

LLM-Powered Security Log Analysis: A Practical Guide

Using Large Language Models to Analyse Security Events and Investigate Threats

A technical reference for SOC analysts, security engineers, and incident response professionals seeking to integrate LLM capabilities into daily security operations, threat investigation, and SIEM workflows.

GSDC PROFESSIONAL SERIES

CYBERSECURITY OPERATIONS

Understanding LLMs in Security Log Analysis

Modern enterprise environments generate enormous volumes of security log data every second — spanning firewalls, endpoints, servers, cloud platforms, and identity systems. Each source produces structured and semi-structured records capturing authentication events, process executions, network connections, and configuration changes. Individually, these records are rarely sufficient to confirm a threat. It is the relationships between events, across users, devices, and time, that reveal suspicious patterns and attacker behaviour.

Large Language Models bring a fundamentally different capability to log analysis. Rather than matching events against rigid detection rules or signatures, LLMs interpret log records using natural language reasoning. An analyst can present a sequence of events and ask the model to summarise what occurred, identify anomalies, suggest related data sources, or generate investigation queries — all in plain English. This reduces the time required to triage complex incidents and helps less experienced analysts structure investigations effectively.

LLMs are not replacements for SIEM platforms or EDR tools. They function best as an analytical layer operating alongside existing infrastructure. SIEM platforms ingest, index, and correlate log data at scale; EDR solutions detect and respond to endpoint threats in real time. LLMs add value by interpreting the outputs of these platforms — summarising alerts, connecting related events, drafting queries, and preparing investigation reports. The model works with the data that SIEM and EDR provide, helping analysts reason through what they are seeing.



Event Summarisation

Convert verbose, multi-field log records into concise, readable summaries that capture the key security-relevant details without requiring analysts to parse raw data manually.



Suspicious Activity Analysis

Examine sequences of related events and identify patterns consistent with credential abuse, lateral movement, privilege escalation, or data exfiltration attempts.



Investigation Support

Help analysts structure investigations by suggesting relevant log sources, related event identifiers, and logical next steps based on the events already under review.



SIEM Query Generation

Draft KQL, SPL, or SQL queries for further log retrieval, allowing analysts to accelerate the data-gathering phase without manually constructing complex query syntax from scratch.

Security Log Sources and Data Types

Effective LLM-assisted analysis depends on the quality and completeness of the underlying log data. Security teams typically draw from multiple distinct log sources, each providing different visibility into user activity, system behaviour, and network communications. Understanding what each source provides — and which fields carry the most investigative value — is essential for structuring effective LLM prompts and investigations.

Key Log Field Types

- **Timestamp** — precise event date and time (UTC preferred)
- **Username / UPN** — account performing the action
- **Source IP** — originating address of the connection
- **Destination IP / hostname** — target system or resource
- **Event ID** — platform-specific event classifier
- **Process name** — executable involved in the event
- **Command-line arguments** — full execution string
- **Action / result** — success, failure, blocked, allowed
- **Device / hostname** — system generating the log
- **Geo-location** — derived from IP where available

Primary Log Source Categories

SIEM Alerts & Event Records

Correlated alerts, normalised event records, and rule-triggered detections from platforms such as Microsoft Sentinel, Splunk, or IBM QRadar.

EDR Logs

Endpoint telemetry including process creation, file writes, registry changes, network connections, and lateral tool transfers from tools such as CrowdStrike Falcon or Microsoft Defender for Endpoint.

Firewall & Network Traffic Logs

Allowed and denied connections, protocol usage, port activity, and data transfer volumes from perimeter and internal network devices.

