AI & Cybersecurity

SOC PROMPT PACK

106 AI Prompts for SOC & Security Teams

Triage alerts faster, analyse logs, write queries, investigate phishing and malware, run incidents and report clearly.

Version 1.0 · November 2026 · Licensed to the purchaser for personal or internal business use. Do not redistribute.
For defensive use on systems you own or are authorised to manage. Test changes in a non-production environment first.

How to use this pack

106 ready-to-use prompts across 10 everyday SOC tasks. They work with ChatGPT, Claude, Gemini, Copilot for Security and most enterprise AI assistants.

1. Fill the placeholders

Anything in `{curly_braces}` is yours to replace: paste the alert, log lines, query language (KQL, SPL, ES|QL), your tech stack and so on.

2. Add a role line for better answers

```
You are a senior SOC analyst at a {industry} company running {stack}.
Be precise. If you are unsure, say so. Do not invent log fields or IOCs.
```

3. Verify everything

AI is a fast junior analyst: great at summarising and drafting, but it can be confidently wrong. Check CVEs against vendor advisories and CISA KEV, test queries before relying on them, and never close an incident on AI output alone.

> **Data safety rules: read before you paste**
>
> - Only paste company data into AI tools your organisation has **approved** for that data class.
> - Remove or mask passwords, API keys, session tokens and personal data (names, emails, SSNs, health data).
> - Replace internal hostnames and IPs with placeholders (e.g. HOST-A, 10.x.x.1) if policy requires.
> - Never paste live malware into a chat; use hashes, sandbox summaries or defanged text.

4. Chain prompts

Best results come from sequences: **Triage** (section 1) → **Log analysis** (2) → **Query** (3) → **Incident response** (8) → **Report** (9).

Contents

Section	Prompts
1. Alert triage	11
2. Log analysis	11

3. Query writing (SIEM)	11
4. Threat intelligence	10
5. Phishing analysis	11
6. Malware & scripts	10
7. Vulnerability management	10
8. Incident response	11
9. Reporting & communication	10
10. Policies, hardening & automation	11
Total	**106**

1. Alert triage

#001 · Triage a single alert

```
You are a Tier-2 SOC analyst. Here is an alert from {tool}:
{alert}
Explain in plain English what it means, list 3 benign and 3 malicious explanations,
rate likelihood of true positive (Low/Med/High) with reasons, and give the next 5
investigation steps.
```

#002 · Is this a false positive?

```
Review this alert and the surrounding context. List the specific evidence that would
confirm it is a false positive and the evidence that would confirm it is malicious.
Then tell me which data sources to query to decide.
Alert:
{alert}
Context:
{context}
```

#003 · Prioritise an alert queue

```
Here are {n} open alerts (one per line: id, rule name, host, user, severity, time).
Rank them for investigation order considering asset criticality ({critical_assets}),
privilege of the user and kill-chain stage. Output a table: rank, id, reason (one
line).
{alerts}
```

#004 · Map an alert to MITRE ATT&CK

```
Map this alert to the most likely MITRE ATT&CK tactic(s) and technique IDs. Explain
why, and list what the attacker would most likely do next and which telemetry would
show it.
{alert}
```

#005 · Explain a detection rule

```
Explain what this detection rule looks for, what attacker behaviour it catches, common false-positive sources, and how an attacker might evade it.
{rule}
```

#006 · Suggest enrichment

```
For this alert, list the enrichment lookups I should run (IP reputation, domain age, user history, asset owner, process tree, etc.), in priority order, and what result would raise or lower severity.
{alert}
```

#007 · Write the triage note

```
Write a concise triage note for the ticket (max 120 words) using this structure:
Summary · Evidence reviewed · Assessment (TP/FP/Benign-TP) · Action taken · Next step.
Facts:
{facts}
```

#008 · Escalation decision

```
Based on these findings, should this be escalated to an incident? Apply this criteria: {criteria}. Answer Yes/No first, then justify in 3 bullet points.
{findings}
```

#009 · Close as benign

```
Draft a closure comment explaining why this alert is a benign true positive, what business activity caused it, and a tuning suggestion to reduce repeats.
{details}
```

#010 · Spot alert fatigue patterns

```
Here is a 30-day export of alert counts by rule name and disposition. Identify the 5 noisiest rules, their false-positive rate, and a specific tuning idea for each.
{csv}
```

#011 · Shift handover

```
Summarise these open investigations into a shift handover: for each, status, what's
been done, what's pending, and any deadline. Keep it under 200 words.
{notes}
```

