Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

SIEM Detection Engineering: A Survey of Practices, Challenges and Research Directions

Jini Fernandez

Independent Researcher, Riyadh, Saudi Arabia

DOI: https://doi.org/10.63363/aijfr.2026.v07i06.8886

Abstract
Security Information and Event Management (SIEM) platforms remain the operational core of most Security Operations Centers (SOCs), yet their value depends less on the platform than on the quality of the detections built on it. Detection engineering, the discipline of designing, testing, deploying and maintaining detection logic, has emerged as a distinct practice, but the literature describing it is fragmented across academic venues, vendor documentation and practitioner reports. This paper surveys detection engineering for SIEM environments. We first describe the SIEM data pipeline and the detection lifecycle. We then propose a taxonomy of detection approaches, covering signature and correlation rules, behavioral and statistical analytics, machine-learning-based detection, and hunt-derived detection, and map them to the MITRE ATT&CK framework. We analyze recurring operational challenges, including false-positive volume and alert fatigue, data quality and normalization, detection coverage measurement, and the migration of content between SIEM platforms. The survey is complemented by a practitioner perspective drawn from more than ten years of SIEM engineering on enterprise and managed-security deployments. We conclude with open research directions, including verified detection migration, outcome-based coverage measurement, and the use of large language models in detection engineering.

Keywords: SIEM, Detection Engineering, Security Operations Center, MITRE ATT&CK, Alert Fatigue, Threat Detection

1. Introduction

Organizations now generate security telemetry at a volume that no analyst team can review manually. Security Information and Event Management (SIEM) platforms address this by centralizing logs from endpoints, networks, identity providers and cloud services, normalizing them into a common structure, and applying detection logic that raises alerts for human investigation [1, 2]. In most organizations, the SIEM is the analytical core of the Security Operations Center (SOC) [3].

Platform capability alone does not deliver detection. A SIEM populated with weak or poorly maintained rules either stays silent during a real intrusion or floods analysts with alerts they cannot triage. Field studies of SOC analysts report that false positives make up the large majority of alerts, and that alert volume contributes to fatigue, burnout and missed incidents [4–6]. The quality of detection content is therefore a primary factor in whether a SIEM investment reduces organizational risk.

Detection engineering has emerged in response as a discipline in its own right. It treats detections as engineered artifacts: each is designed against a threat model, written in a query or rule language, tested against representative data, placed under version control, deployed through change management, measured in production and eventually retired. Shared frameworks support this practice. MITRE ATT&CK provides a common vocabulary of adversary tactics and techniques against which detections can be mapped [7], and the Sigma project provides a platform-neutral rule format that can be converted into the query languages of individual SIEM products [8].

Knowledge about detection engineering nevertheless remains fragmented. Academic literature concentrates on individual detection algorithms, particularly machine-learning methods [9], while day-to-day engineering practice is documented mainly in vendor material and practitioner reports. Existing surveys of SIEM and SOC operations [2, 3] describe architecture and organizational issues but give limited attention to how detections are designed, validated and maintained over time.

This paper addresses that gap. Its contributions are:

1. A structured description of the SIEM detection lifecycle, from data onboarding to detection retirement.
2. A taxonomy of five families of detection approaches, with their strengths, weaknesses and typical ATT&CK coverage.
3. An analysis of five recurring operational challenges, combining published evidence with practitioner experience.
4. A set of open research directions intended to align academic work with operational needs.

The remainder of the paper is organized as follows. Section 2 gives background on SIEM architecture and the detection lifecycle. Section 3 describes the review method. Section 4 presents the taxonomy, and Section 5 discusses operational challenges. Section 6 offers a practitioner perspective, Section 7 outlines research directions, and Section 8 concludes.

2. Background

2.1. The SIEM Data Pipeline

Most detection failures originate before any detection logic runs. A SIEM can be described as a six-stage pipeline (Figure 1), and a weakness at any stage limits what the later stages can achieve.

Collection. Data reaches the SIEM through host agents, syslog forwarding, vendor APIs and cloud-native connectors. Gaps at this stage, such as unmonitored cloud control-plane logs or partially deployed endpoint agents, cannot be compensated for later.

Parsing and normalization. Raw events are parsed and mapped to a common schema so that one detection can apply across vendors. Widely used schemas include the Splunk Common Information Model, the ArcSight Common Event Format (CEF), the Elastic Common Schema and the Open Cybersecurity Schema Framework (OCSF) [10]. Parsing errors silently remove events from detection scope.