Identity & Authentication Records

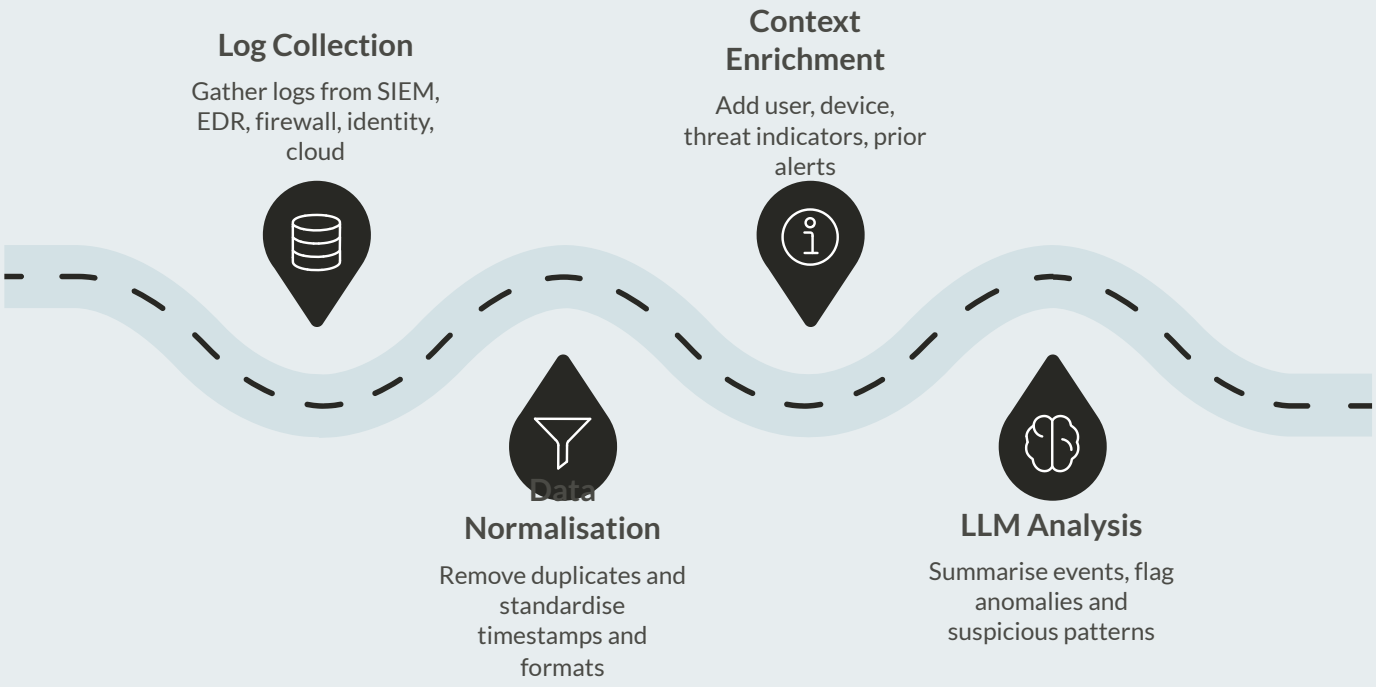
Login attempts, MFA challenges, token issuance, privilege changes, and account lockout events from Active Directory, Entra ID, or Okta.

Cloud Audit & Application Logs

API calls, IAM role assignments, storage access, configuration changes, and service-level activity from AWS CloudTrail, Azure Monitor, or GCP Audit Logs.

LLM-Based Security Log Analysis Workflow

A structured workflow ensures that log data passed to an LLM is accurate, well-contextualised, and actionable. The following six-stage process reflects how security teams can integrate LLM analysis into existing SOC operations without compromising data quality or analytical rigour. Each stage has a defined purpose and produces a specific output that feeds the next stage.



Stages 1–3: Data Preparation

- **Log Collection:** Aggregate relevant records from all applicable security sources into a single investigation dataset, scoped to the relevant timeframe, user accounts, and systems.
- **Filtering & Normalisation:** Remove noise, deduplicate records, standardise timestamps to UTC, and map fields to a consistent schema before presenting data to the model.
- **Context Enrichment:** Append relevant context — previous alerts for the same user, known threat indicators, device ownership, and business role — to give the LLM the broader picture needed for accurate analysis.

Stages 4–6: Analysis and Output

- **LLM-Based Analysis:** Submit the prepared and enriched dataset with a structured prompt. The model summarises events, flags anomalies, and identifies patterns consistent with known attack techniques.
- **SIEM Query Generation:** Request draft queries to retrieve additional related log records. Always review and test queries in a non-production environment before execution against live data.
- **Analyst Validation:** Every LLM output must be validated against original log sources. Analysts confirm findings, discard unsupported conclusions, and document the investigation outcome.

