Hands-on Cybersecurity Career Playbook

Как техническому специалисту продать себя дороже, не уходя в менеджмент

ДЛЯ КОГО МИНИ БРОШЮРА

Junior / Middle / Middle+ / Senior, которые работают руками и хотят перейти на более дорогую позицию.
AppSec, DevSecOps, Product Security, Vulnerability Management, Pentest, Cloud Security, Security Automation, SOC / IR.
Для тех, кто не стремится в Team lead\C-level, но хочет, чтобы его technical output был понятен рынку и стоил дороже.

ГЛАВНАЯ МЫСЛЬ:

Тебя могут покупать как executor. Это нормально. Но даже executor должен уметь показать stack, scope, evidence, impact и цену своей работы. Не надо изображать директора если ты по факту не он. Можно заработать и на своей должности, продать свой скилл дороже, свою экспертизу. Потребуется доказать, что ты сильный hands-on инженер, который быстро врубается, закрывает реальные риски, автоматизирует рутину и не ломает delivery.

0. Аннотация

Этот документ — fork от более leadership-oriented playbook, но здесь фокус другой: hands-on позиции. Не “как стать директором”, а как человеку, который реально пишет код, крутит пайплайны, разбирает findings, чинит CI/CD, смотрит логи, гоняет scanners, делает pentest или закрывает vulns, продать себя дороже и перейти на работу с лучшим compensation.

Главная логика: на международном рынке платят не за список тулов, а за доказуемый technical output. Работодатель покупает не “человека, который запускал SAST”, а человека, который умеет встроить SAST в pipeline, снизить false positives, поднять coverage, ускорить triage и не взорвать разработчиков лишним шумом.

Этот документ **распространяется бесплатно и не предназначен для коммерческого использования**, **перепродажи или выдачи за собственный платный продукт**. Его цель — помочь русскоязычным ребятам качественнее подготовиться к интервью, эффективнее пройти стейджи найма и получить более выгодный оффер, а значит заработать больше, а значит и сделать свою жизнь чуть лучше!

Данный материал — это личная компиляция: мой практический опыт, опыт коллег, рабочие ситуации, идеи, паттерны и best practices из учебников, книг, курсов, блогов, форумов и реальных историй.

Полная персональная подготовка к интервью, быстрый разбор английского под конкретную роль, корректировка self-introduction, упаковка CV, формирование STAR, корректировка LinkedIn, mock interviews и индивидуальная траектория обучения **доступны только в формате one-on-one paid sessions.**

Эта брошюра распространяется бесплатно как базовая практическая опора.

> **Что тебе можно сделать уже сейчас**
>
> Возьми свой последний проект и прогони его через главы 3, 4, 5 и 7. Выпиши 5–7 technical stories, затем перепиши CV bullets по формуле Action + Scope + Evidence + Impact. После этого собери self-intro на 45 секунд и 3 STAR-ответа для интервью.

ОГЛАВЛЕНИЕ

1. Твой новый mindset: не “винтик” в системе, а predictable executor

В СНГ\РФ многие технари привыкли говорить о себе через обязанности: “занимался анализом уязвимостей”, “администрировал сканер”, “участвовал в настройке CI/CD”. На международном рынке такая подача звучит слабо: hiring manager не понимает, какой риск ты снял, какой объём держал и насколько тебе можно доверять без постоянного контроля.

Правильный hands-on narrative строится не вокруг громких титулов, а вокруг надёжности: ты берёшь кусок технической боли, превращаешь хаос в процесс, делаешь результат повторяемым и оставляешь после себя систему, которую можно поддерживать.

Старый паттерн	Новый паттерн
“Я выполнял задачи по безопасности”	“Я закрывал конкретный класс риска и улучшал measurable security coverage”
“Я работал с инструментами”	“Я внедрял, настраивал и доводил tooling до usable state для команды”
“Я не менеджер, просто делал руками”	“Я hands-on engineer с ownership за конкретный technical outcome”
“Мне поручали, я делал”	“Я сам находил bottlenecks, предлагал fixes и доводил до результата”

Формула дорогого executor:

- сильный technical stack без расплывчатых слов;
- понятный scope: сколько apps, repos, services, cloud accounts, findings, hosts, pipelines;
- evidence: примеры артефактов, reports, dashboards, PRs, scripts, rules, playbooks;
- impact: latency, coverage, false positives, MTTR, SLA, risk reduction, automation rate;
- адекватная коммуникация: не токсичный “гений”, а взрослый инженер, с которым можно работать.

