3000	0	0	any	1	undefined	{"instruction_app {"actions":"outpu
flow0	ADD	45035998409453771	1000	0	0	any	1	undefined	{"instruction_app {"actions":"outpu
2	ADD	45035996274192631	3000	0	0	any	1	undefined	{"instruction_app {"actions":"outpu
3	ADD	45035996274192632	3000	0	0	any	1	undefined	{"instruction_app {"actions":"outpu
flow2	ADD	45035998409453773	1000	0	0	any	1	6	{"instruction_app {"actions":"outpu
4	ADD	45035996274192633	3000	0	0	any	1	undefined	{"instruction_app {"actions":"outpu
5	ADD	45035996274192634	3000	0	0	any	1	undefined	{"instruction_app {"actions":"outpu

Figure 5 : Floodlight OpenFlow

In addition to the robust security measures, users wield the authority to dynamically manage and handle rules within our system. This empowerment encompasses the ability to:

- Adjust Network Flow Rules
- Configure Alerting Parameters
- Implement Changes to System Response Mechanisms

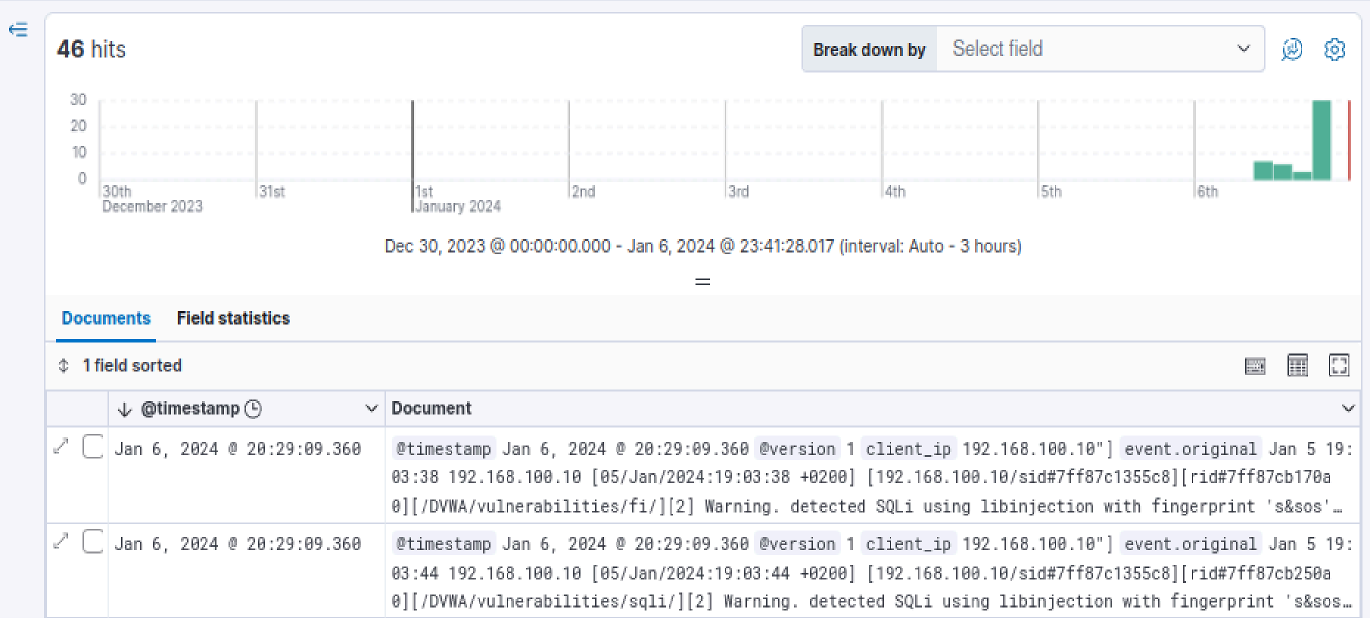
The user-friendly interface of the centralized controller facilitates efficient execution of these actions, empowering users to adapt the system swiftly in response to evolving cyber threats.

A noteworthy feature of our system is its seamless integration with the ELK Stack (Elasticsearch, Logstash, Kibana), providing a powerful toolset for log analysis and visualization. This integration offers users the following capabilities:

1. **Kibana Interface for Real-time Exploration:**
    - Users can access logs through Kibana, providing a user-friendly interface.
    - Enables real-time exploration, analysis, and visualization of log data.

2. **Enhanced Accessibility:**
    - Integration with ELK Stack improves the accessibility of log information.
    - Users can easily retrieve and review relevant data.
3. **In-depth Analysis and Incident Identification:**
    - Facilitates in-depth analysis, empowering users to identify patterns, anomalies, and potential security incidents.