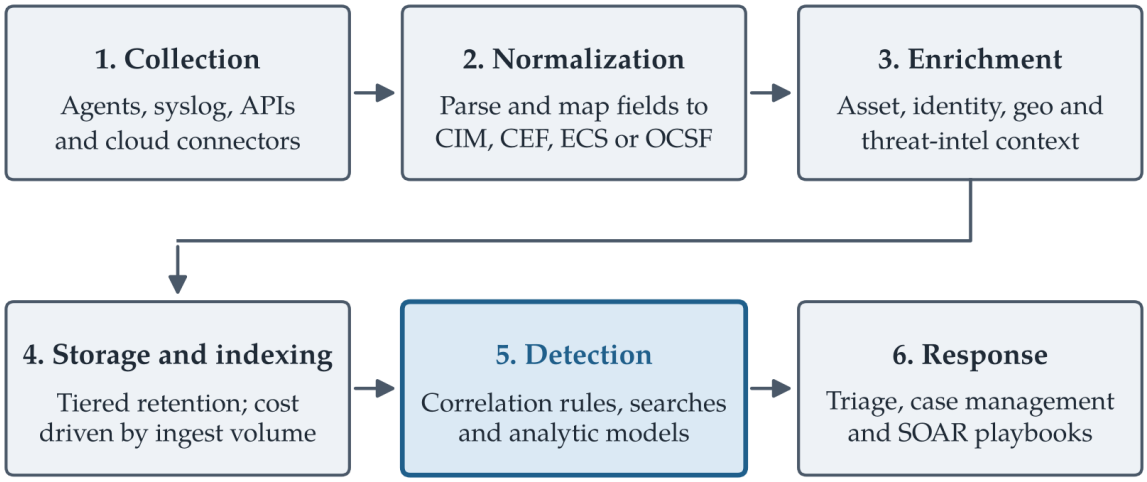
Enrichment. Events are augmented with context such as asset criticality, user identity and role, geolocation and threat-intelligence reputation [11]. Enrichment allows a detection to distinguish an administrator's routine action from the same action performed by a service account.

Storage and indexing. Data is stored in tiers that balance query speed against cost. Because many commercial SIEMs are licensed by ingest volume, retention and indexing decisions directly constrain what can be detected.

Detection. Detection logic runs as scheduled searches, real-time correlation rules or analytic models, producing alerts or notable events.

Response. Alerts are triaged in case-management tools and are increasingly enriched or contained automatically through Security Orchestration, Automation and Response (SOAR) playbooks.

Figure 1: The SIEM Data Pipeline



Stages 1–4 can fail silently: a broken log source or parser disables detections without raising an error.

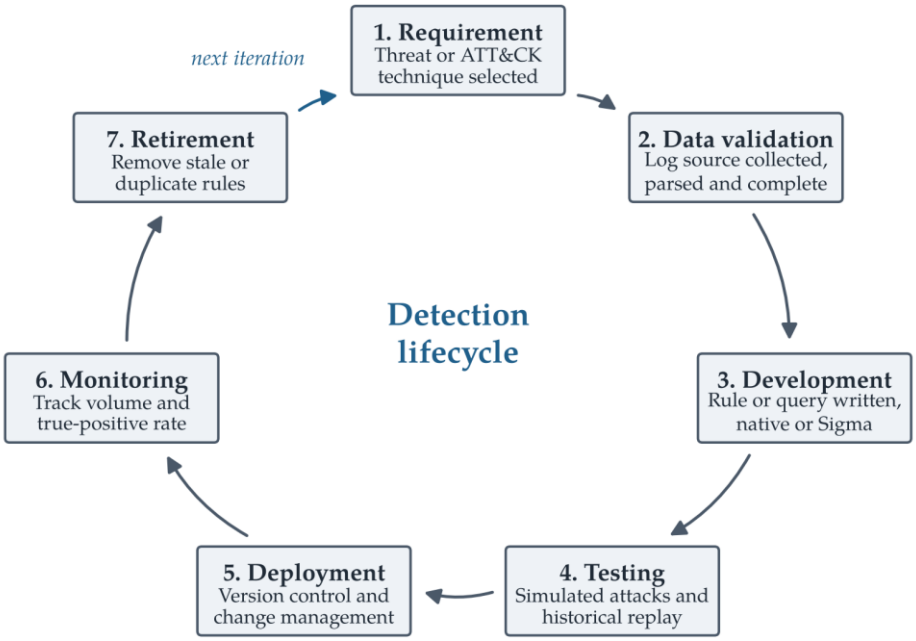
2.2. The Detection Lifecycle

A detection is not a one-time artifact; it passes through a recurring lifecycle (Figure 2):

1. **Requirement.** A threat or ATT&CK technique is selected, based on threat intelligence, incident history or an identified coverage gap.
2. **Data validation.** The engineer confirms that the required log source is collected, parsed and complete.
3. **Development.** Logic is written in the platform's query or rule language, or in a neutral format such as Sigma.
4. **Testing.** The detection is validated against simulated attack activity and replayed historical data, and its expected false-positive rate is estimated.