Preparing and Normalising Log Data

The quality of LLM analysis is directly proportional to the quality of the input data. Raw security logs are frequently inconsistent — timestamps may appear in different time zones, field names vary between platforms, duplicate records inflate event counts, and irrelevant noise obscures the events that matter. Preparing log data before submitting it to an LLM is not optional; it is a prerequisite for reliable analysis.

The normalisation process should follow a consistent set of steps: remove duplicate records caused by log forwarding overlap; standardise all timestamps to UTC in ISO 8601 format; extract and label key fields such as username, source IP, event type, and outcome; group related events by user, device, and time range; and mask or redact sensitive personal data, password hashes, and account credentials in accordance with your data handling policies before passing records to any external model.

Remove Duplicates

Deduplicate log records originating from multiple forwarders or ingestion pipelines.

Standardise Timestamps

Convert all timestamps to UTC ISO 8601. Inconsistent time zones distort event timelines.

Extract Key Fields

Parse and label: username, source IP, destination, event ID, action, and result.

Group by Entity

Cluster events by user account, host, and time range to surface behavioural patterns.

Mask Sensitive Data

Redact password hashes, PII, and credentials before submitting to any LLM endpoint.

Before and After: Log Normalisation Example

Field	Raw Log Record (Before)	Normalised Record (After)
Timestamp	`3/14/2025 9:14 AM EST`	`2025-03-14T14:14:00Z`
Username	`jsmith / DOMAIN\jsmith`	`jsmith@contoso.com`
Source IP	`src=185.220.101.47`	`source_ip: 185.220.101.47`
Event Type	`4625 - Logon failure`	`event_type: authentication_failure`
Sensitive Field	`password_hash: aad3b435...`	`password_hash: [REDACTED]`

Providing Context to LLMs

An LLM analysing a single log event in isolation has limited investigative value. Without context, the model cannot distinguish between a legitimate remote login from a travelling employee and an attacker using stolen credentials from an unfamiliar location. Effective LLM-assisted analysis requires deliberate context assembly — gathering and structuring the information the model needs to reason accurately about the events under review.

Useful context includes: active security alerts associated with the user or device, the user's typical working patterns and business role, device ownership and configuration baseline, known threat indicators (IP reputation, domains, file hashes), and any prior incidents involving the same account. Providing this context transforms a generic event summary into a targeted, security-relevant assessment.

Retrieval-Augmented Generation (RAG)

RAG enables LLMs to retrieve relevant information from approved knowledge sources at query time rather than relying solely on training data. In a security operations context, this means the model can reference your organisation's approved threat intelligence feeds, internal runbooks, incident response playbooks, and known-good baseline documentation when analysing events. This significantly improves the quality and relevance of the model's output — particularly for organisation-specific scenarios — and reduces the risk of the model drawing on outdated or inapplicable general knowledge.

Implementing RAG for security operations requires maintaining a curated, versioned knowledge base of approved sources. Only authorised documentation should be indexed. Access to the retrieval system should be governed by role-based controls consistent with your broader SIEM access policies.

Example: Context-Aware Investigation Prompt

System instruction: You are a security analyst assistant. Analyse the following log events and context. Identify any suspicious patterns, explain what may have occurred, and suggest next investigation steps. Do not make assumptions beyond what the data supports.

Context provided:

- User: jsmith@contoso.com — Finance department, standard business hours Mon–Fri
- Alert: MFA bypass attempt detected at 09:12 UTC
- Device: LAPTOP-FIN-042 — last seen on corporate network 2025-03-13
- Threat indicator: 185.220.101.47 listed in Tor exit node feed

Log events: [normalised log records appended below]

Query: What is the most likely explanation for this sequence of events, and what additional logs should I retrieve?

Never include sensitive personal data, cleartext credentials, or non-anonymised customer records in LLM prompts, particularly when using external model endpoints. Apply data minimisation as a default.

Identifying Suspicious Activity with LLMs

LLMs are particularly effective at recognising sequences of events that, taken together, are consistent with attacker behaviour. A single failed login is routine; twelve failed logins across multiple accounts in four minutes, followed by one successful authentication from an IP address associated with a Tor exit node, tells a very different story. The model's ability to reason about event sequences, timing, and contextual plausibility is what makes it valuable in this phase of analysis.

