

Detecting Low-and-Slow Data Exfiltration: A Behavioural and Cumulative-Signal Approach to Identifying Slow-Moving Data Extraction

Satyabrat Maharana

Cyber Security Wing, Canara Bank, Bengaluru, Karnataka, India.

To Cite this Article: Satyabrat Maharana, "Detecting Low-and-Slow Data Exfiltration: A Behavioural and Cumulative-Signal Approach to Identifying Slow-Moving Data Extraction, "Indian Journal of Computer Science and Technology, Volume 05, Issue 02 (May-August 2026), PP: 1039-1042.



Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution License; Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: Most exfiltration controls are built to catch a single large transfer: a threshold on session size, a daily-volume cap, a DLP rule tuned to a file-size or record-count trigger. An adversary who instead moves data out in small increments over an extended period – kilobytes at a time, spread across days or weeks, often through channels that are individually unremarkable – can stay under every one of those thresholds indefinitely. This paper sets out a detection approach for that low-and-slow pattern. It argues that no single log source or single-session threshold is sufficient, and that reliable detection instead requires stitching together weak, individually-tolerable signals across a long observation window, using periodicity, cumulative volume against an adaptive baseline, and entity-relative behavioural scoring. A practical detection architecture is proposed, built primarily on SIEM correlation logic over proxy, DNS, endpoint and database logs, with worked detection logic, tuning guidance to control false positives, and a discussion of the specific evasion techniques – protocol choice, timing jitter, channel rotation – that a slow exfiltration campaign is likely to use against exactly this kind of detection.

Key Word: Data Exfiltration; Low-and-Slow; SIEM Correlation; Beaconing Detection; Behavioural Analytics; Insider Threat

I. INTRODUCTION

Data exfiltration detection has historically been framed as a volume problem: an attacker steals data, therefore a large amount of data crosses the perimeter, therefore watch for large transfers. This framing is correct for a meaningful share of real incidents, but it is a framing an attacker can trivially defeat once they are aware of it, simply by slowing down. A transfer of several gigabytes compressed into one session is conspicuous; the same volume moved in five-hundred sessions of a few kilobytes each, spread across six weeks, at a cadence that mimics ordinary application traffic, is not conspicuous to any control that evaluates one session, one day or one user-event at a time in isolation.

This pattern is commonly described as low-and-slow exfiltration. It is favoured by adversaries with persistent access – a compromised service account, a malicious or coerced insider, an implant with long-term command-and-control – precisely because persistence removes the need to move data quickly. The detection problem this creates is not a tooling gap so much as a framing gap: the signal that matters is not present in any single log line, but only in the accumulated pattern across many log lines over time, and most detection content is not built to look for that.

II.CHARACTERISTICS OF LOW-AND-SLOW EXFILTRATION

Three characteristics recur across low-and-slow exfiltration cases and distinguish them from bulk transfer, and each has a direct implication for detection design.

First, sub-threshold sizing: every individual transfer, query, file read or API call is deliberately kept below whatever volume or count threshold the environment is known or assumed to enforce. The implication is that no per-event threshold, however well-tuned, will trigger on any single event in the campaign – detection has to look at accumulation, not individual events.

Second, temporal spread with weak periodicity: transfers recur over an extended window – days, weeks, sometimes months – often at a roughly regular interval because the exfiltration is automated (a beaconing implant, a scheduled task, a scripted database dump), even though the interval is frequently jittered slightly to avoid being perfectly regular. The implication is that periodicity analysis over a long window, tolerant of jitter, is a stronger signal than any point-in-time volume check.

Third, channel and identity consistency: the same user, service account, process or destination tends to recur across the campaign, because standing up a new channel repeatedly adds operational cost and risk for the attacker. The implication is that entity-centric correlation – grouping weak signals by the account, host or destination they share – is what turns many individually-invisible events into one visible pattern.

III.WHERE THE SIGNAL LIVES

No single log source carries the full picture; the signal is distributed across network, endpoint, application and identity telemetry, and each source contributes a different piece of the pattern. The table below summarises the principal sources and why

volume-based bulk-transfer detection specifically fails to see the low-and-slow variant of the same behaviour in each.