2. Что покупает работодатель: stack, scope, evidence, autonomy

Для hands-on ролей тебя оценивают по четырём слоям. Если в CV виден только первый слой — tools — ты становишься commodity. Если видны все четыре, ты становишься кандидатом дороже рынка.

Слой	Что показать	Как доказать
Stack	Языки, scanners, CI/CD, cloud, IaC, контейнеры, SIEM, EDR, API security, threat modeling	Skills section + bullets с конкретными toolchains и контекстом применения
Scope	Масштаб: apps, repos, microservices, pipelines, CVEs, hosts, endpoints, cloud accounts, teams	Цифры диапазонами: 20+ repos, 150+ findings/month, 8 product teams
Evidence	Артефакты: scripts, rules, dashboards, playbooks, reports, detections, PRs, training snippets	GitHub, sanitized screenshots, portfolio, “example upon request”
Autonomy	Можешь сам поставить задачу, разобраться, договориться с dev/team lead и довести до fix	STAR stories: issue → action → validation → result

Хороший CV bullet для technical role обычно отвечает на 5 вопросов: что сделал, где, чем, в каком масштабе и что стало лучше.

Не надо притворяться стратегом
Если ты hands-on, твоя сила в том, что ты можешь руками закрыть gap. Но даже такой gap надо называть языком результата: “уменьшил backlog critical vulns”, “сократил triage time”, “поднял coverage SAST/DAST/SCA”, “снизил шум от scanners”.

3. Technical value map: как переводить обычную работу в value

Технический специалист часто думает: “я просто настроил scanner”. Но для компании это может означать: меньше production defects, меньше ручного triage, быстрее release, выше compliance readiness и меньше вероятность breach. Твоя задача — не фантазировать про миллионы долларов, а честно связать technical action с понятным outcome.

Technical action	Что это значит для компании	Как записать в CV
Внедрил SAST/SCA в CI/CD	Vulns ловятся раньше, дешевле исправлять, меньше security debt	Integrated SAST/SCA into CI/CD for 30+ repos, increasing automated security coverage before release
Настроил rules / suppressions	Меньше false positives, меньше раздражения dev teams	Reduced scanner noise by tuning rules and suppression workflow, improving developer adoption
Собрал vuln dashboard	Появился единый источник правды и контроль SLA	Built vulnerability dashboard to track severity, ownership and remediation SLA across product teams
Автоматизировал report/export	Меньше ручной рутины, быстрее weekly/monthly reporting	Automated recurring security reports, reducing manual reporting time and improving visibility
Провёл pentest и добился fixes	Реально закрыты exploitable paths, не просто написан PDF	Performed web/API testing and validated remediation for high-risk findings before production release

Используй простую формулу перевода:

- Tooling → coverage, consistency, earlier detection.
- Automation → reduced manual effort, faster feedback loop, less toil.
- Triage → signal/noise ratio, SLA, focus on exploitable risk.
- Fix validation → lower residual risk, safer release.
- Documentation/playbooks → repeatability, onboarding speed, lower bus factor.

Чего точно нужно не делать

Не придумывай fake impact. Ложь – не твой союзник. Если цифр нет — используй scope и qualitative outcome: “standardized workflow”, “improved visibility”, “reduced manual triage”, “enabled teams to prioritize fixes”. Это честнее и выглядит взрослее. Реальный опыт стоит денег, вымышленный экспириенс – ловушка с риском бан-листа.

4. Банк метрик для hands-on security

Ниже — набор метрик, которые можно использовать в CV, LinkedIn, interviews и performance review. Не нужно вставлять всё подряд. Выбери 5–8 метрик, которые реально отражают твою работу.