5. **Deployment.** The detection is released through version control and change management.
6. **Tuning and monitoring.** Alert volume and true-positive rate are tracked, and justified exclusions are added.
7. **Retirement or redesign.** Detections that no longer fire, duplicate others, or depend on a changed data source are removed or rebuilt.

Figure 2: The Detection Lifecycle



2.3. Platform Evolution

SIEM platforms have evolved through three broad generations. Early platforms, such as ArcSight ESM, centered on real-time correlation rules applied to normalized event streams. Search-centric platforms, such as Splunk Enterprise Security, moved detection toward scheduled queries over indexed data, giving engineers greater flexibility at the cost of latency. Recent platforms, including Microsoft Sentinel and Palo Alto Networks Cortex XSIAM, combine SIEM, SOAR and endpoint telemetry with built-in analytics and machine-learning models. Each generation changes how detections are expressed, but the lifecycle described above applies to all of them.

3. Review Methodology

This paper follows a structured narrative review. Sources were identified through searches of IEEE Xplore, the ACM Digital Library, Scopus, SpringerLink and Google Scholar, using combinations of the following terms: "SIEM" with "detection" or "correlation rule"; "security operations center" with "alert fatigue"; "detection engineering"; "MITRE ATT&CK" with "detection coverage"; and "SIEM" with "machine learning" or "anomaly detection".

Peer-reviewed publications in English from 2000 to 2026 were included if they addressed threat detection in a SIEM or SOC context. Studies limited to network intrusion-detection signatures without SIEM integration, marketing material and duplicate publications were excluded. Because much

detection-engineering practice is not published academically, the review also draws on a limited set of grey literature, including MITRE documentation, open-source rule repositories and vendor technical documentation. These sources are identified as such and are used only to describe frameworks and platform features.

Because the aim was to map the field rather than to aggregate quantitative results, the review is narrative rather than systematic: no formal quality scoring or meta-analysis was performed. Studies were selected for their relevance to the design, validation and operation of detections, and each was classified according to the taxonomy presented in Section 4. A generative AI tool (Claude, Anthropic) was used to assist in drafting text, structuring the review, and locating and verifying references; all output was reviewed, edited and verified by the author, who takes full responsibility for the content.

4. A Taxonomy of Detection Approaches

Detection approaches in the reviewed literature fall into five families, summarized in Table 1. The families are not mutually exclusive. Mature SOCs combine them, typically using precise rule-based detections for known threats and behavioral or statistical methods to surface activity that rules miss.

Table 1: Families of SIEM Detection Approaches

Family	Mechanism	Strengths	Weaknesses	Typical ATT&CK tactics
Signature and indicator matching	Match known indicators (hashes, IPs, domains)	Precise, inexpensive	Brittle, short-lived	Command and Control, Initial Access
Correlation rules	Threshold, sequence or join across events	Explainable, low latency	Tuning burden, false positives	Credential Access, Lateral Movement
Behavioral and statistical analytics	Baseline and deviation (UEBA, rarity, peer groups)	Detects novel activity	Noisy, needs history	Discovery, Exfiltration
Machine-learning detection	Supervised or unsupervised models on log features	Scales to large volumes	Scarce labels, limited explainability	Varies by model
Hunt-derived detections	Hypothesis-driven hunts converted to rules	Grounded in real adversary behavior	Manual, low throughput	Persistence, Defense Evasion

4.1. Signature and Indicator Matching

The simplest family matches events against known indicators of compromise (IOCs), such as file hashes, IP addresses, domains and URLs, usually sourced from threat-intelligence feeds [11, 12]. Indicator matching is computationally cheap and produces few false positives when indicators are accurate. Its weakness is fragility. Adversaries change infrastructure and tooling easily, so indicators expire quickly,

and stale feeds become a source of noise rather than signal, a data-quality problem also reported in studies of threat-intelligence sharing [13, 14].

4.2. Correlation Rules

Correlation rules express a pattern across multiple events: a threshold, such as many failed authentications followed by a success; a sequence, such as a new service installed shortly after a remote logon; or a join between data sources. They are the traditional core of SIEM detection [1, 15]. Several studies organize correlation around attack-progression models such as the intrusion kill chain [16], using them to aggregate related alerts into fewer, richer alarms [17, 18], or to link suspicious information flows into a high-level attack scenario [19]. Correlation rules are explainable and fire with low latency, which makes triage straightforward. Their cost lies in maintenance. Thresholds and exclusions must be tuned to each environment, and rules written against one data schema break when sources or parsers change. Rule logic can also be evaded: an analysis of widely used Sigma rules showed that adversaries could easily evade nearly half of them [20]. Studies of production rulesets in managed SOCs find that a very small fraction of rules generates most alerts and incidents [21], and that design choices such as alert throttling and matching successful rather than attempted malicious actions substantially reduce analyst workload [22].