2. Log analysis

#012 · Summarise auth logs

```
Analyse these Linux auth.log lines. Report: failed vs successful logins by user and
source IP, any brute-force pattern (many failures then success), sudo usage, and
anything unusual. Output a short summary then a table.
{logs}
```

#013 · Windows logon events

```
Analyse these Windows Security events (4624/4625/4672/4648). Identify logon types,
unusual source workstations, off-hours activity and possible pass-the-hash or lateral
movement indicators.
{events}
```

#014 · Find the needle

```
From these logs, find entries related to {indicator}. Show the matching lines, the
timeline, and any related activity within ±10 minutes.
{logs}
```

#015 · Web server attack patterns

```
Review these web access logs. Flag likely SQL injection, path traversal, XSS, scanner
user-agents, credential stuffing and unusual 4xx/5xx bursts. Group by source IP with
counts.
{logs}
```

#016 · Firewall log anomalies

```
Analyse these firewall logs. Identify top talkers, denied connections to sensitive
ports, possible port scans (one source → many ports), and outbound traffic to unusual
countries or ports.
{logs}
```

#017 · DNS log hunting

```
Review these DNS query logs for signs of DGA domains, DNS tunnelling (long/random subdomains, high TXT volume), newly seen domains and known-bad TLDs. Explain each finding.
{logs}
```

#018 · Build a timeline

```
Create a chronological timeline (UTC) from these mixed log sources. Columns: time, source, host, user, event, significance. Highlight the first suspicious event.
{logs}
```

#019 · Normalise messy logs

```
Convert these unstructured log lines into a JSON array with fields: timestamp (ISO 8601), host, user, src_ip, dst_ip, action, message. Use null where missing.
{logs}
```

#020 · Explain an unfamiliar log

```
I don't recognise this log format/source. Explain what system likely produced it, what each field means, and which fields matter for security.
{sample}
```

#021 · Baseline normal

```
Here is a week of {log_type} logs from a normal period. Describe the baseline: typical volumes, hours, users, sources. Then list 5 deviations that would be worth alerting on.
{logs}
```

#022 · PowerShell log review

```
Review these PowerShell script block logs (event 4104). Decode any Base64, flag obfuscation, download cradles, AMSI bypass attempts and credential access. Rate each block Benign/Suspicious/Malicious.
{logs}
```

3. Query writing (SIEM)

#023 · Natural language to KQL

```
Write a Microsoft Sentinel KQL query that {goal}. Use table {table}. Add comments
explaining each line and note any performance considerations.
```

#024 · Natural language to SPL

```
Write a Splunk SPL search that {goal} using index={index} sourcetype={sourcetype}.
Include a stats/table output suitable for a dashboard.
```

#025 · Elastic / ES|QL query

```
Write an Elastic query ({language}) to {goal} across {index_pattern}. Explain the
fields assumed.
```

#026 · Translate a query

```
Translate this {source_lang} query into {target_lang}, keeping the same logic. Point
out any functions that don't map 1:1.
{query}
```

#027 · Debug my query

```
This {lang} query returns {problem}. Find the bug, explain it, and give the corrected
query.
{query}
```

#028 · Optimise a slow query

```
Optimise this {lang} query for speed on a large dataset (filter early, limit fields,
avoid expensive joins). Show the improved version and explain the changes.
{query}
```

#029 · Detection for a technique

```
Write a detection query in {lang} for MITRE ATT&CK {technique_id} ({technique_name}) using {data_source}. Include expected false positives and a tuning tip.
```

#030 · Brute-force then success

```
Write a {lang} query that finds accounts with ≥{n} failed logins followed by a successful login from the same IP within {minutes} minutes.
```

#031 · Impossible travel

```
Write a {lang} query that detects the same user signing in from two countries within {hours} hours, excluding known VPN egress IPs in {watchlist}.
```

#032 · Rare process

```
Write a {lang} query that finds processes executed on fewer than {n} hosts in the last 30 days that ran in the last 24 hours, excluding signed Microsoft binaries if the field exists.
```

#033 · Dashboard panel ideas

```
Suggest 8 dashboard panels for a {audience} SOC dashboard using {data_sources}. For each give the panel title, chart type and the {lang} query.
```

4. Threat intelligence

#034 · Summarise a threat report

```
Summarise this threat intelligence report for a SOC team: actor, targets, initial access, key TTPs (with ATT&CK IDs), IOCs, and 5 detection or hunting ideas relevant to an environment running {stack}.
{report}
```