Область	Метрики, которые звучат сильно	Пример формулировки
AppSec	SAST/DAST/SCA coverage; false positive rate; findings by severity; time-to-triage; fix validation rate; ASVS coverage	Improved AppSec coverage for 25+ services by integrating SAST/SCA and standardizing triage workflow
DevSecOps	pipeline adoption; build failure policy; secrets detection; IaC scan coverage; lead time to feedback; policy-as-code checks	Added security gates to CI/CD without blocking normal delivery flow; reduced manual review work
Vuln Management	critical/high backlog; SLA compliance; KEV exposure; EPSS-driven prioritization; remediation aging; asset coverage	Prioritized remediation using severity, exploitability and business exposure instead of raw CVSS only
Pentest / Offensive	validated critical paths; exploitability; retest pass rate; business logic flaws; auth/session issues; API coverage	Found and validated high-risk web/API issues, then supported dev teams through remediation and retest
Cloud Security	CSPM findings; IAM excessive permissions; public exposure; encryption coverage; logging coverage; misconfig MTTR	Reduced cloud misconfiguration backlog by creating repeatable checks for IAM, storage exposure and logging
SOC / Detection	MTTD; MTTR; alert fidelity; detection coverage; false positive reduction; rule tuning; playbook completion	Tuned detections and playbooks to improve alert quality and analyst response speed

Цифры можно указывать диапазонами, если точные данные нельзя раскрывать:

- “20+ repositories”, “10+ product teams”, “100+ monthly findings”, “30%+ reduction”, “from weekly manual report to automated daily dashboard”.
- “Reduced triage from days to hours” — если точные часы нельзя назвать, но тренд честный.
- “Improved coverage from partial to organization-wide for selected product line” — если цифр нет, но scope понятен.
- “Prioritized internet-facing and KEV-listed vulnerabilities first” — сильнее, чем “worked with CVEs”.

Профессиональная рамка
Для vuln management не ограничивайся CVSS. Современная приоритизация всё чаще смотрит на exploitability, exposure, business criticality, KEV/EPSS и наличие compensating controls. В CV это показывает зрелость, даже если ты hands-on.

Где hands-on специалист становится дороже

Высокооплачиваемая hands-on работа чаще появляется не там, где “ищут человека на сканер”, а там, где security встроена в дорогой engineering process: продукт, cloud, fintech, SaaS, marketplace, enterprise, regulated industries, AI/platform teams. Там меньше романтики, больше ответственности, но и выше цена ошибки.

Где искать больше денег	Почему платят выше	Как себя упаковать
Product / SaaS	Security влияет на release, trust, compliance, customer requirements	Покажи AppSec, CI/CD, API security, developer-friendly remediation
Fintech / payments	Высокая цена fraud, data leakage, downtime и audit findings	Покажи secure SDLC, vuln SLA, logs, access control, threat modeling basics
Cloud / platform teams	Ошибки в IAM, networking и secrets быстро становятся инцидентом	Покажи cloud security, IaC scanning, CSPM, least privilege, automation
Consulting / pentest boutiques	Платят за скорость, качество reports и умение объяснить риск	Покажи methodology, PoC discipline, retest, report quality, client communication
Startups с funding	Нужно быстро закрыть gaps без огромной security team	Покажи autonomy, automation, pragmatic risk reduction, ability to build from scratch

Деньги любят конкретику
Если в вакансии написано “hands-on AppSec” — не продавай себя как general security enthusiast. Собери targeted CV под pain: pipeline security, SCA/SAST, API testing, vulnerability workflow, cloud exposure, developer support. Чем точнее попадание в боль, тем легче защищать higher compensation.

Role-specific упаковка: что подсвечивать

Роль	Что подсветить в CV	Какая история хорошо продаёт
Junior Security Engineer	Labs, scripts, Linux/Windows, networking, Python/Bash/PowerShell, basic cloud, writeups	Самостоятельно собрал lab, автоматизировал проверку, написал понятный report
Middle AppSec Engineer	SAST/SCA/DAST, Burp, API, OWASP ASVS/Top 10, triage, secure SDLC	Встроил проверки в pipeline и помог dev team закрывать findings без хаоса
Senior AppSec / Product Security	Threat modeling, API/business logic, secure design review, developer enablement	Нашёл systemic issue и превратил его в reusable checklist/control
DevSecOps Engineer	CI/CD, IaC, containers, secrets, policy-as-code, GitHub/GitLab/Jenkins	Сделал security gates, которые дают early feedback и не убивают delivery
Vulnerability Manager	Asset coverage, SLA, KEV/EPSS, prioritization, dashboards, remediation tracking	Перевёл raw scanner output в управляемый risk-based backlog
Pentester / Offensive	Web/API/mobile/internal testing, PoC, report writing, retest, safe exploitation	Доказал exploitable path, объяснил impact и довёл до validated fix
Cloud Security Engineer	IAM, network exposure, CSPM, logging, encryption, Kubernetes, Terraform	Автоматизировал checks для cloud misconfigurations и снизил recurring issues

Техническое интервью: как не утонуть в деталях

На technical interview проверяют не только “знает ли человек инструмент”. Проверяют thinking process: как ты собираешь facts, строишь гипотезы, проверяешь assumptions, выбираешь trade-offs и объясняешь результат. Даже если вопрос узкий, отвечай структурно.