4.3. Behavioral and Statistical Analytics

Behavioral approaches, often marketed as user and entity behavior analytics (UEBA), build a baseline of normal activity for users, hosts or peer groups and flag deviations, such as rare processes, unusual logon times or abnormal data volumes [23]. Enterprise-scale examples include mining heterogeneous proxy, authentication and DHCP logs to flag suspicious host behavior [24], detecting early-stage infections from DNS and web-proxy logs [25], and modeling internal login paths to detect lateral movement [26]. These methods can surface novel attacks and insider activity that no rule anticipates. They depend on sufficient clean history to establish a baseline. In dynamic environments, legitimate change often appears anomalous, which leads to high false-positive rates when outputs are used directly as alerts rather than as context.

4.4. Machine-Learning Detection

Machine-learning approaches train supervised or unsupervised models on features extracted from log data [9, 27]. Deep-learning models of log sequences, such as DeepLog, learn normal execution patterns and flag deviations from them [28]; Landauer et al. review this line of work in detail [29], and an independent re-evaluation found that the accuracy reported for such models often does not hold under more realistic evaluation settings [30]. Supervised models require labeled attack data, which is scarce and rarely representative of a given environment. Several authors caution that machine learning in security is prone to specific pitfalls, including unrealistic evaluation datasets, data leakage, and a mismatch between the closed-world assumptions of classifiers and the open-world nature of intrusion detection [31, 32]. In operational SIEMs, machine-learning outputs are therefore most often used to prioritize or enrich alerts rather than to raise them independently, as in the user-centric framework that Feng et al. built for a production SOC to reduce false positives [33]. DeepCASE takes a related approach, using the context of preceding events to cluster security events and cut the number that analysts must inspect manually [34].

4.5. Hunt-Derived Detections

Threat hunting is a hypothesis-driven search for adversary activity that existing detections have missed. When a hunt identifies a reliable pattern, the pattern is converted into a scheduled detection, closing the gap permanently. This family is grounded in observed adversary behavior and tends to address techniques that are hard to detect, such as persistence and defense evasion. Its limitation is throughput, since each new detection depends on skilled manual investigation.

4.6. Cross-Cutting Practices

Four practices apply across all five families.

- **Detection-as-code.** Detections are stored in version control, reviewed through pull requests, tested automatically in continuous-integration pipelines and deployed programmatically. This brings software-engineering discipline to rule management and creates an audit trail.
- **Platform-neutral rule formats.** Sigma [8] allows a detection to be written once and converted to multiple query languages, reducing vendor lock-in and supporting community sharing.
- **Coverage mapping.** Detections are mapped to ATT&CK techniques [7, 35] to identify gaps. A technique with one mapped rule is not necessarily covered, however, because that rule may detect only one of many procedures that implement the technique. A study of four ATT&CK-annotated endpoint rulesets found that many techniques cannot be implemented as detection rules at all, and that claimed coverage of a technique does not consistently mean detecting the same threats [36].
- **Validation by simulation.** Adversary-emulation tools and breach-and-attack simulation execute known techniques in a controlled way, confirming that detections fire as intended and continue to do so after infrastructure changes.

5. Operational Challenges

Five challenges recur across the literature and in operational practice. Each is described below with the mitigations reported to date.

5.1. False Positives and Alert Fatigue

The most widely reported problem in SOC operations is alert volume. Qualitative studies with SOC analysts find that false positives dominate alert queues and that analysts often distrust the alerts they receive [4, 5]. A four-year measurement of 115 million network alerts in one SOC found that only about 0.01% were associated with true attacks [37], a consequence of the base-rate fallacy, which makes very low false-alarm rates essential when intrusions are rare [38]. Sustained high volume contributes to analyst burnout and turnover [6, 39]. Proposed mitigations include provenance-based triage, which uses causal relationships between events to rank alerts [40, 41]; risk-based alerting, which aggregates low-fidelity signals into a cumulative risk score per user or host and alerts only when a threshold is crossed; and automated enrichment through SOAR [42], which reduces manual effort per alert. AI-assisted alert screening has also been applied to filter and prioritize alerts [43]. None of these removes the need for well-tuned detection logic at the source.

