Project Goal: To demonstrate understanding of threat intelligence through practical implementation, including IoC analysis and OpenCTI platform setup.

I. Analysis of Indicators of Compromise (IoCs)

For this mock scenario, let's fabricate two IoCs and analyze them.

IoC 1: Suspicious Domain

- **IoC Value:** `evilcorp.ltd`
- **Context:** Observed in web server logs as the referrer for multiple downloads of a recently released internal document. The document is sensitive and not meant for public distribution.
- **Detection Method:**
  - o **Web Server Log Analysis:** Parsing web server logs for referrer fields and identifying unusual or unknown domains. Tools like `grep`, `awk`, or specialized log analysis platforms (e.g., ELK stack) can be used. For example, using `grep` to find all log entries with "evilcorp.ltd" as the referrer.
  - o **DNS Monitoring:** Monitoring DNS queries within the network for resolutions of `evilcorp.ltd`. This can be done using tools like `tcpdump` or network intrusion detection systems (NIDS).
- **Threat Indication:**
  - o **Data Exfiltration:** The suspicious domain referring downloads of an internal document strongly suggests potential data exfiltration. Attackers often use domains they control to host stolen data or facilitate its transfer.
  - o **Phishing/Social Engineering:** It's possible the domain was used in a phishing campaign to trick employees into downloading the document, although the referrer log indicates direct downloads.
  - o **Command and Control (C2):** While less likely in this scenario, the domain could potentially be a C2 server for malware installed on an internal system.

IoC 2: Malicious File Hash

- **IoC Value:** `275a021bbfb6489e54d471899c7d4139230261c6903r2ba6fd099ec95a91` (This is a fabricated SHA-256 hash)
- **Context:** This hash is associated with an executable file found on an employee's workstation. The file is located in the user's temporary directory and has an unusual name.
- **Detection Method:**
  - o **Endpoint Detection and Response (EDR):** EDR solutions can monitor file creation and execution on endpoints and calculate file hashes. They can then compare these hashes against threat intelligence feeds or internal blacklists.
  - o **Antivirus/Anti-malware:** Traditional antivirus software also calculates file hashes and compares them to their databases of known malware.
  - o **File Integrity Monitoring (FIM):** FIM tools monitor changes to critical files and directories, including the creation of new files.

- **Threat Indication:**
    - **Malware Infection:** A file with an unknown hash, located in a suspicious location, is a strong indicator of malware infection. The malware could be anything from a virus or worm to ransomware or a backdoor.
    - **Compromised System:** The presence of malware indicates that the employee's workstation has likely been compromised, potentially allowing attackers to access sensitive data or use the system as a foothold for further attacks.

II. OpenCTI Platform Implementation

Let's assume we've chosen Docker for the OpenCTI installation.

A. Platform Setup and Connector Integration

1. **Docker Installation:**
    - Installed Docker and Docker Compose on a Linux server (e.g., Ubuntu). (Commands and screenshots would be included here, similar to the "docker-compose.yml" snippet in the original document, showing the installation process).
2. **OpenCTI Docker Deployment:**
    - Obtained the OpenCTI Docker Compose file (e.g., from the official OpenCTI GitHub repository).
    - Configured the Docker Compose file with necessary environment variables (database credentials, etc.).
    - Executed `docker-compose up -d` to start the OpenCTI platform. (Screenshots of the terminal output would be included).
3. **Connector 1: MISP Connector**
    - Installed the MISP connector using Docker or by manually placing it in the appropriate OpenCTI connector directory.
    - Configured the MISP connector with the MISP instance URL and API key.
    - Enabled the MISP connector in the OpenCTI platform.
    - Configured the connector to import threat intelligence data (e.g., malware hashes, threat actor information) from the MISP instance into OpenCTI. (Screenshots of the OpenCTI connector configuration interface would be included).
4. **Connector 2: VirusTotal Connector**
    - Installed the VirusTotal connector.
    - Configured the VirusTotal connector with the VirusTotal API key.
    - Enabled the VirusTotal connector.
    - Configured the connector to retrieve information about files (e.g., reputation, associated malware) based on their hashes. This would enrich the IoC data within OpenCTI. (Screenshots of the VirusTotal connector configuration would be shown).

B. Documentation of Platform Setup and Connector Integration

- A detailed document would be created outlining each step of the installation and configuration process.
- This document would include:
  - System requirements (OS, hardware).
  - Software dependencies (Docker, Docker Compose).
  - Step-by-step instructions with commands and screenshots.
  - Configuration details for OpenCTI and the connectors.
  - Troubleshooting tips.

C. Basic Usage Demonstration

1. **Importing IoCs:**
   - Demonstrated importing the IoCs (the suspicious domain and the file hash) into the OpenCTI platform. This could be done manually or through the MISP connector if MISP contained these IoCs. (Screenshots showing IoC creation in OpenCTI).
2. **Enriching IoCs:**
   - Showed how the VirusTotal connector automatically enriched the file hash IoC with information from VirusTotal (e.g., malware family, detection ratio). (Screenshots of the enriched IoC in OpenCTI).
3. **Creating an Investigation:**
   - Created an investigation in OpenCTI to link the two IoCs and document the analysis.
   - Added observations, relationships, and indicators to the investigation. (Screenshots of the investigation in OpenCTI).
4. **Visualization:**
   - Demonstrated OpenCTI's visualization capabilities to show the connections between the IoCs, related threats, and potential impact. (Screenshots of graphs or charts in OpenCTI).

III. Evidence of Functionality

- All steps would be documented with screenshots and terminal outputs.
- The OpenCTI platform interface would be shown, demonstrating the imported IoCs, connector configurations, and investigation details (similar to the dashboard screenshot in the original document).
- Logs from the connectors would be included to show successful data exchange.

This mock scenario provides a more complete and detailed implementation of threat intelligence principles using OpenCTI, addressing the key requirements of IoC analysis, platform setup, connector integration, and documentation.