Credential Abuse: Failed Logins Followed by Success

Sample events: Event ID 4625 × 14 for jsmith@contoso.com between 09:10–09:13 UTC, source IP 185.220.101.47. Event ID 4624 (successful logon) at 09:14 UTC, same source IP.

Investigate: Is this IP known? Has the user reported travel? Were MFA controls satisfied? Pull authentication logs for the preceding 48 hours for this account.

Unexpected Privilege Escalation

Sample events: Event ID 4728 — user jsmith added to Domain Admins group at 09:27 UTC by account svc-backup. No change management ticket associated with this action.

Investigate: Who authorised svc-backup to modify group membership? Is this account behaving outside its baseline? Review all actions performed by svc-backup in the past 24 hours.

Suspicious PowerShell Execution

Sample events: Process creation event — `powershell.exe -ExecutionPolicy Bypass -EncodedCommand [base64 string]` spawned by `winword.exe` on LAPTOP-FIN-042 at 09:19 UTC.

Investigate: Decode the base64 payload. Examine the parent-child process relationship. Was this document opened as an email attachment? Check for subsequent network connections or file writes.

Unusual External Network Connections

Sample events: Outbound TCP connection from LAPTOP-FIN-042 to 203.0.113.88:443, 1.4 GB transferred over 22 minutes. Destination not in approved application list. No associated business justification.

Investigate: Resolve the destination IP and check reputation. Identify the process initiating the connection. Review DNS query logs for related domains. Check for similar behaviour from other hosts.

Unusual Access to Sensitive Files

Sample events: File access events for `\\fileserver01\finance\payroll_Q1_2025.xlsx` and `\\fileserver01\finance\vendor_contracts\` by jsmith at 09:21 UTC — 47 files accessed in 3 minutes.

Investigate: Is this volume of file access within normal patterns for this user? Were files copied or only read? Check for staging or compression activity on the same host.

Generating SIEM Queries with LLMs

One of the most time-saving applications of LLMs in security operations is the generation of draft SIEM queries. Constructing complex queries in KQL, SPL, or SQL requires familiarity with platform-specific syntax, schema knowledge, and an understanding of which fields to filter on. LLMs can produce syntactically reasonable drafts from plain-language investigation requests, significantly accelerating the data-retrieval phase of an investigation.

It is critical that analysts treat all LLM-generated queries as drafts requiring review. Field names, table names, and filter logic must be verified against your specific SIEM schema before execution. Queries should be tested against a limited time range or non-production dataset first. Never execute an auto-generated query against live production data without analyst validation — particularly queries involving deletion, forwarding, or modification actions.

Example Investigation Request

> Analyst prompt: "Generate a KQL query for Microsoft Sentinel to retrieve all authentication events for the user jsmith@contoso.com in the past 24 hours, including both successful and failed logons, the source IP address, and the device name."

Illustrative KQL Output (Microsoft Sentinel)

```
// ILLUSTRATIVE EXAMPLE ONLY
// Verify field names against your
// Sentinel schema before execution

SigninLogs
| where TimeGenerated >= ago(24h)
| where UserPrincipalName ==
  "jsmith@contoso.com"
| where ResultType in ("0", "50126",
  "50053", "50055", "50056")
| project TimeGenerated,
  UserPrincipalName,
  IPAddress,
  DeviceDetail,
  ResultType,
  ResultDescription,
  Location
| order by TimeGenerated asc
```

Illustrative SPL Output (Splunk)

```
// ILLUSTRATIVE EXAMPLE ONLY
// Verify index names and field
// mappings for your environment

index=windows_security
  EventCode IN (4624, 4625, 4648)
  user="jsmith@contoso.com"
  earliest=-24h@h
  latest=now
| table _time, user,
    src_ip, dest_host,
    EventCode, action,
    failure_reason
| sort _time
```

Query Review Checklist

1 Verify field names

Confirm that field names match your platform's actual schema — table names and field labels vary by version and configuration.

2 Test on limited scope

Run against a narrow time window or specific host before broadening scope to full dataset.

3 Validate result logic

Confirm that the filter conditions accurately represent your investigative intent and do not inadvertently exclude relevant records.