5.2. Data Quality and Normalization

Detections depend on complete and correctly parsed data, yet data-quality failures are often silent. A log source that stops sending events, a parser that fails after a vendor upgrade, or inconsistent time-zone

handling can disable detections without generating any error. Kokulu et al. identify such technical and process issues as recurring sources of friction in SOCs [4]. Mitigations include continuous monitoring of log-source health, alerting on unexpected drops in event volume, and adoption of common schemas that decouple detections from vendor-specific formats [10].

5.3. Measuring Coverage and Effectiveness

Organizations frequently report detection maturity as the number of ATT&CK techniques covered. This measure overstates protection because it ignores the many procedures that implement each technique and the quality of individual rules. More meaningful metrics include the true-positive rate per detection, the ratio of alerts to confirmed incidents, mean time to detect, and the rate at which detections decay as environments change [44, 45]. Few published studies evaluate these metrics in production environments, largely because operational data is confidential.

5.4. Scale and Cost

Licensing models based on ingest volume force a trade-off between visibility and budget. Organizations respond by filtering data at the source, routing low-value data to cheaper storage tiers, or excluding sources entirely. These decisions are usually taken on cost grounds without systematic analysis of their detection impact, creating blind spots that are often discovered only after an incident.

5.5. Detection Content Migration

Organizations periodically replace their SIEM platform, for example when moving from a correlation-centric product to a search-centric or extended detection platform. Each migration requires translating hundreds or thousands of detections between rule languages that differ in data model, temporal semantics and correlation capability. Errors introduced during translation are difficult to detect, since a rule that never fires looks the same as a correct rule that has seen no matching activity. Despite its operational significance, detection migration has received little academic attention.

6. Practitioner Perspective

This section summarizes observations from more than ten years of SIEM engineering across enterprise and managed-security deployments on ArcSight, Splunk and Cortex XSIAM. Customer and environment details are withheld for confidentiality.

Vendor content requires substantial adaptation. Out-of-the-box detection content supplied with SIEM platforms is a useful starting point, but in our experience only a minority of it could be retained unchanged. Most rules needed tuning to the environment, such as adjusted thresholds and exclusions for known administrative activity, and a substantial share was disabled as irrelevant to the monitored estate or excessively noisy.

Tuning reduces alert volume substantially. Initial deployments routinely produced more alerts per day than the analyst team could triage. Structured tuning during the first weeks of operation, combining asset-criticality weighting, validated allow-lists and risk-based aggregation, reduced daily volume substantially while preserving the alerts that led to confirmed incidents.

Data sources fail more often than detection logic. The most frequent causes of missed detections were data problems rather than flawed rules. Recurring examples included endpoint agents that stopped reporting without notice, cloud audit logs that were never onboarded or arrived late, and firewall or proxy logs whose parsing broke after vendor upgrades. In each case the affected detections stayed enabled but could no longer fire.

Platform paradigm shapes detection design. Correlation rules on ArcSight favored real-time, stateful patterns. Scheduled searches on Splunk favored flexible, data-rich queries at the cost of latency. On XSIAM, integrated analytics shifted effort from writing rules toward tuning and validating built-in models. Moving detections between these paradigms required redesign rather than literal translation.

Managed-security settings add multi-tenant constraints. In a managed-security service, detections must work across customers with different data sources, naming conventions and risk tolerances. This favors parameterized, schema-based detections over environment-specific logic.

Five lessons follow from these observations:

1. Validate data completeness before writing or trusting a detection.
2. Manage detections as code, with version control, peer review and automated tests.
3. Measure detections by outcome, such as true-positive rate, rather than by coverage count.
4. Use risk-based aggregation to reduce alert volume before investing in automation.
5. Treat platform migration as a redesign project, with validation of every converted rule.

7. Open Research Directions

The review and practitioner observations point to six directions where research could most improve operational detection.

1. **Automated and verified detection migration.** Methods that translate detections between rule languages and verify semantic equivalence, for example by replaying labeled event sets through both versions, would reduce the risk of silent coverage loss during platform changes.
2. **Outcome-based coverage measurement.** Metrics that combine ATT&CK mapping with simulation results and production true-positive rates would describe protection more accurately than technique counts.
3. **Large language models in detection engineering.** Large language models may assist in drafting detections, translating rules, explaining alerts and summarizing investigations. Their tendency to produce plausible but incorrect logic calls for rigorous validation frameworks before operational use. Recent systematic reviews map the rapidly growing literature on LLMs for security tasks [46, 47].
4. **Public benchmark datasets.** Realistic, labeled, SIEM-scale datasets remain scarce, which limits reproducible research and fair comparison of detection methods [32, 48–50].
5. **Analyst-in-the-loop learning.** Triage decisions recorded in case-management systems are a large and largely unused source of labels that could drive automated tuning of detections.
6. **Regulatory alignment.** National frameworks, such as the Essential Cybersecurity Controls issued by Saudi Arabia's National Cybersecurity Authority [51], set monitoring requirements. Mapping such controls to concrete detection content remains an open practical problem.

