Text

Description automatically generated

A Project Report on

**Strengthening Network Security Posture and Threat Detection with Cyber Canary**

Submitted in partial fulfilment for the award of the degree of

**Master of Technology**

In **Cybersecurity**

Submitted by

**Javeriya Vahid**

R21MTC55

Under the Guidance of

**Nishant Krishna**

Executive Director, Visiminds Technologies

REVA Academy for Corporate Excellence

**REVA University**

Rukmini Knowledge Park, Kattigenahalli,

Yelahanka, Bangalore – 560064

**September 2024**



# Candidate’s Declaration

I, **Javeriya Vahid**, hereby declare that I have completed the project work towards a **Master of Technology in Cybersecurity** at REVA University on the topic entitled**Strengthening Network Security Posture and Threat Detection with Cyber Canary** under the supervision of **Nishant Krishna Executive Director, Visiminds Technologies**. This report embodies the original work done by me in partial fulfilment of the requirements for the award of the degree for the academic year 2024.

A close up of a sign

Description automatically generated

Place: Bengaluru Name of the Student: Javeriya Vahid

Date: 21/09/2024 Signature of Student

A black and orange text

Description automatically generated

# Acknowledgment of Project Ownership and Usage Rights

I, Javeriya Vahid, a student enrolled in the M. Tech Program and in the 2022 year at RACE, hereby acknowledge that any project, including but not limited to software, hardware, research, or other intellectual property created by me during my academic tenure at RACE, is the property of RACE, REVA University.

I understand and agree that RACE has the exclusive rights to use, reproduce, modify, or distribute the aforementioned project for academic, research, and further development purposes. This includes the right to monetize, commercialize, or otherwise exploit the projects as deemed fit by RACE.

I acknowledge that I have no objection to RACE, REVA University, using, reproducing, or further developing the projects for the benefit of the institution and its academic community. I further affirm that any commercial or research activities related to the projects conducted by RACE shall not require additional consent or approval from me.

This acknowledgment is made willingly and without any reservations. I am grateful for the education and opportunities provided by RACE, and I recognize the importance of contributing to the academic and research goals of the institution.

A close up of a sign

Description automatically generated

Place: Bengaluru Name of the Student: Javeriya Vahid

Date: 21/09/2024 Signature of Student



# Certificate

This is to certify that the project work entitled **Strengthening Network Security Posture and Threat Detection with Cyber Canary** has been carried out by **Javeriya Vahid** with **R21MTC55**, who is a bonafide student at REVA University, is submitting the **Second-year** project report in fulfilment for the award of **Master of Technology in Cybersecurity** during the academic year **2024**. The project report has been tested for plagiarism and has passed the plagiarism test with a similarity score of less than 15%. The project report has been approved as it satisfies the academic requirements for the project work prescribed for the said degree.

A signature of a person

Description automatically generated

Signa e Signature of the Director

Nishant Krishna Dr. Shinu Abhi

Director

External Viva

Names of the Examiners

<1>

<2>

Place: Bengaluru

Date: 21/09/2024



# Acknowledgment

I would like to acknowledge the support provided by the founder and Hon’ble Chancellor, **Dr. P Shayma Raju**, Vice-Chancellor, **Dr. N Ramesh**, and Registrar, **Dr. K S Narayanaswamy**.

I would like to acknowledge the great support and encouragement of the Director, of Corporate Training for RACE, Dr. Shinu Abhi, and all the eminent RACE Faculty who mentored with efficacy throughout the course. I would like to express my sincere gratitude to my mentor, Nishant Krishna for his invaluable guidance, support, and encouragement throughout this project. Without his continuous feedback and valuable insights, this project would not have been possible.

I would like to extend my sincere gratitude to the Associate Professor, of Corporate Training for RACE, Dr. Rashmi Agarwal for step-by-step guidance in writing and documentation at all stages of the project. I would like to acknowledge the contributions of the Admin, Infrastructure team, and extended support staff at RACE, namely Irshad, Jordan, Dincy, and the diligent interns for all the help and support.

Finally, I would like to express my heartfelt gratitude to my family and friends for their unwavering support and encouragement throughout this journey. Their love, understanding, and patience have been my pillars of strength and motivation.

Place: Bengaluru

Date: 21/09/2024



# Similarity Index Report

This is to certify that this project report titled **Strengthening Network Security Posture and Threat Detection with Cyber Canary**was scanned for similarity detection. The process and outcome are given below. The plagiarism report is attached in the appendix.

Software Used: Turnitin

Date of Report Generation: 18/09/2024

Similarity Index in %: 4

Total word count: 6332

Name of the Guide: Nishant Krishna

A close up of a sign

Description automatically generated

Place: Bengaluru Name of the Student: Javeriya Vahid

Date: 21/09/2024 Signature of Student

Verified by: Irshad Ahmed

Signature

Dr. Shinu Abhi,

Director, Corporate Training

# List of Abbreviations

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Abbreviation** | **Long Form** |
| 1 | VM | Virtual Machine |
| 2 | DDOS | Distributed Denial of Service. |
| 3 | FTP | File Transfer Protocol |
| 4 | HTTP | Hyper Text Transfer Protocol |
| 5 | SMTP | Simple Mail Transfer Protocol |
| 6 | IP | Internet Protocol |
| 7 | IDS | Intrusion Detection System |
| 8 | DMZ | Demilitarized Zone |
| 9 | POC | Proof of Concept Code |
| 10 | VPS | Virtual Private Server |
| 11 | SMB | Server Message Block |
| 12 | AWS | Amazon Web Services |
| 13 | IDS | Intrusion Detection System |
| 14 | SIEM | Security Information and Event Management |
| 15 | CAV | Connected and Autonomous Vehicle |
| 16 | OWASP | Open Web Application Security Project |
| 17 | LDAP | Lightweight Directory Access Protocol |
| 18 | SQL | Structured Query Language |

# List of Figures

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Page No.** |
| Figure No. 1.1 | Network Canary | 12 |
| Figure No. 1.2 | OWASP | 13 |
| Figure No. 1.3 | Selected OWASP CONTROLS | 13 |
| Figure No. 5.1 | Proposed Methodology for Cyber Canary | 23 |
| Figure No. 5.2 | Attack Simulation | 24 |
| Figure No. 7.1 | Network Canary Architecture | 27 |
| Figure No. 7.2 | Cyber Canary Proposed Solution | 28 |
| Figure No. 8.1 | *rsyslog* Client File | 29 |
| Figure No. 8.2 | Segregation of Log Files | 30 |
| Figure No. 8.3 | *rsyslog* Server File | 30 |
| Figure No. 8.4 | Simulation using LDAP | 31 |
| Figure No. 8.5 | Simulation using MySQL | 31 |
| Figure No. 8.6 | Simulation using FTP | 31 |
| Figure No. 9.1 | LDAP Broken Access Control | 32 |
| Figure No. 9.2 | Injection using LDAP | 32 |
| Figure No. 9.3 | Logs for LDAP Injection | 33 |
| Figure No. 9.4 | LDAP Security Misconfiguration | 33 |
| Figure No. 9.5 | Log for LDAP Security Misconfiguration | 33 |
| Figure No. 9.6 | FTP Broken Access | 33 |
| Figure No. 9.7 | SQL Injection on FTP Server | 34 |
| Figure No. 9.8 | Log for SQL injection on FTP | 34 |
| Figure No. 9.9 | Ftp Security Misconfiguration | 34 |
| Figure No. 9.10 | Log for Security Misconfiguration on FTP | 34 |
| Figure No. 9.11 | Broken Access for MySQL | 35 |
| Figure No. 9.12 | Broken Access causes Sensitive Data Exposure. | 35 |
| Figure No. 9.13 | Blind MySQL Injection | 36 |
| Figure No. 9.14 | Log of MySQL Injection | 36 |
| Figure No. 9.15 | Security Misconfiguration on MySQL | 37 |
| Figure No. 9.16 | Logs Security Misconfiguration on MySQL | 37 |
| Figure No. 10.1 | FTP Logs | 38 |
| Figure No. 10.2 | LDAP Logs | 39 |
| Figure No. 10.3 | SQL Logs | 39 |

# Abstract

Network canaries act as early warning systems by luring attackers to interact with decoy resources, such as files or services, which are closely monitored by Administrators who are notified of possible intrusions when these canary assets are accessed or altered, as alerts are raised. The term *cyber canary* describes a cybersecurity idea that was influenced by the old method of detecting hazardous gas carbon monoxide in coal mines by carrying a canary bird the death of the canary alerted the miners to a high level of gas and miners would leave the place to protect themselves from the hazardous gas.

