Threat Intelligence Implementation Report

1. Introduction

This report details the implementation of threat intelligence principles, including the analysis of Indicators of Compromise (IoCs) and the setup and utilization of the OpenCTI Threat Intelligence Platform.

2. Analysis of Indicators of Compromise (IoCs)

For this project, I'll define two mock IoCs and describe potential detection methods and their threat implications.

IOC #1: Malicious IP Address

- **IoC Value:** 192.168.100.10
- **Type:** IP Address
- **Description:** This IP address is identified as a source of multiple failed login attempts against an organization's mail server within a short timeframe.
- **Detection Method:**
    - **Firewall Logs:** Analyzing firewall logs for excessive connection attempts or failed login attempts originating from this IP address.
    - **Intrusion Detection System (IDS):** Configuring an IDS to monitor network traffic for suspicious activity from this IP.
    - **Security Information and Event Management (SIEM) System:** Using a SIEM to aggregate logs from various sources and correlate events related to this IP.
- **Threat Indication:**
    - **Brute-Force Attack:** The failed login attempts suggest a brute-force attack attempting to gain unauthorized access to email accounts.
    - **Potential for Data Breach:** If the attack succeeds, it could lead to a data breach and compromise sensitive information.

IOC #2: Malicious File Hash

- **IoC Value:** 25624f3554e989cfb58b220c1742a0334766bb42c5b9f555652b83b3b6d2b6d
- **Type:** File Hash (SHA-256)
- **Description:** This file hash is associated with a file found on an employee's workstation. The file is disguised as a PDF document but contains an executable.

- **Detection Method:**
    - **Endpoint Detection and Response (EDR):** EDR systems can detect and analyze file hashes on endpoints.
    - **Antivirus Software:** Traditional antivirus solutions can also detect known malicious file hashes.
    - **File Integrity Monitoring (FIM):** FIM tools monitor changes to critical files and can alert on the presence of files with this hash.
- **Threat Indication:**
    - **Malware Infection:** The executable file poses a high risk of malware infection, which could lead to various malicious activities.
    - **Phishing Attack:** The disguise as a PDF suggests a phishing attack aimed at tricking the user into executing the file.

3. OpenCTI Platform Implementation

3.1 OpenCTI Setup

The provided document shows the use of Docker Compose for setting up OpenCTI. Here's a summary of the setup process:

- **Docker Compose Configuration:** The `docker-compose.yml` file defines the services for OpenCTI and its dependencies.

- **Services:** The configuration includes services like the OpenCTI platform itself and connectors.

- **Environment Variables:** Environment variables are used to configure the services, such as the OpenCTI URL and token.

- **Connectors:** The document highlights the use of connectors like the "CISA Known Exploited Vulnerabilities" connector and the "OpenCTI Datasets" connector.

3.2 Connector Integration

- **CISA Known Exploited Vulnerabilities Connector:** This connector imports data on known exploited vulnerabilities from the CISA catalog.
    - It is configured with parameters like the CISA catalog URL and update frequency.
- **OpenCTI Datasets Connector:** This connector imports various datasets, such as information on sectors, geography, and companies.
    - It is configured to update these datasets periodically.

3.3 Basic Usage Demonstration

- **Data Visualization:** The OpenCTI platform provides visualizations of threat data, such as most active threats and targeted victims.

- **Connector Status:** The platform displays the status and activity of connected workers and connectors.

- **Data Ingestion:** Connectors import data into the platform, enriching it with threat intelligence.

4. Documentation

- **Docker Compose File:** The `docker-compose.yml` file serves as the primary documentation for the OpenCTI setup.

- **Connector Configuration:** The environment variables within the Docker Compose file document the configuration of the connectors.
- **Platform Interface:** The OpenCTI interface provides insights into the platform's status, connected workers, and data ingestion.

5. Evidence of Functionality

- The document includes screenshots of the OpenCTI interface, showing connected workers and data visualizations.

- The `docker-compose.yml` file demonstrates the configuration and integration of connectors.

6. Conclusion

This report demonstrates the implementation of threat intelligence principles through the analysis of mock IoCs and the setup and utilization of the OpenCTI platform. The use of connectors enhances the platform's capabilities by integrating external threat intelligence feeds, providing valuable insights for threat detection and response.