Data source	Signal for low-and-slow exfiltration	Why bulk-transfer tools miss it
Proxy / firewall logs	Small, repeating outbound sessions to the same external endpoint at regular or near-regular intervals over days or weeks	Per-session byte counts stay well under any single-session volume threshold
DNS logs	High query rate to a narrow set of domains, long or high-entropy subdomain labels, TXT/NULL record use, queries at a steady cadence outside business hours	DNS is rarely volume-inspected and is commonly allowed outbound by default
Endpoint / DLP logs	Same user or process repeatedly touching sensitive files or database exports in small chunks, staging to temp directories before each outbound burst	Each individual read or copy is within normal-use thresholds
Database / application logs	Query patterns that page through a large table in small row ranges over an extended period, from an account or job that does not normally run such queries	Row-count and query-frequency alerting is usually tuned for single large pulls, not many small ones
Cloud / SaaS audit logs	Incremental API-driven downloads or sync operations spread across an unusually long session history for one identity	API calls are individually legitimate and rate-limited traffic looks routine

Table 1. Log sources, low-and-slow indicators, and why single-event volume checks miss them.

IV.WHY THRESHOLD-BASED DETECTION IS INSUFFICIENT

It is worth stating plainly why the obvious fix – simply lowering the volume threshold until it catches smaller transfers – does not work in practice. Lowering a per-session or per-day threshold far enough to catch kilobyte-scale transfers also catches the overwhelming majority of ordinary application traffic, telemetry, and legitimate small file transfers, producing an alert volume that is operationally unusable. The threshold that would catch a low-and-slow campaign sized to evade a 50 MB session cap is not a smaller number in the same dimension; it requires a different dimension entirely – accumulation over time and periodicity, rather than size at a point in time.

The comparison below sets out how different classes of detection logic perform against bulk versus low-and-slow exfiltration, to make explicit why a layered approach is necessary rather than a single improved threshold.

Detection approach	Catches bulk exfiltration	Catches low-and-slow exfiltration
Static per-session volume threshold	Yes	No – each session is below threshold
Daily aggregate volume per user/host	Yes, if aggregate crosses threshold same day	No – daily total stays inside normal range
Rolling-window baseline with seasonal adjustment	Yes	Partial – depends on window length relative to exfiltration cadence
Beaconing / periodicity analysis	Not primary purpose	Yes – designed for this pattern
Entity behaviour analytics (UEBA) with cumulative peer-relative scoring	Yes	Yes – accumulates weak signals across a long window

Table 2. Detection approaches compared against bulk and low-and-slow exfiltration patterns.

V.PROPOSED DETECTION ARCHITECTURE

The architecture proposed here is built on three layers that operate over progressively longer time windows and progressively more aggregated data, so that a campaign invisible at the shortest window becomes visible as the windows widen.

5.1 Layer 1 – Adaptive Cumulative Baselines

For each monitored entity – user, service account, host or process – the system maintains a rolling baseline of outbound volume, query count, or file-read count over multiple window lengths simultaneously (for instance 24 hours, 7 days and 30 days), rather than a single fixed window. A campaign sized to stay under the 24-hour baseline will typically still shift the 30-day cumulative total measurably above the entity's own historical norm, because the attacker is optimising against the shortest window they assume is being watched, not against all of them at once. Baselines should be entity-relative and seasonally adjusted – an accounts-payable clerk's month-end query volume is not a security anomaly – to keep the false-positive rate manageable.

5.2 Layer 2 – Periodicity and Beaconing Analysis

Independent of volume, the inter-event time intervals for a given entity-destination pair are analysed for regularity, using standard beaconing-detection techniques such as interval-variance scoring or FFT-based periodicity detection over the connection or query timestamps. A low but statistically regular cadence to the same external destination, sustained over many days, is scored as suspicious even when every individual transfer is trivially small, because legitimate human-driven traffic is rarely this regular over this long a period, whereas scripted or implant-driven exfiltration usually is.

5.3 Layer 3 – Entity-Relative Behavioural Scoring

The outputs of Layers 1 and 2, together with contextual signals – destination reputation and rarity, time-of-day relative to the entity's normal working pattern, whether the destination or channel has been used by this entity before, whether the account or process has recently changed privilege – are combined into a single cumulative risk score per entity, incremented over the observation window rather than reset each day. This is the layer that converts several individually-tolerable weak signals, each of which would be a false-positive-prone alert on its own, into one high-confidence composite alert once enough of them accumulate on the same entity within the same extended window.