There is a challenge in mapping the attacks with real-time vulnerabilities so that the attack can be identified at an earlier stage to protect the network. The flexible network configuration for canaries and map it with OWASP vulnerabilities in real time to identify threats such as security misconfiguration, Injection, and Broken Access Control. To alert any interaction with the vulnerable machines a central server is created to monitor any activity more closely. Hence integrating the network of canary machines with *rsyslog* significantly enhances real-time threat detection.

To mimic the OWASP Top 10 vulnerabilities, a network of susceptible Ubuntu Virtual Machine (VM) running FTP, MySQL, and LDAP services has been established. Controls selected as of 2021 are Broken Access Control (A01), Injection (A03), and Security Misconfiguration (A05). These systems are vulnerable to attacks due to data manipulation, unauthorized access, and misconfigurations. The ability to generate logs when any action takes place inside the network has been set up to have centralized logs using the *rsyslog* utility where the logs from each vulnerable machine are accumulated and available for detection.

The vulnerable machines are successfully placed in the network as canaries; any interaction with these canary machines results in alerting the centralized server. Any interaction or modification is seen through the logs. The segregation of logs done at the centralized server is accurate and helps in analyzing the logs concerning the vulnerable machine.

***Keywords: Honeypot, Cohesive Event Analysis, Centralized VM, Network Canary, Cyber Canary, Security, OWASP***

**Table of Contents**

[Candidate’s Declaration 2](#_Toc172621482)

[Acknowledgment of Project Ownership and Usage Rights 3](#_Toc172621483)

[Certificate 4](#_Toc172621484)

[Acknowledgment 5](#_Toc172621485)

[Similarity Index Report 6](#_Toc172621486)

[List of Abbreviations 7](#_Toc172621487)

[List of Figures 7](#_Toc172621488)

[Abstract 9](#_Toc172621489)

[Chapter 1: Introduction 11](#_Toc172621490)

[Chapter 2: Literature Review 16](#_Toc172621491)

[Chapter 3: Problem Statement 21](#_Toc172621492)

[Chapter 4: Objectives of the Study 22](#_Toc172621493)

[Chapter 5: Project Methodology 23](#_Toc172621494)

[Chapter 6: Resource Requirement Specification 26](#_Toc172621495)

[Chapter 7: Software Design 27](#_Toc172621496)

[Chapter 8: Implementation 29](#_Toc172621497)

[Chapter 9: Testing and Validation 32](#_Toc172621498)

[Chapter 10: Analysis and Results 38](#_Toc172621499)

[Chapter 11: Conclusions and Future Scope 40](#_Toc172621500)

[Bibliography 41](#_Toc172621501)

[Appendix 43](#_Toc172621502)

[Plagiarism Report 44](#_Toc172621503)

[Gitlink 45](#_Toc172621504)

# Chapter 1: Introduction

In today's digital age, cybersecurity has become a critical concern for organizations of all sizes. The increasing reliance on technology and the internet has made networks and infrastructures vulnerable to a wide range of cyber threats. *Cyber Canary* is an innovative approach to network security that focuses on intelligent threat signaling and monitoring to safeguard organizational networks and infrastructure.

**Cyber Canary**

The term *cyber canary* describes a cybersecurity idea that was influenced by the old method of detecting hazardous gasses in coal mines by deploying canaries. In coal mines, canaries were employed as early warning systems to identify harmful gasses like carbon monoxide. The birds were carried by miners in cages, and if the canaries began to sing less or displayed other symptoms of distress, it was an indication that hazardous chemicals were present, and the miners had to flee. This procedure brought attention to the canary's function in life-saving alert systems and influenced contemporary ideas about threat detection. A cyber canary is a proactive monitoring device that identifies possible security problems before they become more serious in the digital sphere. It serves as a tool for early warning, informing companies when harmful behaviour is detected on their networks [1].

**Advanced Cybersecurity Solutions are Required for the Evolving Threats**

The below-mentioned factors are the security threats to any organization, that need to be addressed with the latest technologies like cyber canary.

1. **Evolving Threat Landscape:** Attackers use advanced methods like ransomware, phishing activity, zero-day vulnerabilities, and cyber threats that are always changing. Security solutions must be dynamic and adaptable to these ever-evolving threats.
2. **Increased Data Sensitivity**: Businesses deal with enormous volumes of sensitive data, such as financial records, intellectual property, and personal data. Maintaining confidence and adhering to legal requirements depend on protecting this data from breaches.
3. **Business Continuity:** Cyberattacks have the potential to seriously impair company operations, resulting in downtime and large financial losses. Strong security measures must be put in place to guarantee company continuity and resistance to cyberattacks.

**Network Canary Setup**

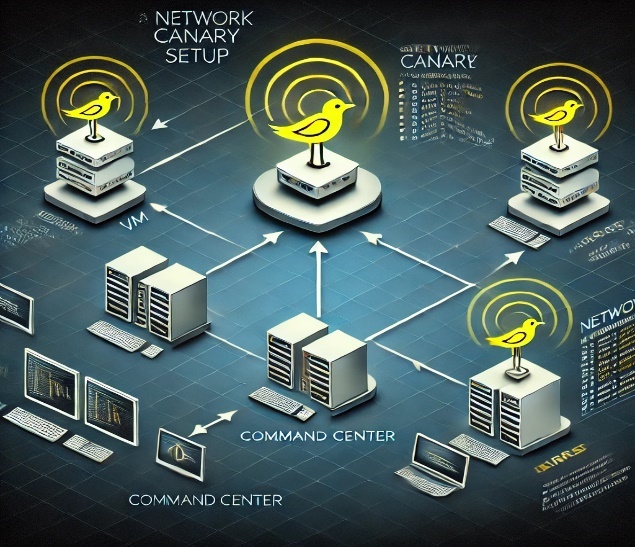


Fig. 1.1: Network Canary

By implementing efficient methods and technology, businesses can safeguard consumer trust, maintain operational continuity, and protect sensitive data. Furthermore, this knowledge strengthens communities by encouraging digital literacy and offering people the tools to stay safe online. Organizations and communities can work together to reduce risks and improve overall resilience against cyberattacks by promoting a culture of cybersecurity awareness. In conclusion, taking a proactive approach like a canary, as shown in Fig. 1.1, safeguards organizational resources and makes everyone's digital environment safer and more secure.

**OWASP**

An international project called Open Web Application Security Project (OWASP) aims to find serious flaws in web applications to improve software security. Its Top Ten list as shown in Fig. 1.2 identifies the most prevalent and serious security threats and offers recommendations for reducing them to shield systems from assaults. From the list of OWASP controls, we have selected three controls as shown in Fig.1.3 and applied them in our project to map with the vulnerable machine in the network.

**A screenshot of a computer screen

Description automatically generated**

Fig. 1.2: OWASP CONTROLS

**A close-up of a computer screen

Description automatically generated**

**A close-up of a computer screen

Description automatically generated**

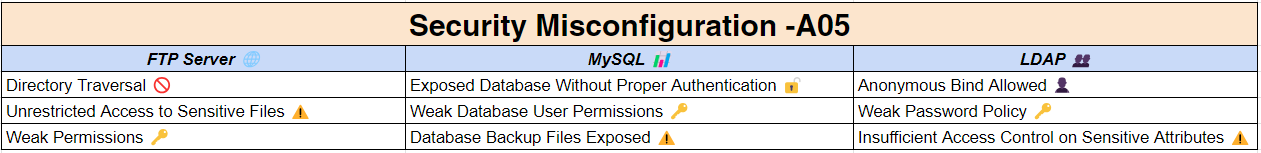
****

Fig. 1.3: Selected OWASP CONTROLS

**Importance of rsyslog in Network Monitoring**

*rsyslog* gathers, processes, and forwards log messages from many sources over a network using the syslog protocol. It can filter, classify, and store logs according to specified criteria and supports both TCP and UDP for log transport, guaranteeing dependability and speed. Log encryption, message queuing, and transferring logs to distant servers for centralized monitoring and analysis are some of its sophisticated features.

Through its effective collection, filtering, and transmission of log data from numerous devices throughout the network, *rsyslog* is an essential tool for network monitoring. It makes centralized log management possible, giving administrators real-time network event monitoring and security incident detection. By handling massive amounts of log data, *rsyslog's* high-performance capabilities enable prompt alerts and reactions to possible risks. It simplifies troubleshooting and auditing by centralizing logs from several systems, making it easier to see trends and abnormalities. Furthermore, *rsyslog* improves compliance by guaranteeing that logs are kept safely and readily available for inspection.

