SIGNALFORGE CONSULTING

5 Critical Splunk Infrastructure Insights

Actionable Queries to Identify Performance Bottlenecks & Optimize Your Deployment

Run these queries on your Splunk instance today. Each insight uses the `_internal` index to reveal hidden performance issues, licensing concerns, and infrastructure pain points that could be impacting your operations.

1 License Usage Trending & Violation Risk

Identify if you're approaching your daily license limit and which sources are consuming the most data. Prevent unexpected license violations and optimize data ingestion.

SPLUNK QUERY

```
index=_internal source=*license_usage.log type="Usage"
                    | eval GB=b/1024/1024/1024
                    | timechart span=1d sum(GB) as daily_gb by pool
                    | appendcols [search index=_internal source=*license_usage.log
type="RolloverSummary" earliest=-30d
                    | head 1 | eval license_limit_gb=round(poolsz/1024/1024/1024,2) | fields
license_limit_gb]
                    | addtotals
                    | eval usage_pct=round((Total/license_limit_gb)*100,2)
```

⚠ What to Watch For:

- Daily usage exceeding 80% of license capacity
- Unexpected spikes in specific source types or pools
- Trending upward toward license violations

2 Indexer Queue Saturation & Bottlenecks

Detect when indexer queues are filling up, causing data delays or drops. Queue saturation is a leading indicator of indexer performance issues.

SPLUNK QUERY

```
index=_internal source=*metrics.log group=queue
                    | eval fill_pct=round((current_size/max_size)*100,2)
                    | stats avg(fill_pct) as avg_fill max(fill_pct) as max_fill by name host
                    | where max_fill > 70
                    | sort -max_fill
```

⚠ What to Watch For:

- Queues consistently above 70% capacity (especially parsingQueue, aggQueue)
- Specific indexers with higher queue fill than others (load imbalance)

3 Search Performance & Resource Hogs

Identify slow searches and users running inefficient queries that consume excessive resources. Optimize search

performance and reduce infrastructure strain.

SPLUNK QUERY

```
index=_audit action=search info=completed
                    | eval search_duration=total_run_time
                    | where search_duration > 60
                    | stats count avg(search_duration) as avg_duration
                    max(search_duration) as max_duration by user, savedsearch_name
                    | sort -avg_duration
                    | head 20
```

⚠ What to Watch For:

- Searches taking over 5 minutes regularly
- Scheduled searches that could be optimized or time-range adjusted
- Users running unoptimized ad-hoc searches

4 Forwarder Connection Health & Data Gaps

Monitor forwarder connectivity to detect data collection gaps. Missing forwarders can indicate infrastructure failures or network issues.

SPLUNK QUERY

```
index=_internal source=*metrics.log group=tcpin_connections
                    | stats latest(_time) as last_seen by hostname
                    | eval hours_since=round((now()-last_seen)/3600,1)
                    | where hours_since > 1
                    | sort -hours_since
                    | eval last_seen_time=strftime(last_seen,"%Y-%m-%d %H:%M:%S")
```

⚠ What to Watch For:

- Forwarders not seen in over 4 hours (potential data loss)
- Critical servers missing from the list

5 Indexer Disk Space & Retention Issues

Track disk utilization across indexers to prevent storage exhaustion. Running out of disk space can halt indexing and cause data loss.

SPLUNK QUERY

```
index=_introspection component=Partitions
                    | eval usage_pct=round((capacity_used/capacity)*100,2)
                    | stats latest(usage_pct) as disk_usage latest(capacity) as total_gb
latest(capacity_used) as
                    used_gb by host data
                    | where disk_usage > 70
                    | sort -disk_usage
```

⚠ What to Watch For:

- Disk usage above 85% (critical threshold)
- Rapid growth trends indicating need for retention policy review
- Imbalanced storage across indexer cluster members