8. Conclusions

Detection quality, not platform choice, determines whether a SIEM reduces risk. This survey described the SIEM detection lifecycle, proposed a five-family taxonomy of detection approaches mapped to MITRE ATT&CK, and analyzed the operational challenges that limit detection effectiveness, from alert fatigue to cross-platform migration. Combining the literature with a practitioner perspective shows that many problems practitioners face daily, such as data-quality failures and content migration, receive little academic attention. We hope the research directions identified here help close that gap between research and SOC practice.

Conflict of Interest

The author declares no conflicts of interest.

Acknowledgement

This research received no external funding. During the preparation of this manuscript, the author used Claude (Anthropic) for the purposes of drafting text, structuring the paper, and identifying and verifying references. The author has reviewed and edited the output and takes full responsibility for the content of this publication.

References

1. S. Bhatt, P.K. Manadhata, L. Zomlot, “The operational role of security information and event management systems”, IEEE Security & Privacy, 2014, 12, 35–41. https://doi.org/10.1109/MSP.2014.103
2. G. González-Granadillo, S. González-Zarzosa, R. Diaz, “Security information and event management (SIEM): Analysis, trends, and usage in critical infrastructures”, Sensors, 2021, 21, 4759. https://doi.org/10.3390/s21144759
3. M. Vielberth, F. Böhm, I. Fichtinger, G. Pernul, “Security operations center: A systematic study and open challenges”, IEEE Access, 2020, 8, 227756–227779. https://doi.org/10.1109/ACCESS.2020.3045514
4. F.B. Kokulu, A. Soneji, T. Bao, Y. Shoshitaishvili, Z. Zhao, A. Doupé, G.-J. Ahn, “Matched and mismatched SOCs: A qualitative study on security operations center issues”, Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS), London, UK, 2019, pp. 1955–1970. https://doi.org/10.1145/3319535.3354239
5. B.A. Alahmadi, L. Axon, I. Martinovic, “99% false positives: A qualitative study of SOC analysts' perspectives on security alarms”, Proceedings of the 31st USENIX Security Symposium, Boston, MA, USA, 2022, pp. 2783–2800.
6. S.C. Sundaramurthy, A.G. Bardas, J. Case, X. Ou, M. Wesch, J. McHugh, S.R. Rajagopalan, “A human capital model for mitigating security analyst burnout”, Proceedings of the Eleventh Symposium on Usable Privacy and Security (SOUPS 2015), Ottawa, ON, Canada, 2015, pp. 347–359.
7. B.E. Strom, A. Applebaum, D.P. Miller, K.C. Nickels, A.G. Pennington, C.B. Thomas, “MITRE ATT&CK: Design and Philosophy”, Technical Report, The MITRE Corporation: McLean, VA, USA, 2018.