***rsyslog* Working**

The installed *rsyslog* utility in the Linux machine works as follows:

1. **Log Collection and Centralization:** To facilitate simpler administration and monitoring, *rsyslog* collects logs from numerous devices connected to a network and centralizes them on a single server.
2. **Flexible Filtering and Parsing:** It enables precise log management and storage by filtering logs according to unique rules like severity levels, keywords, or source devices.
3. **Secure Log Transmission:** To prevent sensitive information from being intercepted, *rsyslog* offers encryption (TLS) for the safe transfer of log data across the network.
4. **Log Storage and Rotation:** To efficiently manage disk space, logs can be stored in a variety of formats (such as plain text or JSON) and have the option of automatic log rotation and archiving.
5. **Real-time Monitoring and Alerts:** By setting up alerts for patterns or events, administrators may keep an eye on the health of their network in real-time and take proactive measures to address security issues.

In conclusion, businesses need to implement cutting-edge technologies like the Cyber Canary to protect critical data and identify threats early in the dynamic digital ecosystem. When combined with *rsyslog* for network monitoring, these tools improve problem-solving in real-time and provide strong security.

# Chapter 2: Literature Review

A thorough analysis of the Cyber canary in the network infrastructure, the procedure of the implementation of these canaries, and the use of the canaries as an early alerting system to prevent any threat from occurring, by filtering out the logs at the right time before the occurrence of the attack is the most useful task to prevent our organizations.

**Analysis of Honeypot**

The research work by Obied Honeypots cannot stop attacks on the network to which they are connected, but when used in conjunction with other defensive technologies like firewalls and intrusion detection systems, they can aid in the identification and detection of illegal activity. One benefit of honeypots is that they frequently produce small amounts of highly valuable data However, depending on the situation, analysts may face difficulties analysing collected dataset if it grows over time. In such circumstances, security researchers might make use of visualization tools and methodologies to gain a deeper comprehension of the functioning of the honeypot [2].

**Analysis of the Exploits**

As per the study on the exploits database, Cybercriminals and malicious malware possessing the ability to spread automatically are continuously searching the Internet for susceptible targets. These could include Computers or servers vulnerable to specific types of attacks. Typically, a human hacker looks for services that are active on a system to determine how to launch the attack. Several vulnerabilities are archived on websites such as Exploit-DB, it contains what is generally called a "POC" (Proof of Concept code) that can be used to exploit a susceptible system and take control of it or do illegal actions [3].

**Experimental Setup of Honeypot**

As effective methods for boosting network security, honeypots, canary tokens, and Docker have grown in popularity. These technologies are often used in conjunction with cloud computing platforms like Amazon Web Services (AWS). This study intends to investigate the advantages of using Docker, Canary tokens, and Honeypots on AWS, offering a thorough examination of the advantages and security concerns of this methodology. Organizations may proactively safeguard their networks and avert cyberattacks by knowing how these technologies interact. Intrusion Detection System (IDS) has been employed in a hybrid context by the author [4].

Applications that are deployed traditionally run on an operating system and depend on shared libraries and dependencies; it can lead to conflicts when updates are applied. On the other hand, applications and all their dependencies, libraries, and configuration files are encapsulated into self-contained units called containers by containerized environments using a container engine. Applications may be created, tested, and run reliably in a variety of contexts because of this packaging. These containers are easier to create and maintain with the help of the open-source container engine Docker. By sitting atop the operating system, it reduces compatibility problems, streamlines deployment, and offers a stable runtime environment. With this method, infrastructure is converted from a monolithic configuration to a modular, container-based architecture [5].

**Analysis of Canary**

Canary birds are used in mines to alert workers to the presence of dangerous gasses, hence the origin of the word *canary*. Canaries are more susceptible to dangerous gasses than humans. A canary's sickness signals to the miners that it is time to leave the shaft. The canaries encounter nearly the same climatic and operational circumstances as the real system because of their integrated placement. A system's canary devices can be utilized to warn of impending failures owing to wear-out failure.

To carry out a successful attack, attackers must first go through a trial-and-error procedure that provides them the chance to identify possibly exploitable flaws in the target application.  
By keeping an eye out for any exceptions that an attacker causes within the program, developers of applications can also benefit from this exploratory behaviour. Applications can become attack-aware when these attacker-induced exceptions are paired with additional high-fidelity attack indications since they behave similarly to a canary [6].

According to a preliminary analysis of PHP-based web application frameworks, even those. that make use of exceptions Regarding web frameworks in general, the D*jang*o web application framework's Suspicious Operation exception classes seem to be the most comparable approach to the suggested attacker-induced exceptions [6]. But rather than being real assaults, these are a small subset of security flaws that need to be manually assigned by a developer.

The canary device's sensitivity to operational loading conditions and the environment determines how much the target system will deteriorate. Having the canary device fail by the same failure mechanism as the target system would be ideal and beneficial. Furthermore, the canary device must fail before the target system. This aids in identifying the underlying reason and implementing the proper health management strategies to improve the availability and dependability of the system. It makes no difference if the mechanisms match in cases when there is an obvious mapping from the canary devices' failures to the target system's failure. Prognostic distance is the difference between the canary device's and the real system's time to failure [7].

The process of evaluating an amount of deviation or degradation of a product from its expected normal operating conditions, and then projecting the future reliability of the product is the definition of PROGNOSTICS. Due to its ability to lower life cycle costs, estimate maintenance needs, and provide early warning of impending problems, prognostics for   
electrical devices and systems have drawn more attention. Prognostic implementation techniques for electronic products and systems involve the use of canary devices, the monitoring of environmental and usage circumstances a product encounters in its application environment, and the monitoring of failure precursors. One method of implementing prognostics is called Physics-Of-Failure (POF), It estimates a product's remaining usable life by using information about its geometry, material qualities, loading conditions, and  
failure processes over its life cycle factors affecting the operational environment include humidity, vibration, temperature, operating voltage, and the presence of corrosive materials [8].

A botnet's attack nodes are distributed in such a way that identifying and countering them depends on how intricately it was constructed. Numerous methods of defence against DDoS   
There are attacks. Early methods of DDoS detection cantered on statistical measurements to compare the distinctions between malicious and normal Machine-to-Machine (M2M) communication. The ability to distinguish malicious from non-malicious M2M communication traffic using standard statistical methods declined as the complexity of malicious and non-malicious traffic became indistinguishable, which gave rise to feature-driven detection [9]

Over the past ten years, a lot of research has been done to increase the functionality, performance, and efficiency of small mobile computers; nonetheless, during that same period, battery performance has only marginally increased [10]. Extending the use of devices requires preserving battery life.

An extensive analysis of Network Security Monitoring (NSM) systems most solutions incorporate at least one of these features: A detector finds anomalies in data streams, an inspector permits data exploration, an actuator initiates automatic network/subsystem configuration actions, a parser converts data formats, an integrator integrates multiple data sources into a single stream, and a sensor gathers data from a network subsystem to generate records or logs [11].

These interconnected systems and components greatly expand the attack surface of vehicles, increasing the vulnerability of the Connected and Autonomous Vehicle (CAV) to malicious attacks. The car industry is facing new cybersecurity challenges because of the advances in automation and connection in CAVs. Every enabling technology has the potential to develop into a new area of attack, and there will be a rise in the number of cybersecurity events involving wireless networks [12].

In the cascading failure study, a series of overload failures could result from the failure of a single node as the flow rates are constant. In this scenario, a reliable indicator of network performance is the size of the largest connected component. However, in our case, to prevent cascade failures, the best bandwidth allocation method reallocates the flow rates. As performance metrics, we employ the bandwidth usage ratio and total utility [13].

A new honeypot approach sets itself apart from traditional strategies that require holding actual crypto assets or falsifying balance information to deceive attackers. In contrast to conventional honeypots, our method associates legitimate third-party high-balance wallet addresses, even though we don't own any crypto assets. Using this technique, we illustrated how appealing such honeypots are to attackers. By grouping in a space according to how frequently the attacker executes each method, we suggested a new analysis technique for attackers. Using this technique, we showed that we could categorize attackers inside large attack logs based on traits, actions, and goals [14].

A framework to set up a programmable, physically interactive honeypot was suggested by you and your colleagues. Using its one-to-many techniques, high interaction capabilities, and cross-network access, the honeypot may mimic a programmable logic controller and collect extensive attack data. The authors have developed an operational scenario to set up the honeypot and obtain affordable attack data. Additionally, honeypots are employed in connected vehicle networks for intrusion detection. They concentrate on determining how many, where, and what kind of honeypots to place throughout the network. Honey Car expands the scope of honeypot applications by emphasizing the tactical reorientation of honeypots to trick adversaries and gather threat intelligence on the Internet of Vehicles [15].

Professionals contribute in the collecting and analysis of data to generate the OWASP Top 10 list. The OWASP Top 10 Cheat Sheet series provides developers and testers with helpful guidance on mitigating the Top 10 security vulnerabilities identified by OWASP. By actively safeguarding their web apps and putting the cheat sheets' instructions into practice, organizations can improve their ability to avert security issues [16]. The goal of this study is to thoroughly assess how well OWASP recommendations and cheat sheets work to improve web application security throughout the Software Development Life Cycle (SDLC). This study also attempts to evaluate the OWASP cheat sheets' efficacy on an enterprise-level online web application to ascertain whether additional security precautions are required.

