Guardian SIEM - The Complete Guide to an Intelligent SOC Platform

Chapter 1: Project Overview and Core Philosophy (Why Guardian SIEM?)

1.1 Project Goal and Intelligence

Guardian SIEM is built as a proof-of-concept to demonstrate modern security operations capabilities: **Collection, Asynchronous Enrichment, Correlation, and AI-Driven Response (SOAR).**

How your SIEM is Superior (Comparison):

Feature	Guardian SIEM (Your Project)	Traditional SIEM (Splunk/Wazuh)
AI Role	**Decision Maker/Action Engine.** Uses Gemini for natural language SOAR commands and complex analysis.	Primarily limited to reporting and dashboard visualization.
Scalability Solution	**Asynchronous Background Queues (Python queue module).** Prevents the server from crashing when thousands of logs arrive simultaneously (fixes the ConnectionResetError 10054).	Heavy systems use dedicated message brokers (Kafka) and clustered databases (Elasticsearch).
Threat Enrichment	Instant lookup of GeoIP, DNS, and AbuseIPDB upon receiving logs.	Batch enrichment only.
Ease of Use	Simple Python scripts and	Requires complex

	single-machine deployment.	infrastructure and dedicated configuration files (conf files, modules).

Chapter 2: Setup, Installation, and API Keys (The A-Z Guide)

2.1 Software and Environment Requirements

1. **Python 3.10+:** Ensure it is installed on your Windows machine.
2. **VMware Workstation Player:** Required for running the MikroTik CHR virtual router.
3. **Npcap:** Required by Scapy for network sniffing (installed with Wireshark).
4. **Virtual Environment (VENV):** You must work inside an activated venv to manage project dependencies.
5. **Required Python Libraries:**
   pip install flask requests scapy google-generativeai pysnmp pywin32

2.2 API Key Configuration and Acquisition

These keys are essential for enrichment and AI capabilities and must be entered into your soc_dashboard.py file.

Service	Key Location in Code	How to Obtain the Key
Google Gemini AI	GEMINI_API_KEY = "AIzaSyBigGgQ50k6eVuHDT-VRWTVaECg8e-OQUU"	Get this key from the Google AI Studio Dashboard .
AbuseIPDB	ABUSEIPDB_API_KEY = "0512eb3e..."	Create a free account on AbuseIPDB and generate an API key from the "API" section of your profile.

2.3 MikroTik CHR (SNMP Agent) Setup Guide

This is the most complex step but crucial for SNMP polling.

1. **VM Setup:** Download and import the MikroTik CHR (.ova image) into your VMware Player (Step 1 of the previous guide).
2. **VM IP Verification:** Log into the MikroTik console and find its IP address (e.g., 192.168.1.208).

3. **SNMP Configuration (Inside MikroTik Console):** Run these commands to enable the SNMP agent and tell it to accept requests from your SIEM host (192.168.1.207):
   ```
   /snmp community set public address=192.168.1.207
   /snmp set enabled=yes
   ```

4. **Firewall Fix (Crucial for SNMP Polling):** MikroTik blocks incoming SNMP (UDP 161) by default. You MUST run this command to allow your SIEM to poll the router:
   ```
   /ip firewall filter add chain=input protocol=udp dst-port=161 src-address=192.168.1.207 action=accept place-before=0
   ```

Chapter 4: Dashboard Metrics and Verification

The key to your SIEM is understanding what each element tracks and how it updates.

4.1 Dashboard Metrics Explained

Dashboard Element	**Tracked Event/Log Type**	**How it is Updated**	**Why it is Important**
Total Events	All logs (Windows 4624, 4625, SNMP Polls, etc.).	Incremented immediately when the log is successfully saved to logs.db.	Measures system ingestion rate and overall activity.
Successful Logins	Windows Event ID **4624**.	Incremented by agent.py and processed by the server.	Measures successful access and potential success step in a Brute Force attack.
Failed Logins	Windows Event ID **4625**.	Incremented by agent.py.	Key metric for detecting brute force or password spray attempts.
App Errors	Windows Event ID **1000** (Application Crash).	Incremented by agent.py.	Measures endpoint stability and potential application exploitation attempts

			(pre-exploit).
Correlated Alerts	Logs matching the **correlation_engine** pattern (e.g., Brute Force Attempt).	Updated by correlation_engine thread every 30 seconds after finding a match.	Measures intelligence—the system detected a complex pattern, not just single event.