Investigating Common Cybersecurity Threats

Different attack techniques leave distinct log signatures. LLMs can connect related events across sources to help analysts spot them quickly.

Threat Category	Key Log Indicators	Relevant Sources	LLM Analysis Value
Brute-Force / Password Spraying	Multiple Event ID 4625 across accounts; account lockouts (4740)	Entra ID / AD logs; SIEM authentication alerts	Distinguishes spraying from single-account brute force
Lateral Movement	Remote logons (4624, Type 3/10); PsExec or WMI execution; admin share access	EDR telemetry; Windows Security logs	Maps movement from initial host to downstream targets
Suspicious Process Execution	Encoded PowerShell; odd parent-child chains (e.g. Word spawning cmd.exe)	EDR process events; Sysmon Event ID 1	Decodes obfuscated commands and flags attack tooling
Data Exfiltration	Bulk file access; large outbound transfers; compression tool use	Firewall and proxy logs; DLP alerts	Links file access volume to outbound transfer timing
Cloud Threats	Unusual API calls (GetSecretValue, CreateAccessKey); IAM role changes	CloudTrail; Entra ID / Azure audit logs	Spots IAM changes followed by resource access

Present the full timeline of related events, not isolated records. Ask the model which attack technique the sequence resembles and which additional logs would confirm or rule it out. Treat its assessment as a starting point, not a conclusion.

Practical Case Study: Investigating a Suspicious Login

The following hypothetical scenario illustrates how an LLM can support a SOC analyst investigating a suspicious authentication event in a typical enterprise environment. All data is fictional and used for illustrative purposes only. The scenario involves a finance department employee account and a sequence of events spanning approximately 17 minutes on a weekday morning.

Incident Event Timeline — jsmith@contoso.com — 2025-03-14

Time (UTC)	Event	Detail	LLM Flag
09:10	14 × Authentication Failure	Event ID 4625; source IP 185.220.101.47 (Tor exit node); targeting jsmith@contoso.com	⚠️ High volume, Tor source — consistent with credential stuffing
09:14	Successful Authentication	Event ID 4624; same source IP; Logon Type 10 (RemoteInteractive); no MFA event recorded	🔴 MFA not satisfied — possible bypass; unfamiliar IP
09:18	Internal Application Access	Access to Finance ERP portal from authenticated session; 23 records viewed	⚠️ Access shortly after suspicious logon; review data accessed
09:23	Access Denied — Restricted Resource	Attempt to access `\\fileserver01\executive\board_reports\`; access denied (insufficient permissions)	⚠️ Attempting to access resources outside user's normal scope
09:27	Permission Change	Event ID 4728; jsmith added to Finance-Senior-Access group by svc-backup account	🔴 Privilege escalation via service account — high severity

LLM-Generated Investigation Summary

The model, when presented with the above timeline and context, produced the following assessment: *"The sequence is consistent with a credential-based account compromise. The pattern of multiple authentication failures followed by a successful logon without MFA from a Tor exit node strongly suggests credential abuse rather than legitimate user activity. Subsequent access to sensitive finance data and an attempt to access board-level resources indicates reconnaissance or data harvesting intent. The privilege escalation via svc-backup warrants immediate investigation of that service account's recent activity."*

LLM-Suggested Next Steps

- Retrieve all actions performed by svc-backup in the past 7 days
- Pull full session activity for the authenticated jsmith session from 09:14 onwards
- Check whether jsmith's credentials appear in known breach databases
- Review MFA configuration for the jsmith account — confirm bypass method
- Identify any files downloaded or exported during the ERP session at 09:18

Analyst Validation and Response

The analyst validated the LLM findings against original log records in Microsoft Sentinel. All flagged events were confirmed. The svc-backup account was found to have been used to modify group membership for three additional accounts in the preceding 48 hours — a finding not initially included in the LLM prompt.

Response actions taken: jsmith account suspended pending investigation; svc-backup account disabled and credentials rotated; MFA policy enforced for all Finance accounts; incident escalated to Tier 3 for full forensic review; IR ticket opened with supporting evidence package.

> The LLM's analysis accelerated initial triage but did not replace analyst judgement. The svc-backup lateral finding was discovered through analyst-initiated follow-up queries, not LLM output.