In summary, including Cyber Canary systems in network architecture improves data security and operational integrity by offering critical early alerts against possible attacks. This proactive strategy greatly strengthens cybersecurity defences, The setup could also be implemented with multiple canaries in a network combined with OWASP top ten and the alerting tools like *rsyslog* utility as it is easy to implement and manage, and flexible to any changes. Organizations may build a safer digital environment, preserve trust, and proactively address threats by utilizing these technologies and encouraging a culture of alertness.

# Chapter 3: Problem Statement

In the field of cybersecurity, organizations consistently strive to identify and address cyber threats. Conventional honeypots are useful in theory, but they have several serious drawbacks that make them less useful in real life.

*Traditional honeypots pose significant challenges due to their high complexity, cost, and maintenance demands, making them difficult for businesses with limited cybersecurity budgets. They require specialized staff for attack analysis, leading to potential response delays. Response times might make organizations more susceptible to emerging threats. Additionally, their typical deployment at network perimeters limits their effectiveness in monitoring internal threats and advanced cyber-attacks. There is also the challenge of mapping the attacks with real-time vulnerabilities so that the attack can be identified at an earlier stage to protect the network.*

# Chapter 4: Objectives of the Study

The main goal is to find out how effective Cyber Canary is at reducing cyber threats and improving network security. Its objective is to assess Cyber Canary's detection capabilities with a focus on speed and accuracy in identifying cyber threats such as malware, illegal access, and unusual behaviors. Cyber Canary and conventional honeypots will be compared to determine their advantages and disadvantages. Additionally, the study looks at how easily Cyber Canary can be deployed across cloud environments, critical infrastructure, and internal subnets, evaluating how well it can adapt to different network configurations.

These are the objectives of the proposed solution to the problem statement:

1. Develop an adaptable Network Setup for Canary using VM.
2. Map the canary network with real-time OWASP vulnerabilities; three controls selected for this project are Broken Access Control, Injection, and Security Misconfiguration.
3. Implement intelligent signaling mechanisms within the Network Canary system.
4. Integrate the alert signaling with the centralized server for better monitoring.

# Chapter 5: Project Methodology

The Cyber Canary project's methodology entails a methodical approach to network environment setup, the deployment of specially designed attack-simulating machines, and the analysis of Cyber Canary's reaction to identify possible threats. As shown in Fig. 5.1 Cyber canary is outlined in the steps that follow how to apply it to improve network security.

Fig. 5.1: Proposed Methodology for Cyber Canary

**Network Setup and Configuration**

The project begins with the creation of three Ubuntu VMs that are part of the network configuration, set up as canaries to identify and record anomalous activity and security breaches. Every canary VM keeps an eye out for possible dangers and signals any illegal access or attacks in advance. The canary VMs' logs are gathered and examined in real-time by a central logging server running on a Linux virtual machine.

**Deployment of Customized Vulnerable Machines**

The network configuration consists of three often occurring OWASP vulnerabilities mapped to susceptible Ubuntu virtual machines File Transfer Protocol (FTP), Light Weight Directory Access Protocol (LDAP), and MySQL. serving as canaries in the sense that they each monitor and record questionable activity to identify security issues. For real-time alerting and analysis, a centralized server VM is set up to accept and process the logs from all three VMs.

**Simulation of Attacks**

To evaluate CyberCanary detection capabilities, simulated attacks are launched against vulnerable VMs within the network environments as shown in Fig. 5.2. These attacks may include but are not limited to:

1. **LDAP Injection:** Test for vulnerabilities by injecting malicious LDAP queries to access or manipulate data.
2. **FTP Brute Force:** Simulate brute force attacks to check if the system enforces account lockout policies.
3. **SQL Injection:** Execute SQL injection attacks to exploit vulnerabilities in MySQL queries.
4. **Sensitive Data Exposure:** Test scenarios where sensitive information is exposed due to misconfigurations.
5. **Access Control Misconfigurations:** Attempt to access restricted data to assess the effectiveness of access controls.



Fig. 5.2: Attack Simulation [17]

**Detection by CyberCanary**

CyberCanary devices are intended to be strategically positioned decoy systems within a network, they are a perfect solution for early threat detection. Because CyberCanary is not used for ordinary business purposes, unlike regular machines, they are extremely good at warning administrators about unauthorized access. Security professionals are alerted immediately of any potential threats or breaches when a CyberCanary is accessed. This method adds a proactive degree of protection by guaranteeing that any interaction with these spoof systems is suspect and calls for additional inquiry. Their presence in a network serves as a tripwire, enabling quick identification and elimination of harmful activity before it gets out of hand.

**Centralized Log Forwarding**

Setting up every VM to send its logs to a central server is known as centralized log forwarding. This is accomplished by configuring log forwarding agent *rsyslog* on every system, which gather and send log data to the centralized logging server in real-time. These logs are collected and stored by the central server, allowing for centralized monitoring and analysis. This methodology guarantees improved network-wide security monitoring, expedited analysis, and effective log management.

To sum up, the Cyber Canary project's methodical approach includes putting up simulated network environments and deploying susceptible machines, successfully mimics and countering a range of cyber threats. Cyber Canary detects possible intrusions and offers insightful information for strengthening network defense through thorough attack simulations and reliable detection algorithms. Cyber Canary is continuously evaluated and improved to ensure that it remains a valuable tool for enhancing network security and fending off emerging attacks.

# [Chapter 6: Resource Requirement Specification](#_Toc165641783)

In this project, Three VMs are deployed as vulnerable machines with one Kali VM to collect all the logs from the vulnerable machines. To ensure optimal performance and stability, the following resource requirements are recommended:

**Physical Computer**

1. **CPU:** At least a quad-core processor (e.g., Intel i5 or higher) to efficiently handle virtualization and processing demands.
2. **RAM:** Minimum of 16 GB to allocate sufficient memory to both VMs and the host operating system.
3. **Storage:** At least 256 GB SSD for fast access speeds and to support the storage needs of both VMs, with a recommended 50 GB disk space per VM.

**Linux VMs**

1. **CPU Allocation:** Each VM should be allocated 2 virtual CPUs to ensure responsive performance.
2. **RAM Allocation:** Allocate at least 4 GB of RAM per VM for running applications and maintaining system stability.
3. **Disk Space:** Allocate 20 GB of disk space per VM to accommodate the operating system, application data, and logs.

**Networking**

A reliable network connection (preferably 1 Gbps) for both the host machine and VMs to facilitate smooth communication and data transfer.

# Chapter 7: Software Design

The proposed setup hosts several intentionally vulnerable VMs in the environment as shown in Fig. 7.1. The VMs have known security flaws. As a centralized server, a different VM oversees the gathering and reviewing of logs produced by the susceptible VMs.

**The Architecture of Canary**

This configuration makes it easier to conduct controlled vulnerability testing since it gives us a secure, isolated environment to watch and examine attacker activity. The system architecture is shown in the graphic that goes with this section. It shows the VMs that are made vulnerable as well as how log data is sent to the centralized server with the help of the Linux *rsyslog* services.

A diagram of a computer network

Description automatically generated

Fig. 7.1: Network Canary Architecture

As shown in Fig. 7.2, illustrates the complete working of the Cyber Canary from hosting vulnerable VMs, and segregating the logs based on the respective attack, and those filtered logs are forwarded to the Centralized Server, the logs can be seen, and required measures can be taken.

A diagram of a computer network

Description automatically generated

Fig. 7.2: Cyber Canary Proposed Solution

1. Centralized Server: This is the Server with the *rsyslog* server setup to receive the logs from the VMs in the network. A centralized server is used to aggregate data, logs, and configurations from multiple systems for streamlined management and monitoring. It enhances security, simplifies administration, and improves network performance by providing a single point for data analysis and alerting.
2. LDAP VM: An LDAP-vulnerable VM hosted as a canary in the network serves as a decoy to detect unauthorized access or exploitation attempts. It triggers alerts when attackers interact with it, helping to identify security threats in real-time.
3. FTP VM: An FTP-vulnerable VM hosted as a canary in the network acts as bait to lure attackers into exploiting weaknesses. It generates alerts upon suspicious activity, helping to detect potential security breaches early.
4. SQL VM: An SQL-vulnerable VM hosted as a canary in the network simulates exploitable database weaknesses to attract attackers. It triggers alerts when malicious SQL activities are detected, enabling early identification of potential threats.

# Chapter 8: Implementation

Implementing the proposed problem requires some basic hardware and software with minimal cost that can set up the canary to help any network, identify the threat, and hence alert the system admin or the users of the infra to secure their network.

