INTRO TO LOG ANALYSIS

AIM:

Gain a foundational understanding of log analysis in cybersecurity by learning to investigate events using log data from various systems. This includes identifying anomalies and suspicious behavior using command-line tools, regular expressions, and platforms like CyberChef.

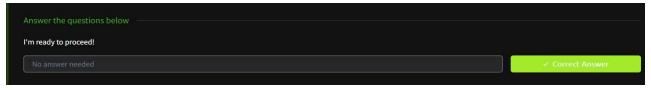
PROCEDURE:

- 1. Begin with the theory of log types, timelines, and threat indicators.
- 2. Use Linux CLI tools like 'cut', 'awk', 'grep', and 'uniq' for log filtering.
- 3. Decode obfuscated payloads with CyberChef (e.g., Base64, MACs).
- 4. Use regex patterns to extract specific values from logs.
- 5. Understand the function of Logstash Grok filters.
- 6. Write and understand detection rules using YARA and Sigma YAML format.

TASK 1 – INTRODUCTION

- Introduces log analysis and its role in cybersecurity operations.
- Explains how logs help detect and investigate malicious activity.
- Describes various log types like system, application, and security logs.
- Sets the foundation for working with forensic tools and log files.
- Highlights how log trails are essential in incident response

- Encourages a mindset of curiosity and pattern recognition



TASK 2 - TYPES OF LOGS

- Covers various types of logs used in analysis, such as Apache, DNS, Syslog.
- Explains the structure and purpose of each log type.
- Helps identify which logs are useful for which kind of threat or anomaly.
- Emphasizes reading timestamps, IPs, and method/status fields.
- Reinforces log relevance in real-world investigations.
- Forms the basis for choosing the right log during triage.



TASK 3 – INVESTIGATION THEORY

- Introduces the concept of timelines and event correlation.
- Defines a "Super Timeline" for cross-system analysis.
- Discusses threat indicators like file hashes (MD5).
- Covers visualizing events and identifying intrusion patterns.
- Questions help reinforce understanding of analysis theory.
- Equips users to think systematically during log review.



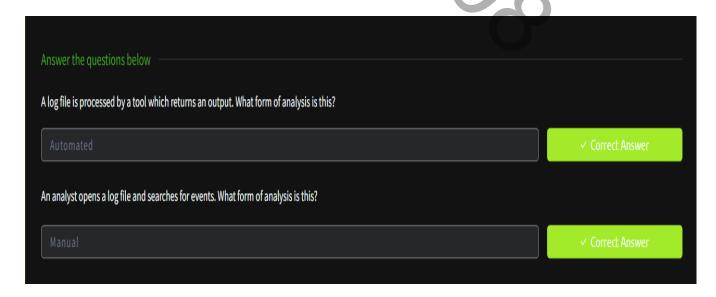
TASK 4 – DETECTION ENGINEERING

- Focuses on identifying suspicious behavior in logs.
- Highlights default log locations, like \u00e4/var/log/nginx/access.log\u00e5.
- Teaches detection of encoded attacks like path traversal.
- Shows how to decode `%2E%2E/` and other encoded threats.
- Builds awareness of signature-based log indicators.
- Practical examples prepare users for real detection tasks.



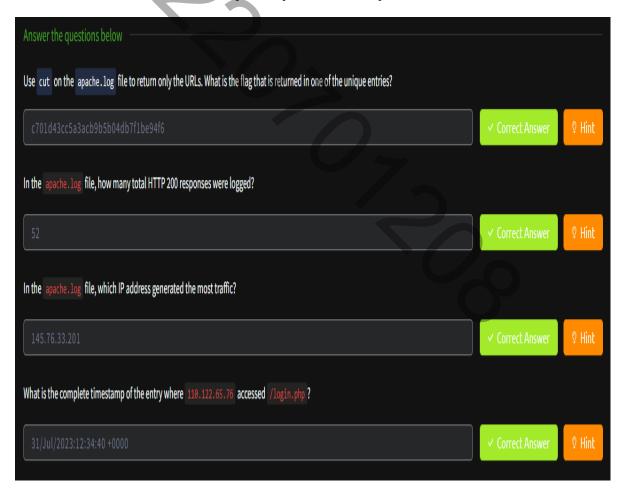
TASK 5 – AUTOMATED VS. MANUAL ANALYSIS

- Compares automated log parsing with manual investigation.
- Shows when to use tools vs. human-led judgment.
- Demonstrates strengths and limits of both approaches.
- Promotes hybrid usage of automated detection and human insight.
- Reinforces how automation saves time, but humans catch context.
- Simple Q&A makes the concept clear and applicable.



