MES Wadia College of Engineering Pune-01

Department of Computer Engineering

Name of Student:	**Class:**
Semester/Year:	**Roll No:**
Date of Performance:	**Date of Submission:**
Examined By:	**Experiment No: Part A-01**

PROBLEM STATEMENT: Write a program for Log Capturing and Event Correlation.

AIM: The aim of the program is to **capture, store, and analyze logs** from various systems and devices, and to correlate events from those logs.

OBJECTIVES:

- To Develop a mechanism to capture logs from different sources (e.g., servers, applications, network devices) in real-time or at scheduled intervals.
- To ensure the program supports multiple log formats (e.g., syslog, event logs, flat files).
- To correlate events based on time, IP addresses, or specific keywords to find relationships between seemingly unrelated events.

APPRATUS:

Operating System recommended: 64-bit Open source Linux or its derivative

PRE-REQUISITES:

Knowledge of C, C++, python programming

Basic knowledge of computer, network and security information

THEORY:

Log Capturing and Event Correlation are crucial for maintaining system security, monitoring network activity, and providing timely responses to any anomalies within IT environments. By systematically collecting and correlating log data, organizations can identify security breaches, troubleshoot performance issues, and ensure compliance with regulatory requirements.

Log Capturing

Log capturing involves gathering detailed records of activities, transactions, and system events from a wide range of sources, including operating systems, network devices, servers, applications, and databases. These logs provide a comprehensive view of system operations, user actions, and potential security threats.

Components of Logs:

- Timestamps: Log entries include precise timestamps to record the exact time an event occurred, helping sequence events and conduct thorough investigations.
- Source Information: Logs capture details such as IP addresses, hostnames, usernames, and system IDs to identify the origin of activities and potential breaches.
- Event Details: Logs describe the nature of each event, including user login attempts, file accesses, system errors, software installations, and configuration changes.
- Severity Levels: Logs categorize events by severity (e.g., informational, warning, error, critical), allowing administrators to prioritize incidents for response.

Log Commands in Linux:

- dmesg: Displays kernel messages, including hardware events and driver information.
- Usage: dmesg | less to view messages with scrolling.
- journalctl: Accesses system logs for services, kernel messages, and user activities.
- Usage: journalctl -u sshd (logs for the SSH service), journalctl -b (logs from the current boot).
- tail: Views the last few lines of a log file, often used for real-time log monitoring.
- Usage: tail -f /var/log/syslog to watch new log entries.
- cat / less: Opens and reads log files in the /var/log directory.
- Usage: cat /var/log/auth.log or less /var/log/messages.
- grep: Searches for specific keywords in log files to quickly find relevant entries.
- Usage: grep "error" /var/log/syslog.
- logger: Adds custom messages to system logs, useful for logging from scripts.
- Usage: logger "Custom log entry".
- logrotate: Manages log file rotation, compression, and removal to optimize disk space.
- Configuration: /etc/logrotate.conf.

Event Correlation:

Event correlation involves analyzing logs to identify patterns, connect related events, and derive meaningful insights. By correlating log entries from different sources, it becomes possible to detect complex incidents and understand their root causes.

Examples of Event correlations are Boot, Authentication etc.

Key Elements of Event Correlation:

- Aggregation: Collecting logs from various sources into a centralized platform for holistic analysis, providing visibility into the entire IT infrastructure.
- Pattern Recognition: Identifying sequences of events or anomalies that could signify a security threat, such as repeated failed login attempts or unexpected changes in network traffic.

- Correlation Rules: Applying predefined rules or machine learning algorithms to connect related events, such as associating a user login from an unusual location with subsequent access to sensitive data.
- Root Cause Analysis: Correlating events to trace issues to their source, facilitating faster incident resolution and prevention of future occurrences.

CONCLUSION: Thus, we have implemented log capturing and event correlation to enhance system security and efficiently diagnose potential issues.

QUESTIONS:

- How does event correlation help in identifying security threats in an IT environment?
- What are the best practices for securely storing and managing log files to prevent tampering during forensic investigations?