**Network Setup and Configuration**

Download and install a hypervisor like VirtualBox or VMware Workstation Get the Ubuntu and Kali Linux ISO or pre-built virtual machine image from the official website, make a new VM, choose the ISO or VM image as the source, and then set up the CPU, memory, and storage. Proceed with the installer's instructions to finish the setup.

To configure *rsyslog*, install *rsyslog* on the VM and edit the *rsyslog.conf* file to define log sources, filters, and destinations. This setup enables centralized log collection and forwarding for enhanced monitoring and analysis in virtual environments. The *rsyslog* configuration file for the client VM is shown in Fig. 8.1followed by the *rsyslog* server configuration in Fig. 8.2 and 8.3

1. ***rsyslog* Client Configuration on VMs**

****

Fig. 8.1: *rsyslog* Client File

1. ***rsyslog* Server Configuration on the Server VM**

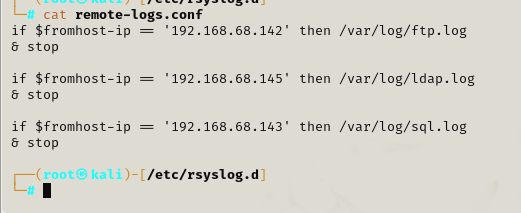


Fig. 8.2: Segregation of Log Files

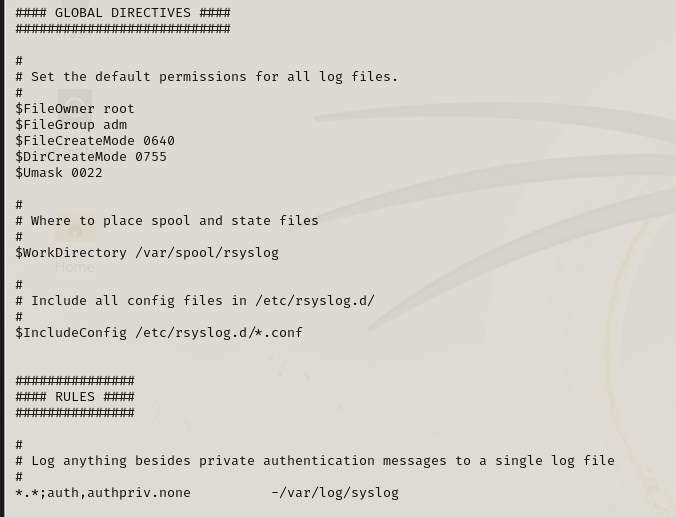


Fig. 8.3: rsyslog Server File

**Deployment of Vulnerable Machines**

As a proactive security approach, the vulnerable SQL, LDAP, and FTP virtual machines are deployed as canaries on the network. To mimic actual attack surfaces, each VM is set up with deliberate vulnerabilities mapped to OWASP Broken Access Control (A01), Injection (A03), and Security Misconfiguration (A05). The purpose of these canaries is to draw attention to and identify malicious activities. The logs are sent to a central server for further examination and oversight. This configuration facilitates early detection of any breaches and improves threat detection.

**Simulation of Attack**

The *ldapmodify* command, in this instance, modifies the LDAP user's common name (cn) attribute as shown in Fig. 8.4. The change in (cn) value for the user indicated by the given DN to "New John Doe" using administrative credentials. The user's display name has changed, and this is confirmed by the successful modification.

**A screenshot of a computer

Description automatically generated**

Fig. 8.4: Simulation using LDAP

****

Fig. 8.5: Simulation using MySQL

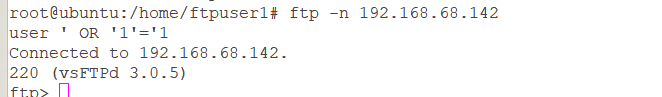


Fig. 8.6: Simulation using FTP

The attack simulation is carried out on FTP, LDAP, and MySQL with the selected controls of OWASP top ten. The simulation can be seen in Fig. 8.4 and Fig. 8.5 where the SQL injection is used to get the data and to log in to the FTP server.

The simulation was successfully done on each canary machine and the next chapter has the Testing of these machines.

# Chapter 9: Testing and validation

This chapter covers the testing and validation procedures used to evaluate the security posture of systems that are deemed susceptible, with an emphasis on LDAP, MySQL, and FTP services. Extensive testing is carried out to detect possible security flaws.

**LDAP Vulnerable Machine**

1. **Broken Access Control:** The command *ldapsearch -x -LLL -b "dc=example,dc=com" "(uid=\*)"* allows users to query any data by bypassing search filter restrictions. It retrieves all entries in the directory with the `uid` attribute under the specified base DN *(`dc=example,dc=com`).* The logs are seen with the mentioned search filters as shown in Fig. 9.1.

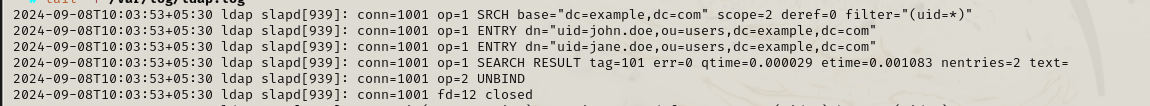


Fig. 9.1: LDAP Broken Access Control Log

1. **Injection:** Using LDAP to Extract Data By injecting wildcard characters or extra filter criteria, attackers can manipulate input and retrieve unauthorized user data. As shown in Fig.9.2. If *ldapsearch -x -LLL -H ldap://localhost -b "dc=example,dc=com" "(uid=\*)(objectClass=inetOrgPerson)"`* is not properly sanitized, it can leak sensitive information about all users. The generated logs are shown in Fig. 9.3.

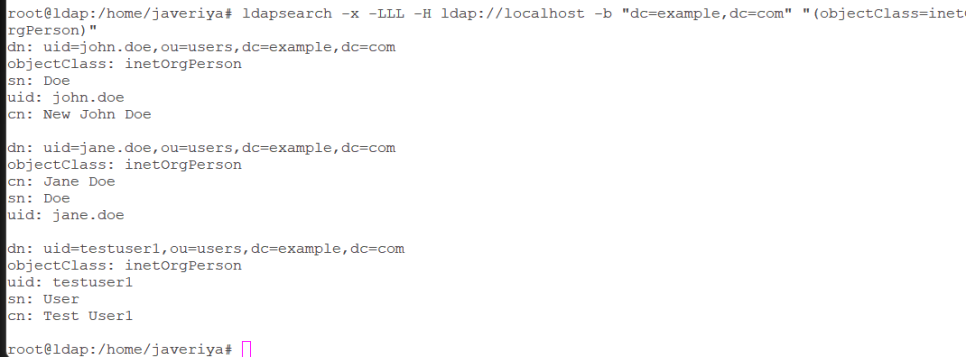
****

Fig. 9.2: Injection Using LDAP

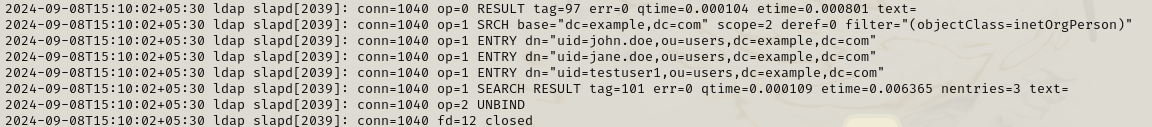
****

Fig. 9.3: Logs for LDAP Injection

1. **Security Misconfiguration:** To access restricted properties in directory replies, including passwords or personal information, attackers must inject extra filter parameters. This technique is known as "exposing sensitive attributes." Misconfigured directory permissions may unintentionally expose critical data. The command *`ldapsearch -x -LLL -H ldap://localhost -b "dc=example,dc=com" "(uid=testuser1)"}* can be used to illustrate this as shown in Fig. 9.4 with the log at the centralized server is shown in Fig. 9.5.

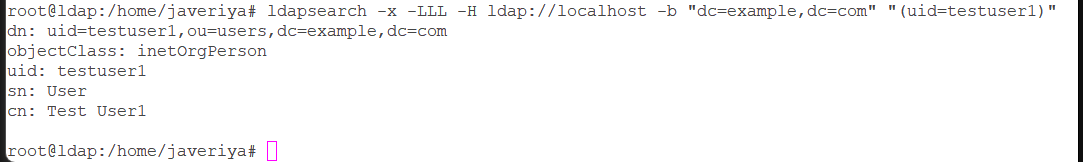


Fig. 9.4: LDAP Security Misconfiguration

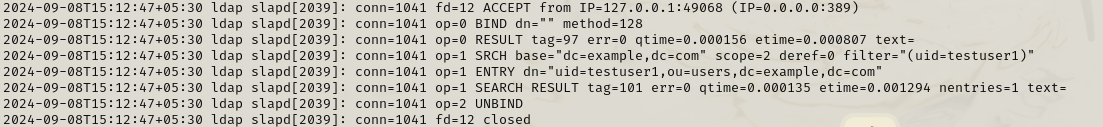


Fig. 9.5: Logs for LDAP Security Misconfiguration

**FTP Vulnerable Machine**

1. **Broken Access Control:** Exposed Sensitive Information in FTP Listings is the result of failing access control, which exposes sensitive data to all users. Incorrect permission settings, like *chmod 644,* may cause this by enabling 78unauthorized users to read files in FTP directory listings as shown in Fig. 9.6.