Preparing Investigation Summaries and Reports

A well-structured investigation report translates raw log findings into a clear, actionable document that supports decision-making by security management, legal teams, and incident response stakeholders. LLMs can significantly accelerate the drafting process by organising identified events into a consistent report structure, reducing the time analysts spend on documentation without compromising accuracy. All LLM-drafted content must be verified against original log records before the report is finalised or shared.

Standard Investigation Report Structure

01

Incident Overview

Brief description of the incident type, detection source, initial alert time, and assigned severity. One paragraph maximum.

02

Timeline of Events

Chronological list of confirmed events with timestamps, event types, source/destination, and analyst notes. Drawn directly from verified log records.

03

Affected Users and Systems

List of impacted accounts, hostnames, services, and cloud resources, with business owner and department where known.

04

Indicators of Compromise

Source IPs, file hashes, domain names, event IDs, and other technical indicators extracted from the investigation dataset.

05

Supporting Evidence

References to specific log records, SIEM alert IDs, EDR detection references, and query outputs used to support findings. All evidence must trace to original sources.

06

Recommended Next Steps

Prioritised list of further investigation actions, containment steps, and remediation recommendations based on confirmed findings.

LLM Prompt for Report Drafting

> System: You are a security analyst assistant. Using only the confirmed findings below, draft an investigation summary following the standard six-section report structure. Do not infer or add details not present in the confirmed findings. Flag any section where additional information is needed.

Quality Assurance for LLM-Drafted Reports

→ Trace every claim to source

Each factual statement in the report must reference a specific log entry, alert ID, or query output. Remove any conclusion the model introduced without evidential basis.

→ Verify timestamps and identifiers

Confirm that usernames, IP addresses, hostnames, and event IDs in the draft match the original log records exactly. LLMs occasionally introduce minor transcription errors.

→ Apply classification before distribution

Apply your organisation's document classification scheme before sharing. Investigation reports typically contain sensitive data subject to access controls

Risks and Limitations of LLM-Based Log Analysis

LLMs introduce specific risks to security log analysis that must be understood and controlled before production use. The matrix below summarises each risk and its mitigation.

Risk	Description	Control Measure	Responsibility
Incorrect Conclusions	Plausible but inaccurate assessments can misdirect investigations.	Validate all findings against original log records.	SOC Analyst / Tier 2
Incomplete Log Data	Gaps or inconsistent timestamps produce partial conclusions.	Verify log completeness before submission.	Security Engineer
Sensitive Data Exposure	Logs may contain personal or regulated data.	Mask sensitive fields; use private model deployments.	Security Architect / DPO
Prompt Injection	Malicious log content can manipulate model behaviour.	Treat all log content as untrusted input.	Security Engineer
Limited Investigation Context	Generic analysis may miss your environment's specifics.	Enrich prompts with user, device, and alert context.	SOC Lead / Analyst
Excessive Permissions	Over-privileged integrations increase breach impact.	Apply least privilege; require analyst approval for actions.	Security Architect

⚠ Prompt injection is an active attacker research area, so prioritise technical controls for untrusted log sources.

Best Practices for Implementing LLMs in Security Operations

Deploying LLMs in a security operations context requires deliberate architectural choices, clear governance policies, and ongoing oversight of model behaviour. The following best practices reflect the operational requirements of enterprise SOC environments and address the specific risks associated with LLM use in security-sensitive workflows.

Implementation Principles

Controlled API Integration

Connect LLMs to SIEM and EDR platforms through dedicated, monitored API gateways with rate limiting, logging, and alerting on anomalous usage patterns. Never grant direct database access to model integrations.

Role-Based Access and Least Privilege

Assign LLM integration accounts the minimum permissions required for read-only log retrieval. Separate the integration service account from accounts with write, delete, or response action permissions.

Sensitive Log Data Protection

Implement automated field masking or tokenisation for PII, credentials, and regulated data fields before log records enter any LLM processing pipeline. Document the masking logic and test it regularly.

Approved RAG Knowledge Sources

Maintain a curated, versioned index of approved documents for retrieval — threat intelligence, runbooks, playbooks, and baseline documentation. Apply access controls to the retrieval layer consistent with SIEM access policy.