VI. WORKED DETECTION LOGIC

As a concrete illustration of Layer 1 and Layer 2 working together, correlation logic of the following form is representative of what a SIEM use case for this pattern looks like in practice: for each internal source identity, compute the 30-day rolling sum of outbound bytes to each distinct external destination, alongside the count and coefficient of variation of the inter-connection intervals to that same destination over the same window. Flag the identity-destination pair when the 30-day cumulative volume exceeds a multiple of that identity's own 30-day historical baseline for the same destination category and the interval coefficient of variation falls below a regularity threshold consistent with scripted rather than human-driven access, excluding destinations already allow-listed as known backup, replication or SaaS-sync endpoints for that identity. The exclusion clause matters as much as the detection logic itself – without it, legitimate scheduled jobs (backup agents, log shippers, SaaS synchronisation) will dominate the alert queue, since they are, by design, small, regular and long-running.

VII. TUNING AND FALSE-POSITIVE MANAGEMENT

A detection layer built around accumulation over a 30-day window is inherently exposed to a higher nominal false-positive rate than a single-event threshold, simply because it is evaluating far more entity-days of data per alert. Three tuning practices keep this manageable in production. First, maintain a living allow-list of known recurring low-volume automated flows – backup jobs, replication, monitoring agents, SaaS sync – built from change-management records rather than from the SIEM inferring them itself, since an attacker-controlled process could otherwise get itself allow-listed by mimicking a legitimate one long enough. Second, weight the composite score in Layer 3 so that a change in destination, channel or timing pattern for an already-known recurring flow raises its score sharply, since drift in a previously-benign pattern is itself a strong signal, distinct from the pattern's mere existence. Third, route Layer 3 alerts to a tier-2 analyst workflow with the full 30-day entity timeline attached rather than a single event, so that triage time is spent confirming or dismissing a pattern rather than reconstructing one from scratch.

VIII. EVASION CONSIDERATIONS

A detection design should be built with its own evasion in mind. Against the architecture proposed here, an adversary aware of periodicity analysis will jitter timing more aggressively, which Layer 2 partially accounts for by using a variance threshold rather than requiring exact regularity, though sufficiently randomised timing will reduce this layer's contribution and shift more weight onto Layer 1. An adversary aware of cumulative-baseline analysis will rotate across multiple destinations or multiple lower-privileged identities to keep any single baseline from accumulating, which is the reason Layer 3 correlates across identities that share a host, a process lineage, or a source IP, rather than scoring identities in complete isolation. An adversary aware of DNS-based monitoring may prefer HTTPS to a rotating set of legitimate-looking cloud storage endpoints precisely because such destinations are hard to blanket-deny; this is best addressed operationally, through egress allow-listing for sensitive data stores rather than through detection alone, since detection is a compensating control and egress restriction is a preventive one.

IX. LIMITATIONS

The architecture depends on having sufficiently long log retention to populate 30-day (or longer) rolling windows, which has direct storage and SIEM-licensing implications and should be sized deliberately rather than left as a byproduct of whatever retention policy already exists for other purposes. It also depends on baseline quality: an entity with genuinely irregular or sparse historical activity will produce a noisy baseline that either misses real drift or over-alerts on normal variation, and new accounts or newly provisioned service accounts have no meaningful baseline at all for their first observation window. Finally, the approach is a detection layer, not a prevention layer – it is designed to surface a campaign that is already under way, and should be paired with preventive controls (egress filtering, data classification and DLP at the point of access, least-privilege on data stores) rather than relied on as the sole control against exfiltration.

X. CONCLUSION

Low-and-slow exfiltration is not defeated by a smaller threshold; it is defeated by changing what is being measured, from the size of one event to the accumulated behaviour of one entity over an extended window. The three-layer approach set out here – adaptive cumulative baselines, periodicity analysis, and entity-relative composite scoring – is intended to make that shift practical within a conventional SIEM-centric detection stack, without requiring new hardware or a wholesale replacement of existing volume-based controls, which remain useful for the bulk-transfer cases they were built for. The two approaches are complementary, not competing, and an organisation exposed to both attack patterns needs both.

REFERENCES

- [1]. MITRE ATT&CK, Exfiltration Over Alternative Protocol (T1048) and related sub-techniques.
- [2]. MITRE ATT&CK, Exfiltration Over C2 Channel (T1041) and Scheduled Transfer (T1029).
- [3]. SANS Institute, guidance on detecting beaconing behaviour and DNS-based exfiltration channels.
- [4]. Reserve Bank of India / CERT-In, cyber security and data-leakage prevention guidelines for regulated entities.