This integrated approach not only enriches the user experience by simplifying rule management but also provides a powerful toolset for proactive analysis, strengthening the system's resilience against emerging threats.



```
Jan 6, 2024 @ 20:29:09.360   @timestamp Jan 6, 2024 @ 20:29:09.360 @version 1 client_ip 192.168.100.10"] event.original Jan 5 19:
                             03:38 192.168.100.10 [05/Jan/2024:19:03:38 +0200] [192.168.100.10/sid#7ff87c1355c8][rid#7ff87cb170a
                             0][/DVWA/vulnerabilities/fi/][2] Warning. detected SQLi using libinjection with fingerprint 's&sos'…
Jan 6, 2024 @ 20:29:09.360   @timestamp Jan 6, 2024 @ 20:29:09.360 @version 1 client_ip 192.168.100.10"] event.original Jan 5 19:
                             03:44 192.168.100.10 [05/Jan/2024:19:03:44 +0200] [192.168.100.10/sid#7ff87c1355c8][rid#7ff87cb250a
                             0][/DVWA/vulnerabilities/sqli/][2] Warning. detected SQLi using libinjection with fingerprint 's&sos…
```

Figure 6 : Kibana Logs sample

Finally, the automated logging system, coupled with centralized control through the controller and log analysis capabilities in the ELK Stack, ensures that users can efficiently manage and respond to cyber threats. This comprehensive approach not only simplifies incident response but also contributes to the overall resilience and security of the network monitoring and deception system.

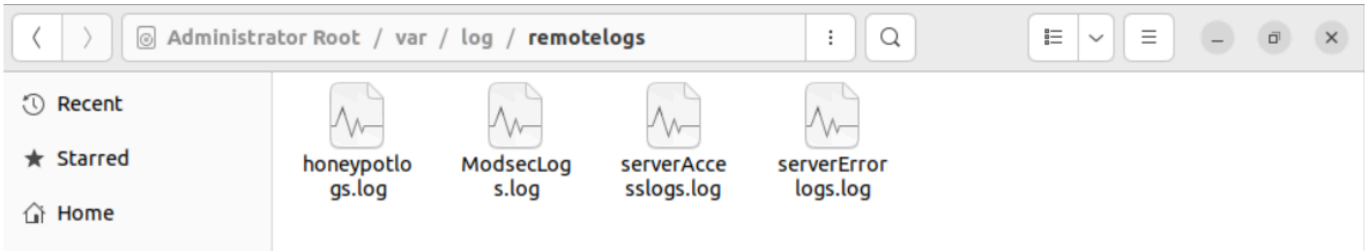


Figure 7 : Remote Logs Folder

2.4 Testing

2.4.1 Data Plan

In our project, the primary data sources are the extensive logs collected from two key components: Snort, serving as the Network Intrusion Detection System (NIDS), and ModSecurity, functioning as the Web Application Firewall (WAF). These logs are essential for documenting network activities, security incidents, and potential cyber threats. Here's a breakdown of the data sources:

- **Snort Logs:**
  - Role as NIDS: Snort acts as the NIDS, monitoring and analyzing network traffic to identify potential security threats. Specific rules are configured to detect activities such as Denial of Service (DoS) attacks, SSH brute-force attempts, and unauthorized access.
  - Log Content: The logs generated by Snort contain details about detected security incidents for ssh attacks and dos attacks , including information on the type of threat, source IP addresses, targeted services, and the specific rules triggered.
  - Testing Significance: During simulated attacks, these logs play a vital role in evaluating the effectiveness of intrusion detection, providing insights into the system's ability to recognize and respond to various network-level threats.
- **ModSecurity Logs:**
  - Function as WAF: ModSecurity operates as the WAF on the main server, actively monitoring and filtering HTTP traffic to secure web applications. It is configured to protect against web-based vulnerabilities, including OWASP Top 10 attacks like SQL injection and cross-site scripting.
  - Log Information: Logs generated by ModSecurity contain details about web application security events, highlighting any suspicious or malicious activities identified during HTTP traffic monitoring.
  - Testing Importance: During simulated web application attacks, ModSecurity logs are crucial for assessing the effectiveness of the WAF in detecting and mitigating threats, providing insights into the system's capacity to safeguard web applications.
- **Additional Data Sources:**
  - Syslog-ng: The syslog-ng logging infrastructure is set up to collect logs from various sources, including the main server, Snort, and ModSecurity. This centralized log collection ensures a comprehensive overview of network activities.
  - ELK Stack (Elasticsearch, Logstash, Kibana): The ELK Stack is integrated for log analysis, storage, and visualization. Elasticsearch stores and indexes logs, Logstash processes and enriches raw log data, and Kibana offers a user-friendly interface for real-time exploration and analysis.

Finally, our project relies on logs from Snort and ModSecurity as primary data sources, capturing insights into network-level and web application security events. These logs are pivotal for testing the system's resilience, evaluating the efficacy of security measures, and providing actionable insights for incident response and analysis. The inclusion of syslog-ng and the ELK Stack enhances centralized log collection, storage, and visualization for comprehensive monitoring and assessment.