#035 · Extract IOCs

```
Extract all indicators of compromise from this text into a table: type (IP, domain, URL, hash, email, file name, registry key), value, and context. Defang URLs and IPs.
{text}
```

#036 · Are we exposed?

```
Given this vulnerability/threat advisory and our environment ({environment}), assess whether we're likely exposed, what to check first, and the urgency (patch now / this week / monitor).
{advisory}
```

#037 · Brief for executives

```
Turn this threat intel into a 5-sentence executive brief: what's happening, who's targeted, likelihood it affects us ({industry}), business impact, and what we're doing.
{intel}
```

#038 · Actor profile

```
Create a one-page profile of threat actor {actor}: aliases, motivation, typical targets, notable campaigns, common TTPs with ATT&CK IDs, and recommended defences. Flag anything you are unsure about.
```

#039 · CVE explainer

```
Explain {cve_id} in plain English: affected products/versions, how it is exploited, whether it's known to be exploited in the wild, and mitigation steps. Remind me to verify against the vendor advisory and CISA KEV.
```

#040 · Hunt hypotheses from intel

```
From this report, write 5 threat-hunting hypotheses in the form 'If {actor} is in our network, we would see X in Y data source'. Include a query idea for each.
{report}
```

#041 · Compare two reports

```
Compare these two threat reports. What overlaps (TTPs, infrastructure, IOCs), what's new, and does it suggest the same actor?
{report_a}
---
{report_b}
```

#042 · Weekly intel digest

```
Turn these {n} news items into a weekly threat digest: top 3 items relevant to {industry}, one paragraph each, plus a 'what to do this week' list.
{items}
```

#043 · IOC to blocklist format

```
Convert these IOCs into {format} (e.g. firewall blocklist, Sentinel watchlist CSV, STIX 2.1 indicators). Exclude private IPs and common legitimate domains.
{iocs}
```

5. Phishing analysis

#044 · Analyse a suspicious email

```
Analyse this email (headers + body). Check sender vs reply-to, SPF/DKIM/DMARC results,
display-name spoofing, lookalike domains, urgency cues and links. Verdict: Phishing /
Suspicious / Legitimate, with evidence.
{email}
```

#045 · Decode email headers

```
Explain these email headers hop by hop. Identify the true originating server and IP,
authentication results and any red flags.
{headers}
```

#046 · Lookalike domain check

```
Is {domain} a lookalike of {our_domain}? Explain the technique used (typosquatting,
homoglyph, combosquatting, TLD swap) and suggest 10 other lookalikes we should
monitor.
```

#047 · Analyse a URL safely

```
Without visiting it, analyse this URL's structure: domain, subdomains, path,
parameters, redirects, encoding and what it is likely trying to do. Defang it in your
answer.
{url}
```

#048 · User reply template

```
Write a friendly reply to a user who reported this email. Tell them whether it was
malicious, thank them, and give one tip. Under 80 words.
Verdict: {verdict}
```

#049 · Company-wide warning

```
Write a short internal warning about an active phishing campaign: what it looks like, subject lines, what to do if you clicked. Calm tone, under 150 words.
Details: {details}
```

#050 · Clicked — now what?

```
A user clicked a phishing link and entered their password at {time}. Give me a step-by-step response checklist for M365/Google Workspace, in order, including session revocation, MFA review, mailbox rule checks and sign-in log review.
```

#051 · Find all recipients

```
Write a {lang} query to find every mailbox that received an email with subject '{subject}' or from sender {sender} in the last {days} days, and whether anyone clicked.
```

#052 · BEC red flags

```
Review this conversation thread for business email compromise indicators (payment change requests, tone shifts, new reply-to, thread hijacking). Summarise risk.
{thread}
```

#053 · Training quiz from real phish

```
Turn this real (sanitised) phishing email into a 3-question training quiz with answers and explanations.
{email}
```

#054 · Attachment triage

```
Based on this attachment metadata and sandbox summary, is it malicious? Explain the behaviour observed and the IOCs to block.
{sandbox_output}
```

6. Malware & scripts

#055 · Explain a suspicious script

```
Explain step by step what this {language} script does. Flag any malicious behaviour (download, persistence, credential theft, defence evasion) and list IOCs. Do not execute or improve it.
{script}
```

#056 · Deobfuscate safely

```
Statically deobfuscate this command line/script (Base64, char codes, string concatenation). Show each decoding step and the final readable command. Do not run it.
{blob}
```