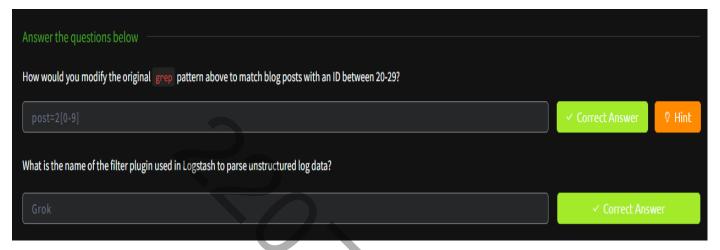
TASK 6 – LOG ANALYSIS TOOLS: COMMAND LINE

- Uses CLI tools like 'cut', 'awk', 'sort', 'uniq', and 'wc'.
- Extracts URLs, IPs, and counts response codes in Apache logs.
- Helps identify most active IPs or anomalies in logs.
- Tasks include timestamp extraction, pattern filtering.
- Encourages hands-on practice and efficient log handling.
- Reinforces Linux CLI as a primary skill for analysts.



TASK 7 – LOG ANALYSIS TOOLS: REGULAR EXPRESSIONS

- Introduces regex for log pattern extraction and filtering.
- Teaches matching ranges (e.g., post=2[2-6]) and wildcards.
- Shows how regex simplifies locating key data entries.
- Explains the Grok plugin for parsing unstructured logs.



- Forms the base for automation in SIEM log parsing.
- Builds muscle memory in log filtering precision.

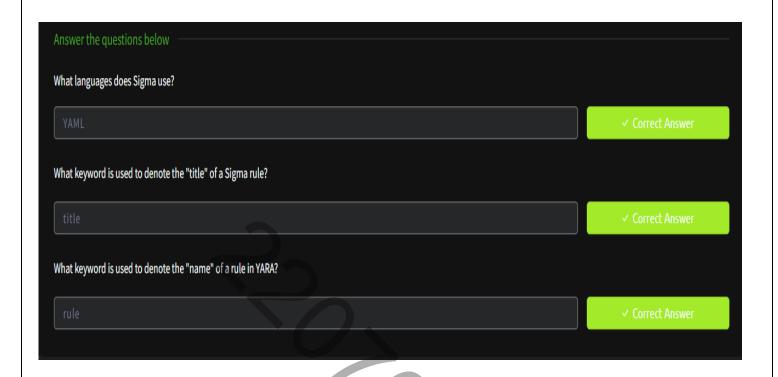
TASK 8 - LOG ANALYSIS TOOLS: CYBERCHEF

- Demonstrates use of CyberChef for IP/MAC extraction and decoding.
- Shows regex matching for IPv4 and Base64 decoding.
- Uses filters to refine large datasets into actionable data.
- Tasks include decoding embedded flags and extracting patterns.
- Reinforces visual/logical chaining of transformations.
- Makes advanced parsing accessible for beginners.



TASK 9 - LOG ANALYSIS TOOLS: YARA AND SIGMA

- Introduces detection rule writing with YARA (malware) and Sigma (logs).
- Explains syntax like 'rule' (YARA) and 'title' (Sigma YAML).
- Demonstrates how Sigma helps standardize detection across platforms.
- Teaches rule readability and structure in threat detection.
- Builds a bridge between manual detection and automated alerts.
- Finalizes the room by integrating rules into practical use.



RESULT:

Successfully understood the principles of log analysis, practiced log filtering and decoding, and applied detection rule writing using industry tools, laying a strong foundation for real-world security operations.