8. SigmaHQ, “Sigma: Generic Signature Format for SIEM Systems”, Accessed 9 November 2026. https://github.com/SigmaHQ/sigma
9. A.L. Buczak, E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection”, IEEE Communications Surveys & Tutorials, 2016, 18, 1153–1176. https://doi.org/10.1109/COMST.2015.2494502
10. “Open Cybersecurity Schema Framework (OCSF)”, Accessed 9 November 2026. https://schema.ocsf.io
11. V. Mavroeidis, S. Bromander, “Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence”, Proceedings of the 2017 European Intelligence and Security Informatics Conference (EISIC), 2017, pp. 91–98. https://doi.org/10.1109/EISIC.2017.20
12. W. Tounsi, H. Rais, “A survey on technical threat intelligence in the age of sophisticated cyber attacks”, Computers & Security, 2018, 72, 212–233. https://doi.org/10.1016/j.cose.2017.09.001
13. C. Sillaber, C. Sauerwein, A. Mussmann, R. Breu, “Data quality challenges and future research directions in threat intelligence sharing practice”, Proceedings of the 2016 ACM Workshop on Information Sharing and Collaborative Security (WISCS), 2016, pp. 65–70.
14. T.D. Wagner, K. Mahbub, E. Palomar, A.E. Abdallah, “Cyber threat intelligence sharing: Survey and research directions”, Computers & Security, 2019, 87, 101589.
15. F. Valeur, G. Vigna, C. Kruegel, R.A. Kemmerer, “A comprehensive approach to intrusion detection alert correlation”, IEEE Transactions on Dependable and Secure Computing, 2004, 1, 146–169. https://doi.org/10.1109/TDSC.2004.21
16. E.M. Hutchins, M.J. Cloppert, R.M. Amin, “Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains”, Proceedings of the 6th International Conference on Information Warfare and Security (ICIW), Washington, DC, USA, 2011, pp. 113–125.
17. B.D. Bryant, H. Saiedian, “A novel kill-chain framework for remote security log analysis with SIEM software”, Computers & Security, 2017, 67, 198–210. https://doi.org/10.1016/j.cose.2017.03.003
18. B.D. Bryant, H. Saiedian, “Improving SIEM alert metadata aggregation with a novel kill-chain based classification model”, Computers & Security, 2020, 94, 101817. https://doi.org/10.1016/j.cose.2020.101817
19. S.M. Milajerdi, R. Gjomemo, B. Eshete, R. Sekar, V.N. Venkatakrishnan, “HOLMES: Real-time APT detection through correlation of suspicious information flows”, Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), 2019, pp. 1137–1152. https://doi.org/10.1109/SP.2019.00026
20. R. Uetz, M. Herzog, L. Hackländer, S. Schwarz, M. Henze, “You cannot escape me: Detecting evasions of SIEM rules in enterprise networks”, Proceedings of the 33rd USENIX Security Symposium, Philadelphia, PA, USA, 2024, pp. 5179–5196.
21. M. Vermeer, M. van Eeten, C. Gañán, “Ruling the rules: Quantifying the evolution of rulesets, alerts and incidents in network intrusion detection”, Proceedings of the 2022 ACM Asia Conference on Computer and Communications Security (ASIA CCS), 2022, pp. 799–813. https://doi.org/10.1145/3488932.3517412

22. K.T.W. Teuwen, T. Mulders, E. Zambon, L. Allodi, “Ruling the unruly: Designing effective, low-noise network intrusion detection rules for security operations centers”, Proceedings of the 20th ACM Asia Conference on Computer and Communications Security (ASIA CCS), Hanoi, Vietnam, 2025, pp. 1428–1441. https://doi.org/10.1145/3708821.3710823
23. V. Chandola, A. Banerjee, V. Kumar, “Anomaly detection: A survey”, ACM Computing Surveys, 2009, 41, 15. https://doi.org/10.1145/1541880.1541882
24. T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson, A. Juels, E. Kirda, “Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks”, Proceedings of the 29th Annual Computer Security Applications Conference (ACSAC), New Orleans, LA, USA, 2013, pp. 199–208. https://doi.org/10.1145/2523649.2523670
25. A. Oprea, Z. Li, T.-F. Yen, S.H. Chin, S.A. Alrwais, “Detection of early-stage enterprise infection by mining large-scale log data”, Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2015, pp. 45–56. https://doi.org/10.1109/DSN.2015.14
26. G. Ho, M. Dhiman, D. Akhawe, V. Paxson, S. Savage, G.M. Voelker, D. Wagner, “Hopper: Modeling and detecting lateral movement”, Proceedings of the 30th USENIX Security Symposium, 2021, pp. 3093–3110.
27. A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, “Survey of intrusion detection systems: Techniques, datasets and challenges”, Cybersecurity, 2019, 2, 20. https://doi.org/10.1186/s42400-019-0038-7
28. M. Du, F. Li, G. Zheng, V. Srikumar, “DeepLog: Anomaly detection and diagnosis from system logs through deep learning”, Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), 2017, pp. 1285–1298. https://doi.org/10.1145/3133956.3134015
29. M. Landauer, S. Onder, F. Skopik, M. Wurzenberger, “Deep learning for anomaly detection in log data: A survey”, Machine Learning with Applications, 2023, 12, 100470. https://doi.org/10.1016/j.mlwa.2023.100470
30. V.-H. Le, H. Zhang, “Log-based anomaly detection with deep learning: How far are we?”, Proceedings of the 44th International Conference on Software Engineering (ICSE), 2022, pp. 1356–1367. https://doi.org/10.1145/3510003.3510155
31. R. Sommer, V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection”, Proceedings of the 2010 IEEE Symposium on Security and Privacy, 2010, pp. 305–316.
32. D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, K. Rieck, “Dos and don'ts of machine learning in computer security”, Proceedings of the 31st USENIX Security Symposium, Boston, MA, USA, 2022, pp. 3971–3988.
33. C. Feng, S. Wu, N. Liu, “A user-centric machine learning framework for cyber security operations center”, Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), 2017, pp. 173–175. https://doi.org/10.1109/ISI.2017.8004902
34. T. van Ede, H. Aghakhani, N. Spahn, R. Bortolameotti, M. Cova, A. Continella, M. van Steen, A. Peter, C. Kruegel, G. Vigna, “DeepCASE: Semi-supervised contextual analysis of security events”, Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP), 2022, pp. 522–539.
35. B. Al-Sada, A. Sadighian, G. Oligeri, “MITRE ATT&CK: State of the art and way forward”, ACM Computing Surveys, 2025, 57, 12. https://doi.org/10.1145/3687300