2.4.2 Validation & Verification

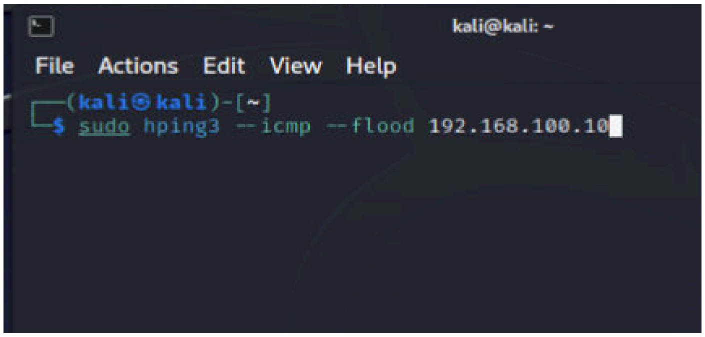
The team executed a series of simulated attacks to gauge the effectiveness of implemented measures, evaluating the system's resilience against a spectrum of cyber threats. This rigorous testing encompassed various aspects, including:

- **Simulated SSH Attacks:**

Emulated real-world SSH brute-force attempts and unauthorized access scenarios, which aimed to validate Snort rules configured for NIDS. Testing focused on the system's ability to identify and mitigate SSH-related security threats.

- **Denial of Service (DoS) Attack Simulations:**

Conducted simulated DoS attacks using hping3 to assess the system's resilience under high traffic and intentional attempts to overwhelm network resources Focused on validating Snort rules designed to detect and respond to DoS attacks. Emphasized the system's capacity to maintain network stability and performance under deliberate disruption attempts.

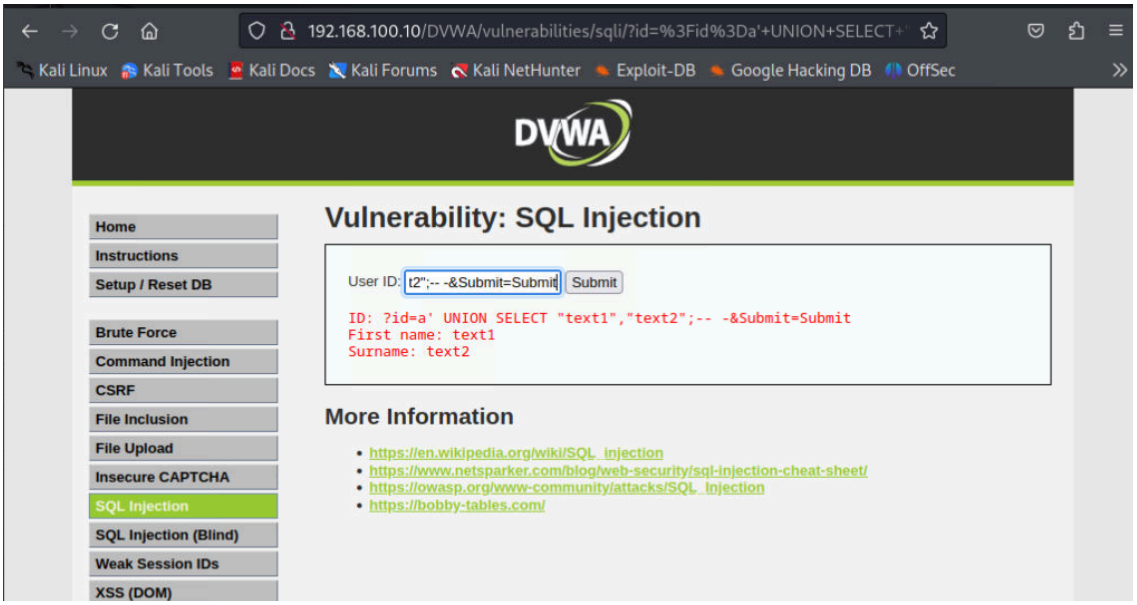


```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 --icmp --flood 192.168.100.10
```

Figure 8 : DoS Attack

- **Web Application Attacks:**

Simulated various web application attacks, including SQL injection and cross-site scripting. Evaluated the ModSecurity WAF and other security measures on the main server. Objective was to confirm WAF's effectiveness in monitoring and filtering HTTP traffic, providing protection against common web-based vulnerabilities. The DVWA website was used just for testing purposes, not for real use.