Тип вопроса	Как отвечать	Пример фразы
Design / architecture	Сначала assumptions, потом controls, потом trade-offs	“I would first clarify data flow, trust boundaries and deployment model, then choose controls.”
Troubleshooting	Опиши steps: reproduce, logs, scope, isolate, fix, validate	“I would check whether it is a scanner issue, pipeline config issue or actual dependency exposure.”
Tooling	Не просто “использовал”, а как настраивал и что улучшил	“The hard part was not enabling the scanner, but tuning severity and ownership workflow.”
Security finding	Evidence, exploitability, impact, remediation, retest	“I would provide PoC, affected endpoint, risk, fix recommendation and validation criteria.”
Unknown topic	Честность + подход к изучению	“I have not run this exact setup, but I know the underlying model and would test it this way.”

Что бесит интервьюеров

Когда кандидат говорит уверенно, но без доказательств. Лучше честное “я не делал именно этот вариант, но понимаю принципы” + хороший reasoning, чем fake expertise. В security доверие важнее театра.

ATS и keywords без самообмана

Для hands-on ролей keywords важны: ATS и recruiter сначала матчят vocabulary. Но keyword stuffing быстро ломается на technical screen. Правильная стратегия: взять vocabulary вакансии и связать его с реальными примерами. Смотри табличку ниже.

Keyword из вакансии	Плохая вставка	Нормальная вставка
SAST / SCA	SAST, SCA, DAST, DevSecOps	Integrated SAST/SCA checks into GitLab CI and tuned false positives for product teams
Kubernetes security	Kubernetes security experience	Reviewed Kubernetes workload configs for secrets, RBAC, image policy and network exposure
Threat modeling	Knowledge of threat modeling	Participated in lightweight threat modeling for API flows using data flow and trust boundary review
Incident response	IR, SOC, SIEM	Investigated alerts, collected evidence, updated detection logic and documented response steps
Cloud security	AWS/Azure/GCP security	Reviewed IAM permissions, public exposure, logging and encryption settings in cloud environments

5. CV checklist для hands-on ролей

Хорошее technical CV для middle/senior hands-on роли должно быть скучным в хорошем смысле: понятным, плотным, конкретным и без театра. Не надо писать “visionary cybersecurity professional”. Для executor лучше звучит: “AppSec / DevSecOps engineer with hands-on experience in CI/CD security, vulnerability triage, SAST/SCA/DAST, cloud and automation.”

Раздел	Что должно быть	Проверка качества
Headline	Роль + специализация + сильный stack	Понятно за 5 секунд, на какую работу ты подаёшься
Summary	3–5 строк: domain, scope, tools, outcomes	Без воды, без “passionate”, без общих слов
Skills	Сгруппировано: AppSec, CI/CD, Cloud, Tools, Languages, Standards	Не свалка keywords, а карта твоей специализации
Experience bullets	Action + Scope + Tool + Evidence + Impact	Каждый bullet можно проверить вопросом “so what?”
Projects	Automation/scripts/tools/labs, если коммерческого опыта мало	Есть доказательство, что ты умеешь руками
Certs/Education	Только релевантное	Не занимает больше места, чем реальный опыт

Чеклист перед отправкой CV:

- □ В первых 10 строках видно, какую роль ты ищешь: AppSec, DevSecOps, Pentest, Cloud Security, SOC, Vulnerability Management.
- □ Есть technical keywords из вакансии, но они встроены естественно, не как keyword stuffing.
- □ Каждая позиция содержит 3–6 bullets с результатами, а не список обязанностей.
- □ Есть numbers или scope: repos, apps, findings, endpoints, users, pipelines, incidents, cloud accounts.
- □ Не написано “participated in” там, где ты реально делал руками.
- □ Нет длинных абзацев. Recruiter должен быстро сканировать глазами.
- □ Есть 2–3 bullets про automation: scripts, dashboards, pipelines, rules, reports.
- □ LinkedIn и CV не противоречат друг другу по датам, ролям и seniority.