Fig. 9.6: FTP Broken Access

1. **Injection:** When user inputs are not adequately sanitized, bad actors can insert SQL queries into FTP commands, leading to SQL Injection on an FTP server. Attackers might use this to alter the database by sending out numerous queries in a single request. An attacker may circumvent authentication and destroy tables, for instance, by entering (' OR '1'='1') as shown in Fig. 9.7 in a susceptible FTP login form. Inadequate security can allow these assaults to compromise confidential information or even bring down the site these logs could be viewed through a centralized server as shown in Fig. 9.8.

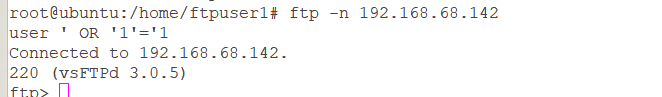


Fig. 9.7: SQL Injection on FTP Server

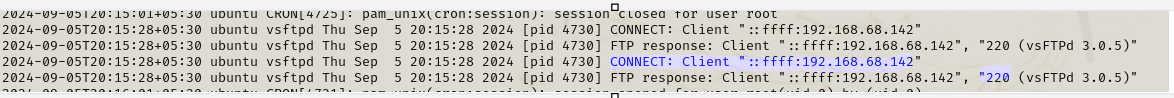
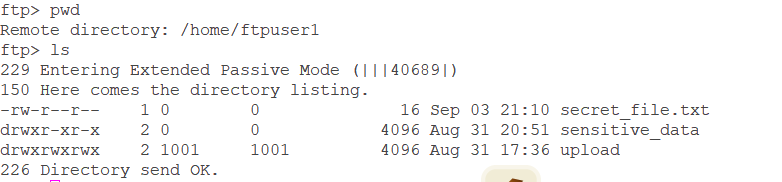
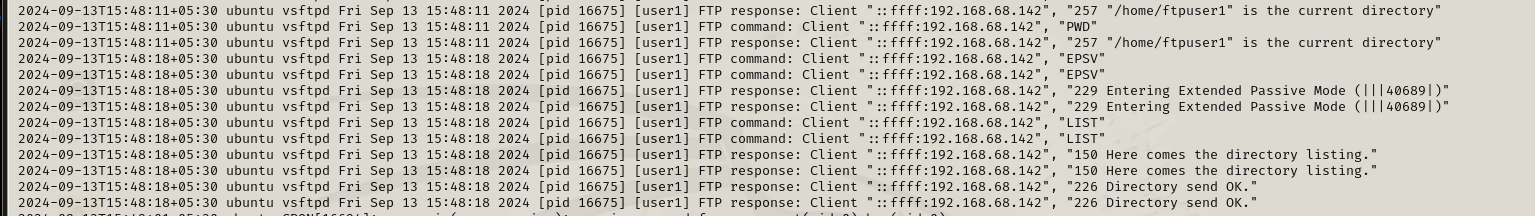


Fig. 9.8: Log for SQL injection on FTP

1. **Security Misconfiguration:** If permissions are not correctly established, a security misconfiguration on an FTP server can result in unfettered access to sensitive files. Sensitive data exposure may happen, for instance, if file permissions permit unauthorized users (like anonymous FTP users) to see or alter files like /home/ftpuser1/secret\_file.txt as shown in Fig. 9.9. Restricted access restrictions and appropriate directory and file permissions should be set up to restrict exposure to authorized users exclusively to avoid this the alert would also notify the traversal to sensitive file path as shown in Fig. 9.10 through the centralized logging method.

****Fig. 9.9: Ftp Security Misconfiguration

****Fig. 9.10: Log for Security Misconfiguration on FTP

**MYSQL Vulnerable Machine**

**1. Broken Access Control:** In a MySQL environment, Broken Access Control leading to horizontal privilege escalation occurs when a user is granted more privileges than necessary As Shown in Fig. 9.11, allowing them to access data or functions meant for other users. For example, a user with excessive permissions could access sensitive information or perform unauthorized actions. Implementing the principle of least privilege and enforcing strict access control mechanisms can mitigate these risks. Otherwise, the sensitive data would be seen as shown in Fig. 9.12 as the user has complete access to the database.

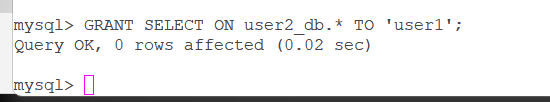
****

Fig. 9.11: Broken Access for MySQL

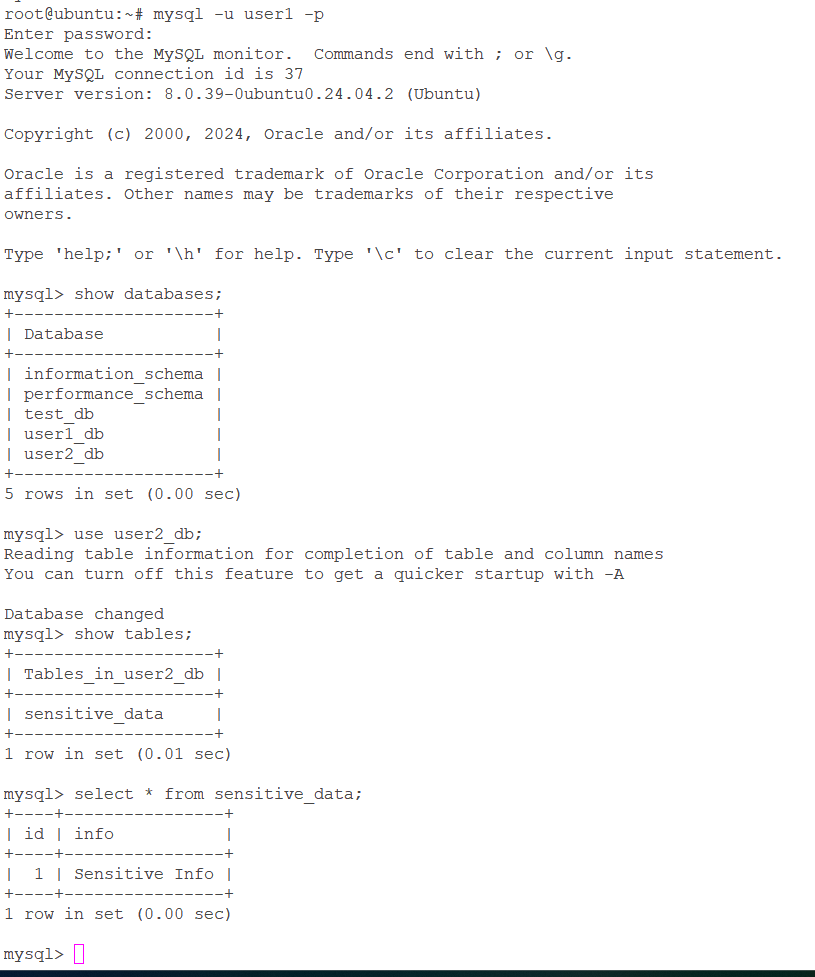
****

Fig. 9.12: Broken Access causes Sensitive Data Exposure.

1. **Injection:** An attacker might obtain unauthorized access or modify sensitive data by manipulating input data to execute undesired commands on a database through injection attacks, such as Blind SQL injection as shown in Fig. 9.13. An attacker could use inadequately sanitized inputs in a MySQL environment to run malicious SQL queries. By immediately recording and passing suspicious queries and inputs to a centralized server for processing, centralized logging can assist in the detection of such attacks as shown in Fig. 9.14. This improves security monitoring and incident response by enabling quick identification and handling of possible injection attempts.