Vulnerability: SQL Injection

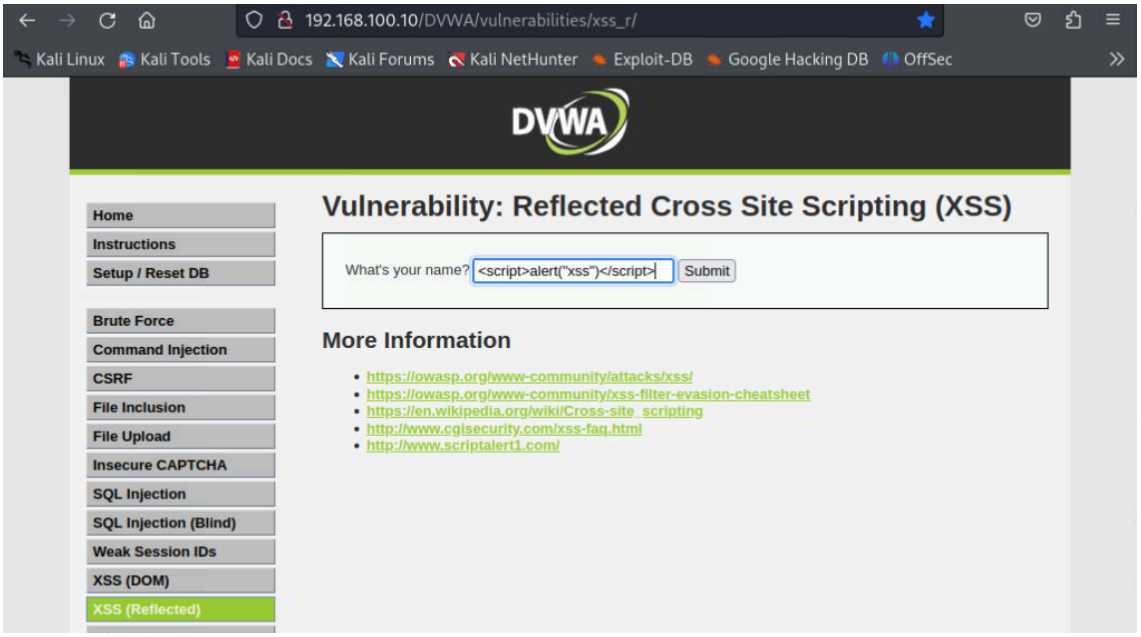
User ID: t2";-- -&Submit=Submit

```
ID: ?id=a' UNION SELECT "text1","text2";-- -&Submit=Submit
First name: text1
Surname: text2
```

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
- https://owasp.org/www-community/attacks/SQL_Injection
- https://bobby-tables.com/

Figure 9 : SQL Injection Attack



Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name? <script>alert("xss")</script>

More Information

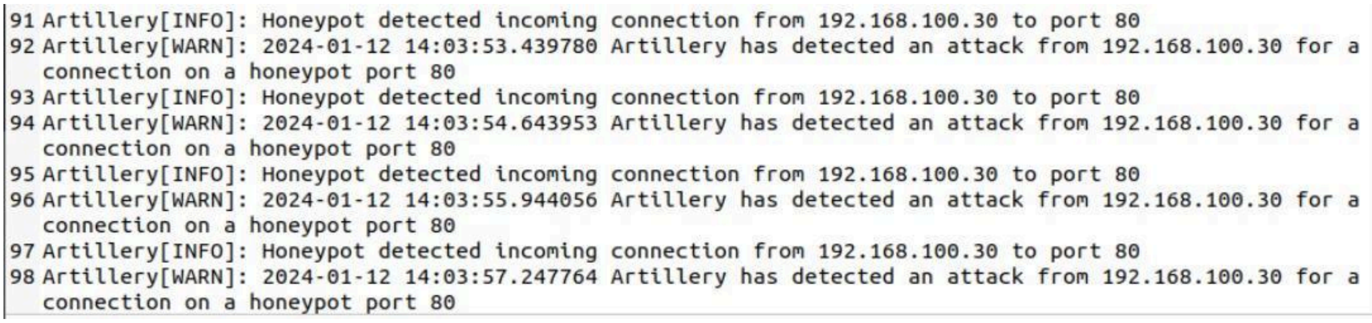
- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/xss-filter-evasion-cheatsheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- http://www.cgisecurity.com/xss-faq.html
- http://www.scriptalert1.com/

Figure 10 : XSS Attack

- **Automated Response Mechanisms:**

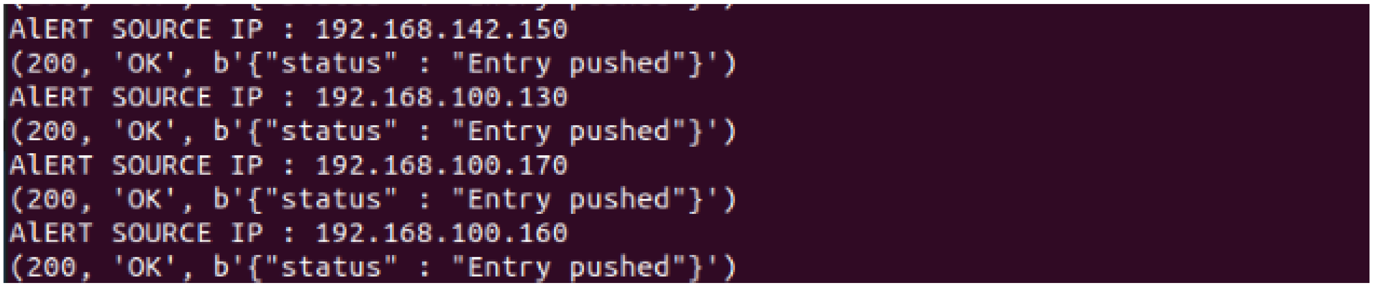
Validated the system's automated response mechanisms, including configurable alerting rules. Integration with the SDN Controller was tested. System demonstrated prompt and precise responses to suspicious activities, such as traffic redirection to the honeypot.

The following figures shows the successful redirection from the main-server to the honeypot after an attack on port 80, and that the honeypot got the alert of this attack.