Главный фильтр
Если bullet можно применить к любому человеку из отдела — он слабый. “Worked on vulnerability management” слабый. “Triaged 100+ monthly findings across 40+ assets and helped reduce aging critical backlog” уже похоже на реальную работу.

6. “Было → стало”: замена слабых формулировок

Слабо	Сильнее для hands-on CV
Занимался анализом уязвимостей	Triaged and prioritized vulnerability findings across internal services using severity, exploitability and asset exposure
Работал с Burp Suite	Performed manual web/API security testing with Burp Suite, validating auth, session management and business logic issues
Настраивал SAST	Integrated SAST into CI/CD pipelines and tuned rules to improve signal quality for development teams
Писал отчёты по безопасности	Prepared actionable security reports with severity, evidence, remediation steps and retest status
Участвовал в DevSecOps	Automated security checks in CI/CD and helped shift vulnerability detection earlier in the SDLC
Мониторил алерты	Investigated and triaged security alerts, reducing noise through rule tuning and playbook updates
Проверял облако	Reviewed cloud misconfigurations around IAM, public exposure, encryption and logging coverage
Помогал разработчикам исправлять баги	Worked with developers to validate fixes and reduce recurring security defects

Русская версия формулы для самопроверки:

> **Формула bullet для подачи своего профиля**
> “Сделал X для Y, используя Z, в масштабе N, чтобы улучшить / снизить / ускорить / стандартизировать R.” Если нет цифры N — добавь scope словами: для продуктовой команды, для линейки сервисов, для CI/CD пайплайна, для external-facing assets. Запомни – конкретика, пусть меньше, да четче, пусть уже за то понятнее для бизнеса

Шаблон	Пример
Implemented / automated / integrated [security control] for [scope] using [tools], resulting in [outcome].	Integrated SCA checks for 30+ repositories using GitLab CI and Dependency-Check, improving visibility into vulnerable dependencies before release.
Reduced / improved / standardized [metric/process] by [action].	Reduced manual vulnerability reporting by automating weekly exports and ownership mapping for product teams.
Performed / validated / retested [technical assessment] and helped remediate [risk].	Performed API security testing and retested fixes for auth/session issues before public launch.

7. STAR для технаря: technical stories без менеджерской позы

Для hands-on кандидата STAR должен быть техническим, но не утопленным в деталях. Интервьюеру важно услышать: проблема была реальная, ты понял контекст, выбрал подход, сделал руками, проверил результат и можешь объяснить trade-offs.

Блок	Что говорить	Вопросы к себе
S — Situation	Контекст: система, команда, риск, ограничение	Где это было? Что ломало процесс? Почему это было важно?
T — Task	Твоя конкретная задача и зона ответственности	Что именно было на тебе? Что считалось успехом?
A — Action	Технические действия: tools, scripts, configs, tests, validation	Что ты сделал руками? Какие trade-offs были?
R — Result	Измеримый или честно описанный результат	Что изменилось? Как проверили? Что стало проще/быстрее/безопаснее?

Пример STAR история для DevSecOps / SCA

> **STAR: Пример ответа**
> Situation: зависимости проверялись нерегулярно, findings приходили поздно и без ownership. Task: встроить SCA в CI/CD для группы сервисов и сделать процесс triage пригодным для разработчиков. Action: подключил scanner, настроил policy thresholds, suppression workflow, ownership mapping и weekly export. Result: команда начала видеть vulnerable dependencies до релиза, manual reporting сократился, а backlog high/critical стал управляемым через SLA.

Пример STAR история для Pentest / API

> **STAR: Пример ответа**
> Situation: перед релизом API не было уверенности в auth/session controls. Task: провести targeted testing и дать actionable findings. Action: проверил authorization bypass, token handling, IDOR-like paths, rate limits и error handling; оформил PoC без разрушительных действий; после fixes сделал retest. Result: критичные paths закрыли до релиза, команда получила checklist для похожих endpoints.

Технические детали стоит раскрывать по принципу “сначала summary, потом depth on demand”. Не начинай с 5 минут про флаги Burp или YAML pipeline, пока тебя не попросили углубиться.

8. Self-intro и interview answers

Self-intro на 45 секунд должен дать интервьюеру карту: кто ты, в чём специализация, какой stack, какой scope, какой тип задач ты хочешь решать дальше. Без “я ответственный, коммуникабельный, стрессоустойчивый”.