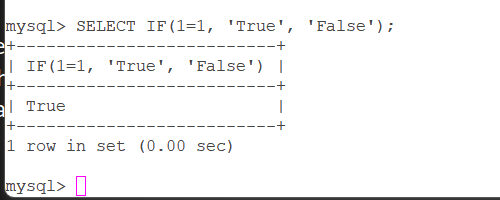
****

Fig. 9.13: Blind MySQL Injection

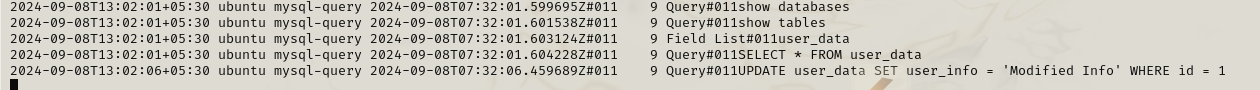
****

Fig. 9.14: Log of MySQL Injection

1. **Security Misconfiguration:** Backup files that hold private information, like a MySQL database backup, are susceptible to unwanted access if they are incorrectly stored or have inadequate permissions. For instance, attackers can readily extract data if a backup file, such as */var/www/html/exposed\_backup.sql*  is publicly available through a web server. These backups are exposed due to security misconfigurations, such as the use of weak permissions ls -l /var/www/html/ as shown in Fig. 9.15 To prevent unwanted access to important backups, safe storage procedures, and appropriate file permissions should be put in place.

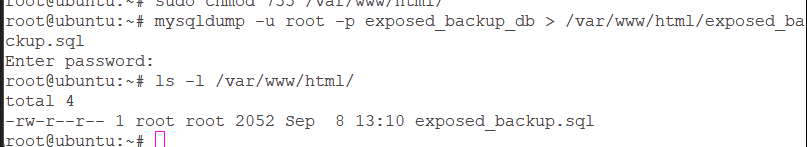


Fig. 9.15: Security Misconfiguration on MySQL

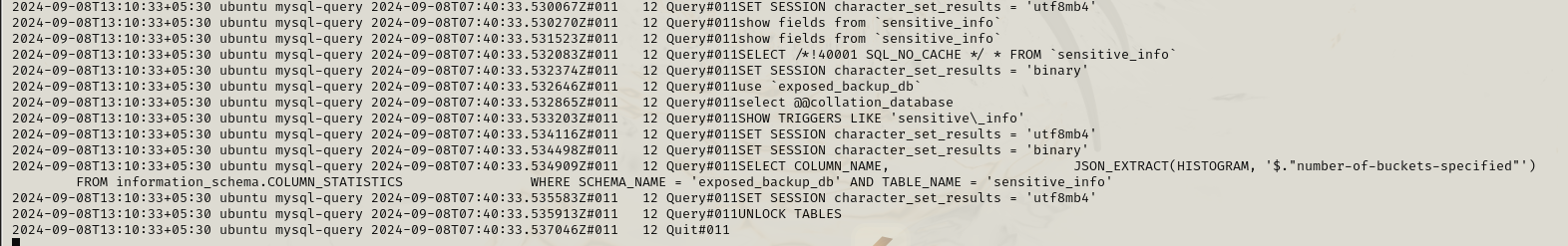


Fig. 9.16: Logs Security Misconfiguration on MySQL

If the logging settings are not carefully controlled, an improperly setup MySQL database may unintentionally reveal sensitive data as shown in Fig. 9.16, including user credentials and personal information. Inadequate MySQL configuration could expose important information that hackers could use against you. To protect sensitive data from unwanted access and security breaches, it is imperative to maintain appropriate log management and security procedures.

In conclusion, all VMs have been successfully configured to send logs to the centralized server, ensuring comprehensive logging and monitoring. The centralized setup is functioning as expected, with logs properly received and segregated for each VM. This confirms that the infrastructure is effectively capturing and centralizing security events, supporting a robust monitoring and alerting system.

# Chapter 10: Analysis and Results

This chapter presents the analysis of Centralized Server logs. The vulnerabilities of MYSQL, FTP, LDAP mapped with OWASP Broken Access Control, Injection and Security Misconfiguration alerts could be seen as shown in Fig. 10.1 All the logs from these vulnerable machines are forwarded to this VM and stored in separate files under */var/log* to view the logs.

The live logs are captured using the command *tail -f /var/log/ftp.log* to check the logs from the FTP vulnerable machine.

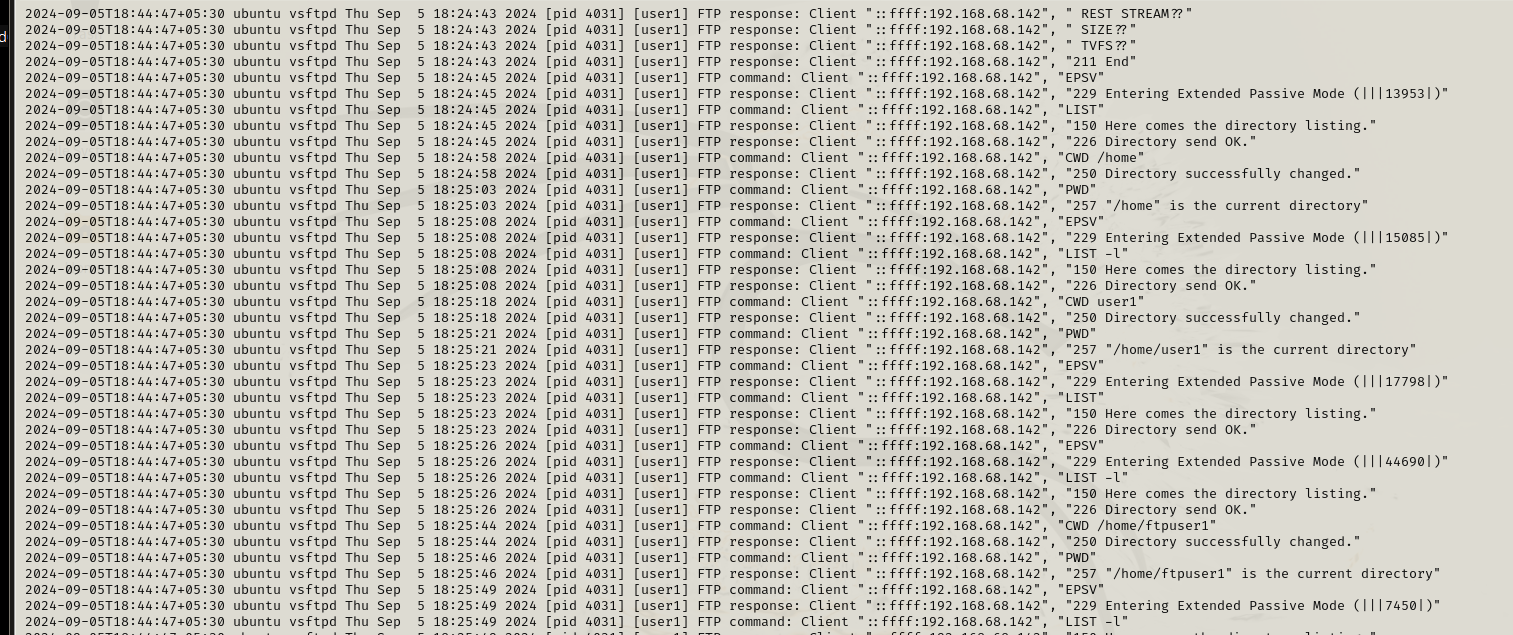
**

Fig. 10.1: FTP Logs

The live logs are captured using the command *tail -f /var/log/ldap.log* to check the logs from the LDAP vulnerable machine as shown in Fig. 10.2.



Fig. 10.2: LDAP Logs

The live logs are captured using the command *tail -f /var/log/mysql.log* to check the logs from the SQL vulnerable machine as shown in Fig. 10.3.