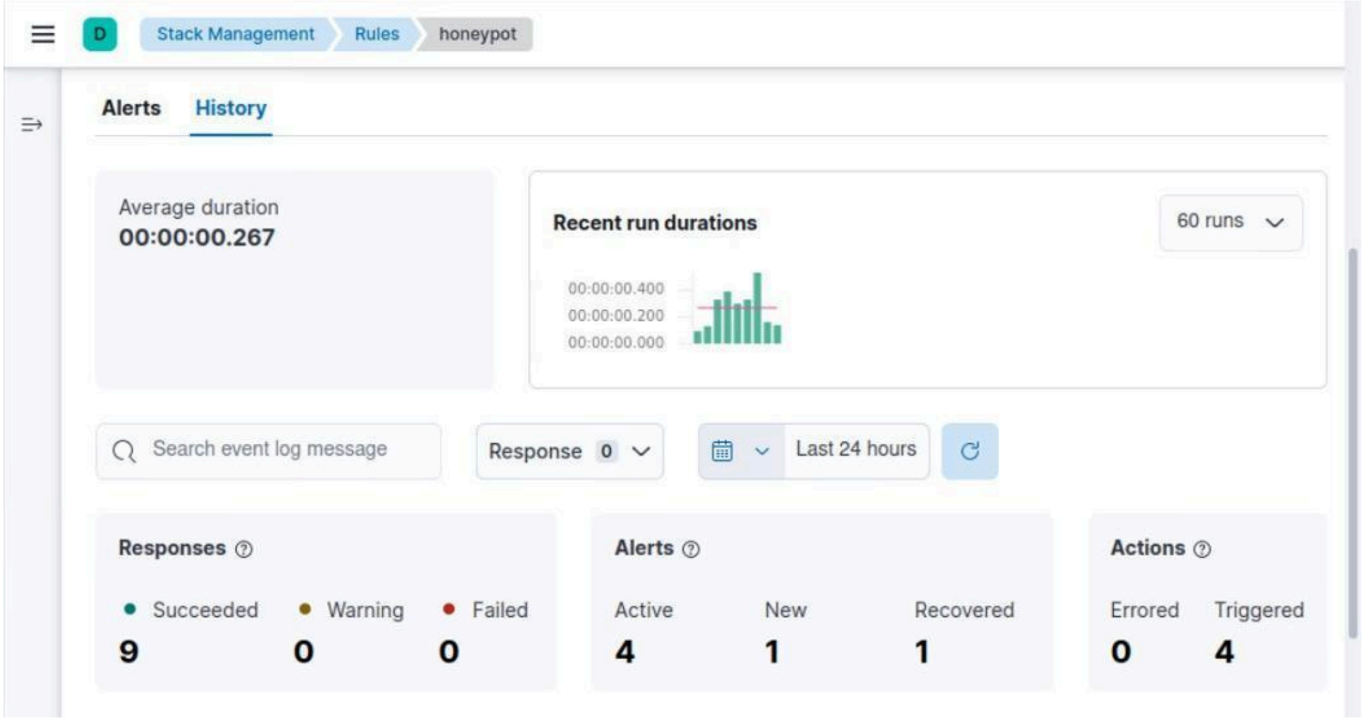
```
Artillery[INFO]: Honeypot detected incoming connection from 192.168.100.30 to port 80
Artillery[WARN]: 2024-01-12 14:03:53.439780 Artillery has detected an attack from 192.168.100.30 for a
  connection on a honeypot port 80
Artillery[INFO]: Honeypot detected incoming connection from 192.168.100.30 to port 80
Artillery[WARN]: 2024-01-12 14:03:54.643953 Artillery has detected an attack from 192.168.100.30 for a
  connection on a honeypot port 80
Artillery[INFO]: Honeypot detected incoming connection from 192.168.100.30 to port 80
Artillery[WARN]: 2024-01-12 14:03:55.944056 Artillery has detected an attack from 192.168.100.30 for a
  connection on a honeypot port 80
Artillery[INFO]: Honeypot detected incoming connection from 192.168.100.30 to port 80
Artillery[WARN]: 2024-01-12 14:03:57.247764 Artillery has detected an attack from 192.168.100.30 for a
  connection on a honeypot port 80
```

Figure 11 : Honeypot Attack Logs



```
AlERT SOURCE IP : 192.168.142.150
(200, 'OK', b'{"status" : "Entry pushed"}')
AlERT SOURCE IP : 192.168.100.130
(200, 'OK', b'{"status" : "Entry pushed"}')
AlERT SOURCE IP : 192.168.100.170
(200, 'OK', b'{"status" : "Entry pushed"}')
AlERT SOURCE IP : 192.168.100.160
(200, 'OK', b'{"status" : "Entry pushed"}')
```

Figure 12 : Source IP Alert



Average duration 00:00:00.267

Responses: ● Succeeded 9, ● Warning 0, ● Failed 0

Alerts: Active 4, New 1, Recovered 1

Actions: Errored 0, Triggered 4

Figure 13 : Honeypot Alert on Kibana

- **ELK stack:**

Thorough testing of the logging infrastructure, including Syslog-ng and the ELK Stack. Ensured accurate collection, transmission, and analysis of logs from various sources. Validation confirmed the system's capacity to provide comprehensive visibility into network activities. Visualize the logs to start analyzing on it, as shown in figures an example for dashboard and the alert page:

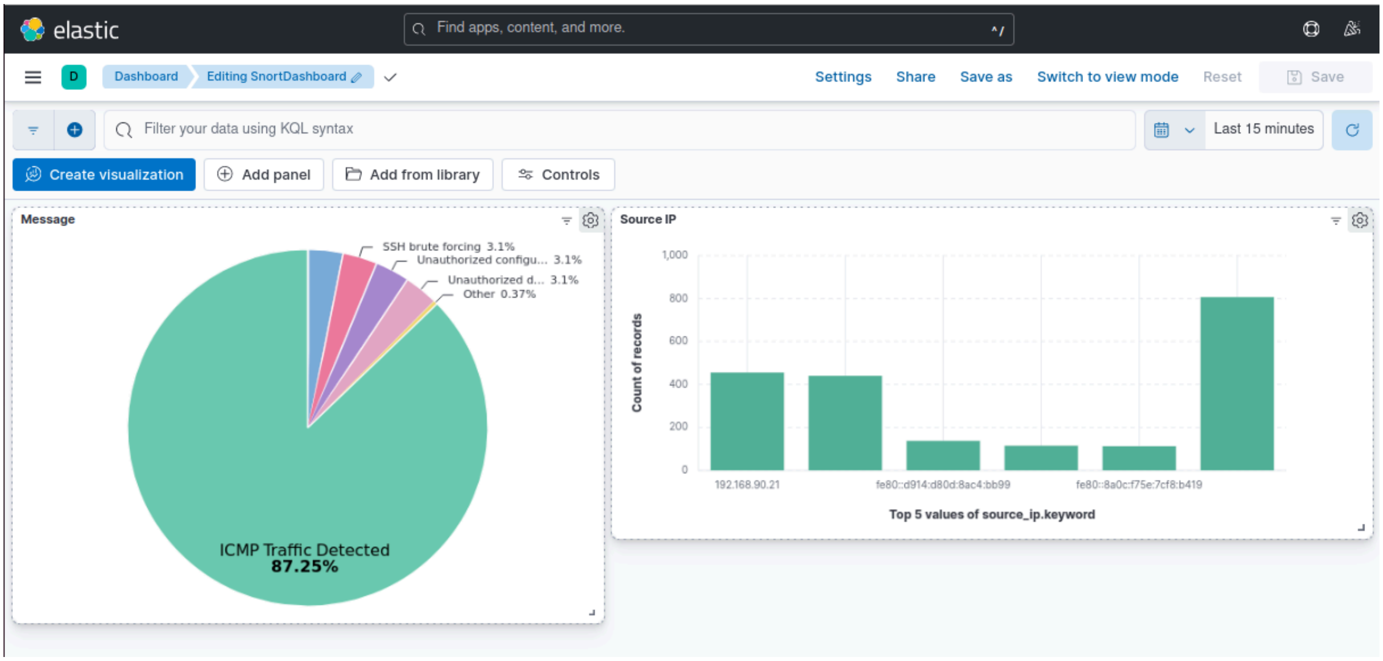
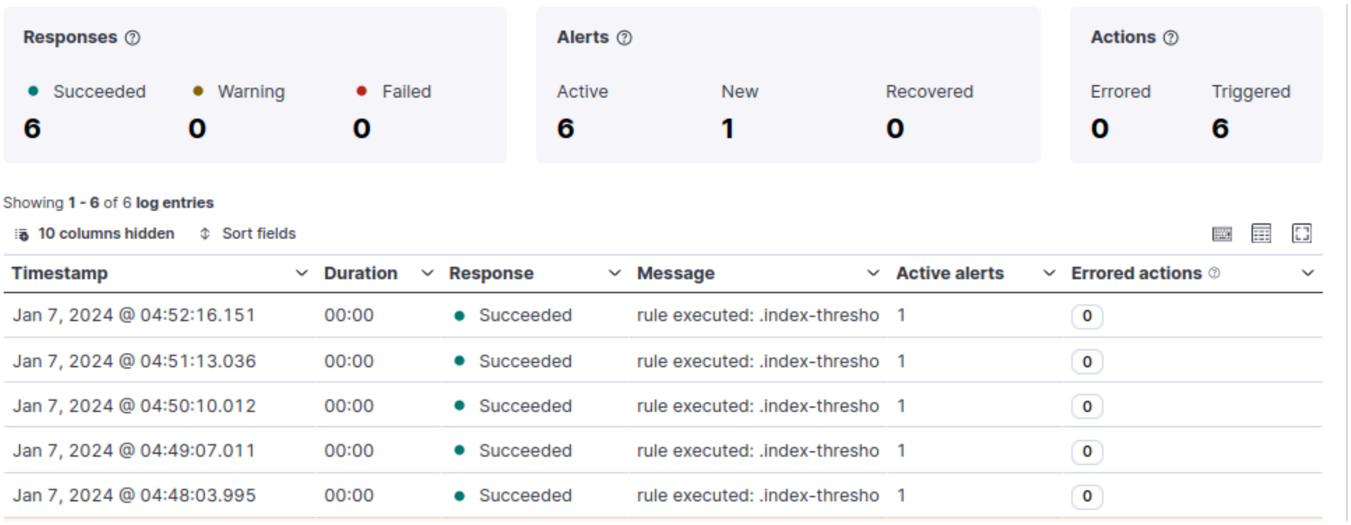