36. A. Virkud, M.A. Inam, A. Riddle, J. Liu, G. Wang, A. Bates, “How does endpoint detection use the MITRE ATT&CK framework?”, Proceedings of the 33rd USENIX Security Symposium, Philadelphia, PA, USA, 2024, pp. 3891–3908.
37. L. Yang, Z. Chen, C. Wang, Z. Zhang, S. Booma, P. Cao, C. Adam, A. Withers, Z. Kalbarczyk, R.K. Iyer, G. Wang, “True attacks, attack attempts, or benign triggers? An empirical measurement of network alerts in a security operations center”, Proceedings of the 33rd USENIX Security Symposium, Philadelphia, PA, USA, 2024, pp. 1525–1542.
38. S. Axelsson, “The base-rate fallacy and the difficulty of intrusion detection”, ACM Transactions on Information and System Security, 2000, 3, 186–205. https://doi.org/10.1145/357830.357849
39. S.C. Sundaramurthy, J. McHugh, X. Ou, M. Wesch, A.G. Bardas, S.R. Rajagopalan, “Turning contradictions into innovations or: How we learned to stop whining and improve security operations”, Proceedings of the Twelfth Symposium on Usable Privacy and Security (SOUPS 2016), Denver, CO, USA, 2016, pp. 237–251.
40. W.U. Hassan, S. Guo, D. Li, Z. Chen, K. Jee, Z. Li, A. Bates, “NoDoze: Combatting threat alert fatigue with automated provenance triage”, Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 2019. https://doi.org/10.14722/ndss.2019.23349
41. W.U. Hassan, A. Bates, D. Marino, “Tactical provenance analysis for endpoint detection and response systems”, Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP), 2020, pp. 1172–1189. https://doi.org/10.1109/SP40000.2020.00096
42. C. Islam, M.A. Babar, S. Nepal, “A multi-vocal review of security orchestration”, ACM Computing Surveys, 2019, 52, 37. https://doi.org/10.1145/3305268
43. T. Ban, S. Ndichu, T. Takahashi, D. Inoue, “Combat security alert fatigue with AI-assisted techniques”, Proceedings of the Cyber Security Experimentation and Test Workshop (CSET), 2021, pp. 9–16. https://doi.org/10.1145/3474718.3474723
44. E. Agyepong, Y. Cherdantseva, P. Reinecke, P. Burnap, “Challenges and performance metrics for security operations center analysts: A systematic review”, Journal of Cyber Security Technology, 2020, 4, 125–152. https://doi.org/10.1080/23742917.2019.1698178
45. E. Agyepong, Y. Cherdantseva, P. Reinecke, P. Burnap, “A systematic method for measuring the performance of a cyber security operations centre analyst”, Computers & Security, 2023, 124, 102959. https://doi.org/10.1016/j.cose.2022.102959
46. H. Xu, S. Wang, N. Li, K. Wang, Y. Zhao, K. Chen, T. Yu, Y. Liu, H. Wang, “Large language models for cyber security: A systematic literature review”, ACM Transactions on Software Engineering and Methodology, 2025, arXiv:2405.04760. https://arxiv.org/abs/2405.04760
47. I. Hasanov, S. Virtanen, A. Hakkala, J. Isoaho, “Application of large language models in cybersecurity: A systematic literature review”, IEEE Access, 2024, 12, 176751–176778. https://doi.org/10.1109/ACCESS.2024.3505983
48. M. Landauer, F. Skopik, M. Frank, W. Hotwagner, M. Wurzenberger, A. Rauber, “Maintainable log datasets for evaluation of intrusion detection systems”, IEEE Transactions on Dependable and Secure Computing, 2023. https://doi.org/10.1109/TDSC.2022.3201582
49. I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization”, Proceedings of the 4th International Conference on

Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116. https://doi.org/10.5220/0006639801080116

50. N. Moustafa, J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)”, Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), 2015, pp. 1–6. https://doi.org/10.1109/MilCIS.2015.7348942
51. National Cybersecurity Authority, “Essential Cybersecurity Controls (ECC-2:2024)”, National Cybersecurity Authority: Riyadh, Saudi Arabia, 2024. Accessed 9 November 2026. https://nca.gov.sa/en/regulatory-documents/controls-list/ecc/