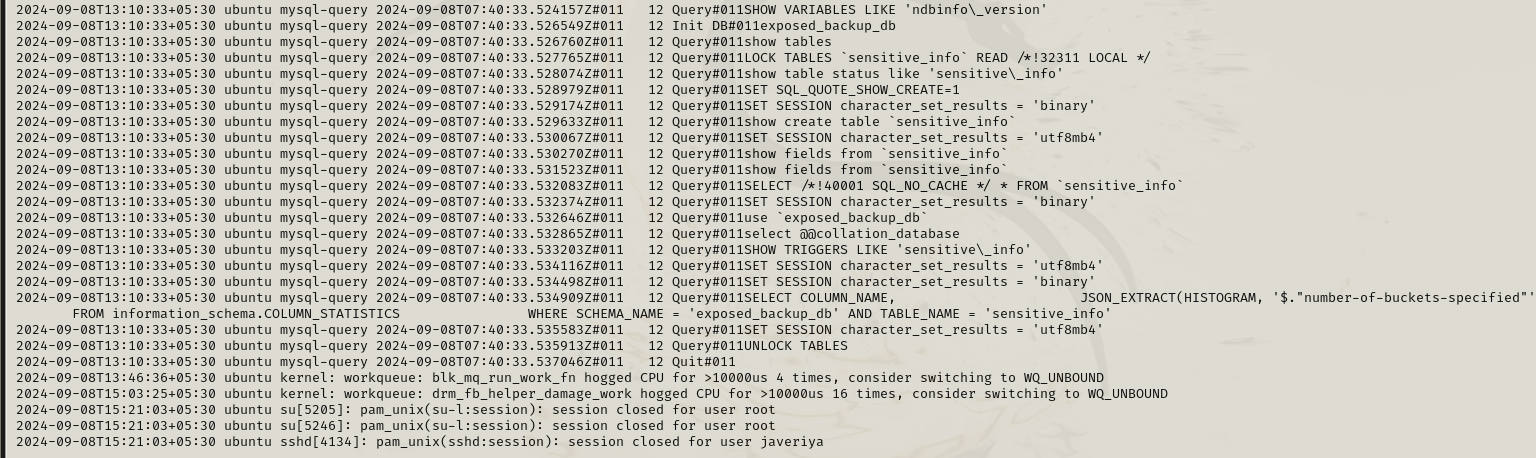


Fig. 10.3: SQL Logs

For effective security monitoring and incident response, a centralized server housing log files is essential. It makes it possible to compile all logs from different VMs in one place or can also be stored in separate files in the server to have more insights on each machine's data as shown in Fig.10.3 making it simpler to correlate events across the network. By increasing visibility, this centralized method makes it simpler to identify anomalies like injection assaults or illegal access. Furthermore, it guarantees that logs are offloaded from the source system and stored safely, free from manipulation

To sum up, every virtual machine has been effectively set up to transmit logs to the central server, guaranteeing thorough logging and observation. The centralized configuration is operating as planned, and each virtual machine's logs are correctly received and divided.

# Chapter 11: Conclusions and Future Scope

The Cyber Canary initiative introduces a novel approach to protect network environments that involves the inclusion of intentional vulnerabilities into running VMs along with the ability to generate logs when any action takes place inside the network. It does this by utilizing the *rsyslog* server and client installed in the VMs to monitor and forward the logs.

**Key Achievements of Cyber Canary Include:**

1. **OWASP A01 (Broken Access Control):** By setting off alarms when access control vulnerabilities were exploited, this demonstration showed how to detect illegal access on LDAP and FTP canaries.
2. **OWASP A03 (Injection):** Captured and logged malicious injection attempts to the centralized server, this model simulates SQL injection attacks on the SQL canary
3. **OWASP A05 (Security Misconfiguration):** Highlighted exposed sensitive data and warned of attempted exploitation by attackers by showcasing security misconfigurations on FTP and SQL canaries.
4. **Real-time Threat Detection:** By using canaries to warn the centralized server of attempted exploitation, early detection and reaction to such attacks were made possible, enhancing network security.

**Future Scope**

To further improve detection and response capabilities, this Cyber Canary architecture will eventually be mapped with the MITRE ATT&CK framework. Attack patterns including privilege escalation, lateral movement, and credential access can be more precisely identified by matching the FTP, LDAP, and SQL canaries with certain MITRE tactics and approaches. This integration would make it possible to comprehend hostile conduct more thoroughly, enhance threat-hunting procedures, and make it easier to create proactive defense plans that are based on actual attack situations.

Canary can also be embedded into hardware. Furthermore, by arranging these canaries into a mesh network and allowing them to interact with one another, for the overall security of the network.

# Bibliography

[1] P. Chauhan, S. Mathew, M. Osterman, and M. Pecht, “In situ interconnect failure prediction using canaries,” *IEEE Transactions on Device and Materials Reliability*, vol. 14, no. 3, pp. 826–832, Sep. 2014, doi: 10.1109/TDMR.2014.2326184.

[2] A. Obied, “Honeypots and Spam,” 2006. Accessed: Jul. 19, 2024. [Online]. Available: https://ahmed.obied.net/research/papers/honeypots\_spam.pdf

[3] I. Koniaris, G. Papadimitriou, P. Nicopolitidis, and M. Obaidat, “Honeypots deployment for the analysis and visualization of malware activity and malicious connections,” in *2014 IEEE International Conference on Communications (ICC)*, Sydney, NSW, Australia: IEEE, Jun. 2014, pp. 1819–1824. doi: 10.1109/ICC.2014.6883587.

[4] N. C. Brintha, V. V. Joliya, G. Bhuvnesh, and S. Malini, “Securing your network with Honeypot, Canerytokens and Docker on AWS,” in *Proceedings of International Conference on Computational Intelligence and Sustainable Engineering Solution, CISES 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 683–687. doi: 10.1109/CISES58720.2023.10183431.

[5] A. Malhotra, A. Elsayed, R. Torres, and S. Venkatraman, “Evaluate Canary Deployment Techniques Using Kubernetes, Istio, and Liquibase for Cloud Native Enterprise Applications to Achieve Zero Downtime for Continuous Deployments,” *IEEE Access*, vol. 12, pp. 87883–87899, Jun. 2024, doi: 10.1109/ACCESS.2024.3416087.

[6] T. Ünlü, L. A. Shepherd, N. Coull, and C. McLean, “Poster: Angry birding: Evaluating application exceptions as attack canaries,” in *Proceedings - 2021 IEEE European Symposium on Security and Privacy, Euro S and P 2021*, Institute of Electrical and Electronics Engineers Inc., Sep. 2021, pp. 701–703. doi: 10.1109/EuroSP51992.2021.00052.

[7] Institute of Electrical and Electronics Engineers, IEEE Reliability Society, Tex. IEEE Conference on Prognostics and Health Management 2015.06.22-25 Austin, Tex. IEEE International Conference on Prognostics and Health Management 2015.06.22-25 Austin, and Tex. IEEE PHM 2015.06.22-25 Austin, *2015 IEEE Conference on Prognostics and Health Management (PHM) 22-25 June 2015, Austin, Texas*.

[8] W. Wang and M. Pecht, “Economic analysis of canary-based prognostics and health management,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 7, pp. 3077–3089, Jul. 2011, doi: 10.1109/TIE.2010.2072897.

[9] W. Howard and M. Borowczak, “Detecting DDoS Attacks near the Edge with Router Canaries,” in *IEEE International Conference on Consumer Electronics - Berlin, ICCE-Berlin*, IEEE Computer Society, Nov. 2020. doi: 10.1109/ICCE-Berlin50680.2020.9352164.

[10] Timothy K. Buennemeyer, “Battery-Sensing\_Intrusion\_Protection\_System,” *IEEE Workshop on Information Assurance United States Military Academy, West Point, NY*, pp. 1–8, 2006.

[11] M. Fuentes-Garcia, J. Camacho, and G. Macia-Fernandez, “Present and Future of Network Security Monitoring,” *IEEE Access*, vol. 9, pp. 112744–112760, 2021, doi: 10.1109/ACCESS.2021.3067106.

[12] Y. Wang, B. Yu, H. Yu, L. Xiao, H. Ji, and Y. Zhao, “Automotive Cybersecurity Vulnerability Assessment Using the Common Vulnerability Scoring System and Bayesian Network Model,” *IEEE Syst J*, vol. 17, no. 2, pp. 2880–2891, Jun. 2023, doi: 10.1109/JSYST.2022.3230097.

[13] Y. Xia and D. J. Hill, “Attack vulnerability of complex communication networks,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 1, pp. 65–69, 2008, doi: 10.1109/TCSII.2007.908954.

[14] H. Uchibori, K. Yoshioka, and K. Omote, “Honeypot Method to Lure Attackers Without Holding Crypto-Assets,” *IEEE Access*, vol. 12, pp. 16059–16071, 2024, doi: 10.1109/ACCESS.2024.3357785.

[15] S. Panda, S. Rass, S. Moschoyiannis, K. Liang, G. Loukas, and E. Panaousis, “HoneyCar: A Framework to Configure Honeypot Vulnerabilities on the Internet of Vehicles,” *IEEE Access*, vol. 10, pp. 104671–104685, 2022, doi: 10.1109/ACCESS.2022.3210117.

[16] A. Khanum, S. Qadir, and S. Jehan, “OWASP-Based Assessment of Web Application Security,” in *2023 18th International Conference on Emerging Technologies (ICET)*, Peshawar : IEEE, Nov. 2023, pp. 240–245. doi: 10.1109/ICET59753.2023.10374730.

[17] https://deepai.org/machine-learning-model/text2img, “image creation wesbite text to image generator.”

# Appendix

## Plagiarism Report[[1]](#footnote-1)

**Plagiarism Report** with below 15% Similarly index to be attached in the annexure. The title page and last pages with the similarity index report are attached.

A screenshot of a cell phone

Description automatically generated

A screenshot of a computer

Description automatically generated

Gitlink

https://github.com/Javeriya123/Cyber-Canary

1. Turnitn report to be attached from the University. [↑](#footnote-ref-1)