Figure 14 : Snort Dashboard



Responses

● Succeeded	● Warning	● Failed
6	0	0

Alerts

Active	New	Recovered
6	1	0

Actions

Errored	Triggered
0	6

Showing **1 - 6** of 6 **log entries**

10 columns hidden   Sort fields

Timestamp	Duration	Response	Message	Active alerts	Errored actions
Jan 7, 2024 @ 04:52:16.151	00:00	● Succeeded	rule executed: .index-thresho	1	0
Jan 7, 2024 @ 04:51:13.036	00:00	● Succeeded	rule executed: .index-thresho	1	0
Jan 7, 2024 @ 04:50:10.012	00:00	● Succeeded	rule executed: .index-thresho	1	0
Jan 7, 2024 @ 04:49:07.011	00:00	● Succeeded	rule executed: .index-thresho	1	0
Jan 7, 2024 @ 04:48:03.995	00:00	● Succeeded	rule executed: .index-thresho	1	0

Figure 15 : Snort Alerts

3. Overall Results and Analysis

The project has yielded significant progress and outcomes in enhancing cybersecurity measures through the implementation of automated decoys, honeypots, and SDN. While there have been notable achievements, some challenges and unexpected hurdles have provided valuable learning experiences.

3.1 Positive Aspects and Achievements:

1. Successful Implementation of Decoys and Honeypots: The successful deployment of automated decoys and honeypots has provided valuable insights into potential threats and attacker tactics.
2. Dynamic Adaptation with SDN: The integration of SDN for dynamic network configurations has proven effective in creating a more challenging environment for attackers.
3. Centralized Logging and Monitoring: The establishment of a centralized logging system, along with real-time monitoring and alerting, has enhanced the team's ability to identify and respond to security incidents promptly.
4. Web Server Protection: Implementing Snort and ModSecurity for web server protection has fortified the system against unauthorized access and potential modifications to critical assets.
5. Real-time Alerts and Automated Responses: The system demonstrated the ability to generate real-time alerts and trigger automated responses, redirecting suspicious traffic to honeypots for containment.

3.2 Challenges and Lessons Learned:

1. Artillery Honeypot Limitations: The limitations of Artillery honeypot, including resource constraints and compatibility issues with specific Node.js versions, required additional effort to overcome. Future projects may benefit from exploring alternative honeypots.
2. ELK Stack Configuration Complexity: The complexity involved in configuring the ELK stack components highlighted the importance of detailed documentation. A more comprehensive guide can streamline the setup process for future projects.
3. Resource Availability and Collaboration: Limited availability of resources and challenges in collaborating with industry experts emphasized the need for proactive communication and resource planning.
4. Continuous Monitoring and Automation: The need for continuous monitoring, automation of honeypot deployment, and setting up email alerts for specific attack vectors emerged as crucial aspects for further improvement.
5. Tool Transition Challenges: Occasionally, after investing significant effort in configuring and integrating a particular tool into the system, the team encountered the need to transition to another tool. This transition was prompted by the discovery of an alternative

tool offering similar features along with additional functionalities aligned with the evolving project requirements.

3.3 Learning Outcomes and Career Objectives:

1. Technical Skill Development: The project has provided hands-on experience in developing deception solutions, deploying honeypots, and utilizing SDN. These technical skills are directly applicable to the graduate program and enhance the team's overall cybersecurity expertise.
2. Problem-Solving and Adaptation: Overcoming challenges such as blocked malicious IPs and unforeseen compatibility issues has strengthened the team's problem-solving and adaptation skills, essential for success in real-world cybersecurity scenarios.
3. Collaboration and Communication: The collaboration with industry experts and the challenges faced in communication underscore the importance of effective collaboration and clear communication in addressing complex cybersecurity issues.
4. The project's focus on real-world cybersecurity challenges and collaboration with industry experts has provided practical insights that align with career objectives in the cybersecurity field. The experience gained is directly applicable to addressing evolving threats in professional settings.

3.4 Project Success and Future Recommendations:

While the project has been successful in achieving its objectives, continuous refinement and improvement are essential for long-term success. Future recommendations include:

1. Documentation Enhancement: Continue to enhance and maintain detailed documentation for all aspects of the project, from system configuration to troubleshooting guides.
2. Ongoing Monitoring and Automation: Implement continuous monitoring and automated responses to security incidents, ensuring a proactive defense approach.
3. Explore Alternative Technologies: Explore alternative honeypots and technologies to address limitations and further diversify the defensive measures.
4. Regular Updates and Patching: Regularly update software components and address compatibility issues promptly to ensure a secure and resilient cybersecurity infrastructure.
5. Strategic Tool Selection: Consider a strategic approach to tool selection, anticipating the possibility of tool transitions, and evaluating new tools based on both current and potential future project requirements. This proactive approach can minimize disruptions caused by tool changes.