#057 · Process tree review

```
Review this process tree from EDR. Is the parent-child chain normal? Highlight suspicious relationships (e.g. Office spawning PowerShell) and explain.
{tree}
```

#058 · Persistence hunt

```
List the common Windows persistence locations (Run keys, services, scheduled tasks, WMI, startup folder) and give a PowerShell command to list each read-only. Then review my output for anything suspicious.
{output}
```

#059 · YARA rule draft

```
Write a YARA rule to detect this sample family based on these strings and characteristics. Explain each condition and the false-positive risk.
{indicators}
```

#060 · Sandbox report summary

```
Summarise this sandbox report for a ticket: family (if known), behaviour, network IOCs, file/registry changes, and recommended containment.
{report}
```

#061 · Hash lookup plan

```
I have these file hashes from an incident. Tell me what sources to check (VirusTotal, internal EDR, threat intel), what each result would mean, and how to scope affected hosts.
{hashes}
```

#062 · Living-off-the-land check

```
Is this use of {binary} likely legitimate or LOLBin abuse? Explain typical admin use vs attacker use and what extra context to check.
{command_line}
```

#063 · Ransomware note analysis

```
Analyse this ransom note (text only): identify the likely family, extension, contact method and whether a free decryptor might exist. Remind me to verify on No More Ransom.
{note}
```

#064 · Containment options

```
Given this malware behaviour, compare containment options (isolate host, block hash, block domain, disable user) with pros/cons and the order to apply them.
{behaviour}
```

7. Vulnerability management

#065 · Prioritise scan results

```
Here are vulnerability scan results (CSV: host, CVE, CVSS, title, exposed-to-internet,
asset criticality). Prioritise using exploitability (CISA KEV, public exploit),
exposure and criticality, not CVSS alone. Output top 15 with reasoning.
{csv}
```

#066 · Explain to an asset owner

```
Explain {vulnerability} to a non-technical system owner: what it is, real-world risk
to their system, what they need to do, and expected downtime. Under 150 words.
```

#067 · Fix steps

```
Give exact remediation steps for {finding} on {os_or_app}, including how to verify the
fix and a rollback plan.
```

#068 · Compensating controls

```
We cannot patch {vulnerability} on {system} for {reason}. Suggest compensating
controls ranked by effectiveness and effort.
```

#069 · Group findings

```
Group these {n} findings by root cause (e.g. missing patch, weak config, outdated
software) so we can fix many with one change. Show counts per group.
{findings}
```

#070 · Write a remediation ticket

```
Write a remediation ticket: title, affected assets, risk summary, steps, acceptance
criteria, due date based on severity SLA ({sla}).
Finding: {finding}
```

#071 · SLA policy draft

```
Draft a vulnerability remediation SLA table for a {company_size} company: severity definitions, internet-facing vs internal deadlines, exception process.
```

#072 · Pen test report digest

```
Summarise this penetration test report into: 5 most important issues, quick wins (<1 day), and items needing budget. Keep findings IDs.
{report}
```

#073 · Risk acceptance memo

```
Draft a risk acceptance memo for {finding}: description, business reason, compensating controls, residual risk, owner, review date.
```

#074 · Trend summary

```
Compare these two monthly vulnerability summaries and describe the trend (new, fixed, overdue, mean time to remediate). Suggest 3 improvements.
{month_a}
{month_b}
```

8. Incident response

#075 · First 30 minutes

```
We suspect {incident_type} on {scope}. Give a first-30-minutes checklist: who to call, what to preserve, what to contain, what NOT to do.
```

#076 · Scoping questions

```
List the key scoping questions for this incident (who, what, when, where, how, data involved) and which logs answer each.
{summary}
```

#077 · Containment plan

```
Draft a containment plan with options for short-term and long-term containment for this incident, including business impact of each step.
{details}
```

#078 · Compromised account playbook

```
Write a step-by-step playbook for a compromised {platform} user account: verification, containment, investigation queries, recovery, and user communication.
```

#079 · Ransomware playbook

```
Write a ransomware response playbook for a {company_size} organisation: detection, isolation, decision points (restore vs other), communications, legal/insurance notifications, recovery verification.
```

#080 · Evidence checklist

```
List the evidence to collect for {incident_type} on {os}, in order of volatility, with the tool/command for each and chain-of-custody notes.
```

#081 · Incident status update

```
Write a status update for stakeholders: current status, impact, actions in the last {hours} hours, next steps, next update time. Max 150 words.
Notes: {notes}
```