Mandatory Analyst Review

Establish a policy requiring analyst review and approval before executing any LLM-generated query or taking any response action based on model output. Document the approval in the incident record.

Audit Trail of Model Interactions

Log all prompts submitted to the model, all responses received, and all subsequent analyst actions taken. Retain these records in accordance with your incident documentation policy.

Implementation Checklist

Use the following checklist when preparing to deploy or audit LLM capabilities within security operations:

- [ ] Define scope: which log sources and use cases will the LLM process?
- [ ] Identify and classify all log fields — flag sensitive fields for masking
- [ ] Select deployment model: cloud API, private cloud, or on-premises
- [ ] Configure field masking pipeline and test against representative log samples
- [ ] Create dedicated, least-privilege API service accounts for SIEM/EDR integration
- [ ] Build and curate approved RAG knowledge base; version control enabled
- [ ] Define and document mandatory analyst review policy for model outputs
- [ ] Implement prompt logging and response audit trail with retention policy
- [ ] Configure monitoring for prompt injection indicators and anomalous model outputs
- [ ] Conduct red team exercise: attempt prompt injection via log data
- [ ] Establish output quality review cadence — monthly minimum
- [ ] Train SOC analysts on LLM capabilities, limitations, and review responsibilities
- [ ] Document all integration components in security architecture register
- [ ] Schedule quarterly review of LLM access permissions and API scope

Conclusion

Large Language Models offer genuine, practical value for security operations teams. When applied thoughtfully — with appropriate data preparation, context enrichment, analyst oversight, and governance controls — they can meaningfully accelerate log analysis, support threat investigation, and reduce the burden of documentation on experienced analysts. The case study and workflows presented in this guide reflect practical approaches to integrating LLM capabilities into real SOC environments.

LLMs are most effective when they complement, rather than replace, existing security infrastructure and analyst expertise. SIEM platforms, EDR solutions, and experienced security professionals remain the foundation of effective security operations. LLMs add an analytical layer that helps analysts reason through complex event sequences, generate investigation queries, and structure findings — but every output requires human validation before action is taken.

As the technology matures and more organisations deploy generative AI within their security operations, understanding both the capabilities and the risks of these tools becomes a core competency for security professionals. The teams that will use this technology most effectively are those who approach it with both enthusiasm for its potential and rigour in their implementation practices.

GSDC CERTIFICATION

Prepare for the Next Wave of AI-Powered Cyber Threats

The **GSDC Generative AI in Cybersecurity Certification** is designed for security professionals who want to develop structured, practical knowledge of generative AI applications and their cybersecurity implications — including both how to use these tools responsibly and how to defend against AI-powered threats.



AI Applications in Cybersecurity

Understand how generative AI and LLMs are applied across security operations, threat detection, and incident response in modern enterprise environments.



AI-Driven Threats and Attack Techniques

Identify emerging attack techniques enabled by generative AI, including AI-generated phishing, deepfake social engineering, and automated vulnerability research.



Security Risks of Generative AI

Explore the security vulnerabilities and risks associated with deploying generative AI systems, including prompt injection, data leakage, and model supply chain threats.



Risk Assessment and Responsible AI

Learn risk assessment frameworks, security control design, and responsible AI practices applicable to organisations deploying generative AI in security-sensitive contexts.

Explore the GSDC Generative AI in Cybersecurity Certificati... | **Visit GSDC — www.gsdcouncil.o...**

GSDC
Global Skill Development Council

CERTIFICATION IN GENERATIVE AI IN CYBERSECURITY



ABOUT GSDC CERTIFICATION

EBOOK

Extensive and exclusive Ebook created by world's experts to help you with understanding core concepts.

LEARNING MATERIALS

Get access to learning materials such as videos, ebooks, templates, and practice exams, which will help you clear the certification exam.

CREATED BY EXPERTS

GSDC certifications are created and authored by world's leading experts in the field.

LEARNING OBJECTIVE

- Gain insights into autonomous decision-making processes
- Apply knowledge using ready-to-implement templates
- Demonstrate ability to work with Agentic AI models
- Validate your skills wit

Enroll now with the code **LEARN20** To avail **20%** discount

Enroll Now

www.gsdcouncil.org