Overall Evaluation:

The project has undeniably achieved success in advancing cybersecurity measures, showcasing the team's dedication, adaptability, and technical prowess. The challenges encountered have served as valuable learning experiences, contributing to the team's problem-solving skills. The commitment to continuous improvement, as outlined in the recommendations, positions the project for sustained success and prepares the team for future cybersecurity endeavors. Overall, the project's success is a testament to the team's competence and resilience in tackling complex cybersecurity challenges.

4. Deployment Plan

Enhancing Cybersecurity with Automated Deception, our primary objective is to orchestrate a seamless transition to our advanced cybersecurity solution, ensuring end users feel confident and secure throughout the process.

1. **Pre-Deployment Preparation:**

   - Environment Check: We'll start with a deep dive into the operational environment, understanding its nuances, and ensuring compatibility.
   - Resource Readiness: We want to make sure the environment has all the necessary resources, from hardware to software, to handle our deployment.

2. **Installation and Configuration:**

   - Setting Up Components: We'll deploy our software components – the Centralized Monitoring System, SIEM, Artillery honeypot, Open vSwitch, and other security tools.
   - Integration Magic: We'll integrate these components seamlessly into the existing systems, making sure they play well together.

3. **Testing and Validation:**

   - Functional Test Run: Extensive testing to confirm that everything works smoothly – from the Centralized Monitoring System to the automated traffic redirection to honeypots.
   - Security Stress Test: Penetration testing time to see how resilient our system is against potential cyber threats.

4. **User Training:**

   - Knowledge Transfer: Training sessions for our end users and IT team, complete with detailed documentation.
   - Hands-On Simulations: We'll run simulations to get everyone comfortable with the system's response during various attack scenarios.

5. **Deployment Rollout:**

- Step-by-Step Launch: We're thinking of a phased rollout, starting small and gradually expanding to avoid any hiccups.
- Eyes on the Prize: Continuous monitoring during deployment to catch and fix issues on the fly.

6. **Post-Deployment Maintenance:**

- Always Watchful: Continuous monitoring doesn't stop after deployment. We'll keep an eagle eye on the system's performance and security alerts.
- Stay Updated: Regular updates to keep the software on its toes against evolving cyber threats.

7. **User Feedback and Iterative Improvements:**

- Feedback Welcome: We want to hear from our users – their thoughts, concerns, and suggestions.
- Learn and Upgrade: Use this feedback to fine-tune the system, continuously improving and staying ahead of the curve.

5. Conclusions

In conclusion, this project addresses the critical challenges in traditional cybersecurity measures by introducing an innovative approach that combines automated decoys, honeypots, and SDN for dynamic network reconfigurations. The problem definition emphasizes the limitations of existing security measures, highlighting the need for a proactive solution to swiftly identify and respond to cyber threats.

Also navigated a complex landscape, engaging with external systems and utilizing diverse tools such as Artillery Honeypot, Open vSwitch, KVM, IDS, ModSecurity WAF, and ELK Stack. From routers to honeypots, each component plays a crucial role, dynamically altering network configurations, simulating real-world scenarios, and triggering alerts in real-time.

The architectural view provides a clear understanding of the key components, such as log collection, SIEM integration, network monitoring, web application security, alert generation, automated traffic redirection, and the honeypot infrastructure. These components collectively form a centralized control system that acts as the nerve center for network security.

Addressing specific requirements ensures that the system meets stakeholders' needs, covering aspects like network monitoring, logging, SIEM integration, real-time alerting, user interface, performance, scalability, and reliability. The design showcases a thoughtful and strategic approach, deploying cutting-edge technologies, a resilient architecture, and adaptive solutions to fortify cybersecurity defenses against evolving threats.

6. References

[1] S. Hudak Jr., "Automatic Honeypot Generation and Network Deception," United States Military Academy, West Point, NY, 10996, USA.

[2] C.-Y.J. Chiang, A. Poylisher, and R. Chadha, "Enhancing Cyber Defense with Autonomous Agents Managing Dynamic Cyber Deception (Position Paper)," Vencore Labs.

[3] C.-Y.J. Chiang, Y.M. Gottlieb, S.J. Sugrim, R. Chadha, C. Serban, A. Poylisher, L.M. Marvel, and J. Santos, "ACyDS: An Adaptive Cyber Deception System," Applied Communication Sciences, Basking Ridge, NJ, USA.

[4] S. Vasylyshyn, V. Susukailo, I. Opirskyy, Y. Kurii, and I. Tyshyk, "A Model of Decoy System Based on Dynamic Attributes for Cybercrime Investigation," Lviv Polytechnic National University, Department of Information Security, Lviv, Ukraine

[5] P. Manso, J. Moura, and C. Serrão, "SDN-Based Intrusion Detection System for Early Detection and Mitigation of DDoS Attacks,