4.2 Verification & Troubleshooting (How to Check)

Feature	**How to Check (Test Case)**	**Expected Result**	**Troubleshooting (If '0')**
Windows Agents	Run python agent.py and enter 3 wrong passwords then 1 correct one.	**Failed Logins** counter should increment (e.g., 3 -> 4). **Successful Logins** increases by 1.	Ensure soc_dashboard.py is running *first* and there are no ConnectionResetError messages in the agent terminal.
SNMP Monitoring	In MikroTik VM, run /ping 8.8.8.8.	**Live Log Stream** shows a MikroTik Health Poll log with changing Uptime and increasing ether2_Traffic_IN bytes.	Ensure the **Firewall Rule (Step 3.5)** is correctly applied to MikroTik.
Brute Force Alert	Run the 3 Failed + 1 Successful login test.	**Correlated Alerts** counter increments (e.g., 2 -> 3) and a **Red Alert** appears in the Threat Center.	This confirms the database SQL query logic (the correlation_engine function) is correct.
Threat Intel	Ping a known malicious IP (e.g., 118.25.6.39).	After 1-2 minutes, the ping log turns **CRITICAL** (Red) and shows Threat	This confirms background worker threads are correctly

		Intel (Dest): Malicious (100%)...	connecting to GeoIP and AbuseIPDB APIs.

Chapter 5: Advanced Features and AI Interaction

5.1 AI-Powered SOAR (Remediation)

The Gemini AI is configured to parse your command and execute it via the run_command Python function.

Gemini Interaction (Input)	SOAR Action Executed	Purpose
block 8.8.8.8	Executes netsh advfirewall firewall add rule... command on the host OS.	Instantly blocks the malicious IP at the host firewall level.
unblock 8.8.8.8	Executes netsh advfirewall firewall delete rule... command.	Reverses the block for investigation or remediation.
Show logs for user 'Admin'	Queries the logs.db database using SQLite filters defined by the AI.	Retrieves all matching log entries.

5.2 The Asynchronous Advantage (Async Queues)

This design choice separates the quick tasks (logging data to the database) from the slow tasks (Internet Lookups).

1. **Log Reception:** soc_dashboard.py gets a log.
2. **Queueing:** It immediately places the Log ID and IP into the geoip_queue, dns_queue, and threat_intel_queue.
3. **Database Insert:** The Log is saved immediately (as a placeholder).
4. **Background Work:** Worker threads run independently to perform the slow API calls and then update the log entry in the database.

Chapter 6: Future Roadmap (World-Class Features)

To evolve this into a truly unique and powerful platform, the following features are recommended:

1. **AI-Powered "Attack Storytelling":**
    - **Idea:** When a Brute Force Alert triggers, the correlation engine automatically sends *all* related logs to Gemini.
    - **Result:** Gemini generates a narrative ("The attacker first probed the network, then tried three usernames, succeeding with the fourth, 'Eshan', before creating a new temp user.") and attaches this **Attack Story** to the critical alert.
2. **Predictive Threat Scoring (AI-Driven Risk):**
    - **Idea:** Use the AI to calculate the probability of a critical event *before* it happens.
    - **Input:** Feed low-priority signals (1 Failed Logon + Ping + Low AbuseIPDB score) to Gemini.
    - **Result:** The AI outputs a numerical **"Attack Probability: 65%"**, allowing the SOC analyst to proactively investigate the suspicious IP, making the system predictive rather than just reactive.
3. **SNMP Trap/Correlated Alert Auto-Remediation:**
    - **Idea:** Implement logic so that when a severe MikroTik Trap is received (e.g., a port goes down), the SIEM automatically runs a command (e.g., MikroTik ssh command to reboot port).