#082 · Root cause analysis

```
Based on these findings, draft a root cause analysis using the '5 Whys' method and identify contributing factors (people, process, technology).
{findings}
```

#083 · Post-incident review agenda

```
Create a 45-minute post-incident review agenda with questions for each phase (detection, response, communication, recovery) and an action-tracking table.
```

#084 · Lessons-learned actions

```
Turn these lessons learned into SMART actions with owner role and target date.
{lessons}
```

#085 · Tabletop scenario

```
Write a 60-minute tabletop exercise for {scenario} with 4 injects, discussion questions for each, and what a good response looks like.
```

9. Reporting & communication

#086 · Executive incident summary

```
Write an executive summary of this incident (max 200 words): what happened, business impact, what we did, current risk, decisions needed. No jargon.
{technical_notes}
```

#087 · Monthly SOC report

```
Turn these metrics into a monthly SOC report: headline numbers, trends vs last month, top 3 incidents, improvements made, next month's focus.
{metrics}
```

#088 · Explain risk to the board

```
Explain {risk} to a board audience in 5 bullet points focusing on likelihood, impact in money/reputation, and what investment reduces it.
```

#089 · Rewrite for clarity

```
Rewrite this technical finding so a manager can act on it. Keep facts, remove jargon, add a clear 'what we need from you'.
{text}
```

#090 · Customer security answer

```
Draft a response to this customer security questionnaire question based on our controls. Be accurate, don't over-claim.
Question: {question}
Our controls: {controls}
```

#091 · Audit evidence summary

```
Summarise how this evidence demonstrates control {control_id} ({framework}). Note any gaps an auditor might raise.
{evidence}
```

#092 · Breach notification draft

```
Draft a customer breach notification letter for {incident} covering: what happened, what information was involved, what we are doing, what they can do, contact info. Mark it DRAFT – legal review required.
```

#093 · KPI ideas

```
Suggest 10 meaningful KPIs/KRIs for a {team_type} team, with definitions, data source and target.
```

#094 · Slide outline

```
Create a 6-slide outline for presenting {topic} to {audience}: slide title, 3 bullets each, and a suggested chart.
```

#095 · Plain-English glossary

```
Create a plain-English glossary of these security terms for non-technical staff, one sentence each: {terms}.
```

10. Policies, hardening & automation

#096 · Policy gap check

```
Compare this policy against {framework} requirements for {area}. List gaps and suggest wording to close each.
{policy}
```

#097 · Hardening review

```
Review this {system} configuration against security best practice. List risky settings with severity and the recommended value.
{config}
```

#098 · Firewall rule review

```
Review these firewall rules. Flag any-any rules, exposed admin ports, shadowed/duplicate rules and missing logging. Suggest a cleaned-up rule set.
{rules}
```

#099 · Cloud misconfiguration

```
Review this {cloud} IAM policy/resource config for over-permissive access, public exposure and missing encryption. Suggest least-privilege changes.
{json}
```

#100 · Write a SOAR playbook outline

```
Outline a SOAR playbook for {use_case}: trigger, enrichment steps, decision branches, automated actions, human approval points.
```

#101 · Script a safe audit

```
Write a read-only {language} script that checks {setting} across {scope} and outputs a CSV. It must not change anything. Add comments.
```

#102 · Regex for parsing

```
Write a regex to extract {fields} from these log lines. Explain it and show the matches.
{samples}
```

#103 · Sigma rule

```
Write a Sigma rule for {behaviour} using {logsource}. Include title, status, description, detection, falsepositives and level.
```

#104 · Review my detection

```
Review this detection rule for logic errors, evasion gaps and noise. Suggest improvements.
{rule}
```

#105 · Security awareness tip

```
Write a short, friendly security tip of the week about {topic} for all staff (under 90 words) with one action they can take today.
```

#106 · Interview prep

```
Ask me 10 interview questions for a {role} security role, one at a time, then give feedback on each answer I give.
```

Build your own prompts

Use this structure for any new task:

```
ROLE:    You are a {role} at a {industry} organisation.
TASK:    {what you want done}
INPUT:   {data, delimited by triple quotes}
RULES:   Don't invent facts. Say "unknown" if data is missing. Cite line numbers.
OUTPUT:  {format: table / JSON / 120-word summary / numbered steps}
```

Free updates to this pack are emailed to the address you purchased with. Questions: **contact@aiandcybersecurity.com**