Шаблон self-intro

“Я [роль/уровень], последние [N] лет работаю с [domain]. Основной фокус — [2–3 направления]. На практике делал [scope/tools]: [examples]. Сильнее всего я полезен там, где нужно [technical value]: встроить security в delivery, разобрать backlog, автоматизировать проверки, снизить noise, довести findings до fix. Сейчас ищу роль, где смогу глубже работать с [target stack/domain].”

Твоя задача – максимум плотности смысла на минимум длины сообщения\слов. Это по-деловому. Меньше говорить – больше показывать. Деньги тебе в итоге платят на за часы с «9 до 18ч» и не за X писем и N митапов в неделю. А за твой вклад в проект, это про то как это бустит прибыль тех людей на которых ты работаешь.

Пример для AppSec / DevSecOps

Pitch: Готовый вариант

“Я AppSec / DevSecOps engineer с hands-on опытом в SAST/SCA/DAST, CI/CD security и vulnerability triage. Работал с product teams, подключал scanners в pipeline, настраивал rules и помогал разработчикам доводить findings до fixes. Мне интересны роли, где нужно не просто запускать tooling, а сделать security checks usable для engineering: меньше шума, быстрее feedback, понятный ownership и нормальные отчёты.”

Вопрос	Слабый ответ	Сильный hands-on ответ
Why are you looking?	Хочу больше зарплату	Хочу роль с более зрелым engineering environment, где мой AppSec/DevSecOps опыт даст больше value и позволит работать с большим scope.
What is your strength?	Я быстро учусь	Я быстро превращаю неструктурированный security backlog в понятный workflow: triage, ownership, SLA, reports, retest.
Tell me about conflict	Разработчики не хотели чинить	Был friction из-за scanner noise; я разобрал false positives, настроил severity policy и сделал findings более actionable для dev team.
Why this role?	Интересная компания	Мне подходит stack: CI/CD, cloud, AppSec tooling. Я уже делал похожие задачи и могу быстро включиться в pipeline/security automation.

9. Типичные ошибки русскоязычных кандидатов

Ошибка	Как это выглядит	Как исправить
Слишком скромно	“Ну я просто помогал с уязвимостями”	Назови ownership: что именно делал, в каком scope, какой результат
Слишком технично без контекста	10 минут про tool flags, но непонятно зачем	Начни с риска/задачи, потом дай детали по запросу
“Мы” вместо “я”	Непонятно, твой вклад или команды	Разделяй: “team achieved”, “I owned / implemented / automated”
Боязнь цифр	Нет масштаба, всё звучит маленьким	Используй диапазоны и scope: repos, apps, endpoints, findings, incidents
Резкий негатив о прошлой работе	“Там всё было плохо, менеджеры тупые”	Говори взросло: constraints, trade-offs, lessons learned
Ожидание, что “по скиллам и так видно”	CV как список технологий	Покажи evidence: projects, artifacts, metrics, STAR stories
Синдром “меня должны заметить”	Пассивная позиция в поиске	Пиши targeted CV под роль, готовь pitch, тренируй ответы

Заметка от практика: Психологический shift
На интервью ты не оправдываешься и не просишь шанс. Ты продаёшь рабочую гипотезу: “я уже решал похожие задачи, понимаю ограничения, могу включиться и дать результат”. Это спокойная взрослая позиция, не понты.

Для junior это особенно важно: если коммерческого опыта мало, продаётся не seniority, а trajectory: labs, GitHub, writeups, CTF, home projects, automation scripts, понятный learning plan и честность по границам компетенции. Не ссы если ты джун, ссы если ты 0. Любой спец, тимлид, директор кода то был стажером или джуном. Это нормальный путь. Задача не в в том что бы махнуть с джуна в лиды за один хоп, задача пройти это путь быстрее, эффективнее и с большим количеством $ на своем счету. Не, правда ли, мой друг?

10. Психологическая подготовка и salary talk

Перед важным интервью техническому кандидату часто мешают две крайности: зажатость “я никто” и агрессивная защита “я всё знаю”. Нужна третья позиция: спокойный инженер, который уверен в своём опыте, но честно говорит о неизвестном.

Ситуация	Что делать
За 24 часа до интервью	Повтори 5 STAR stories, 10 ключевых bullets CV, target stack вакансии и 3 вопроса к команде. Не учи всё подряд.
За 30 минут	Открой notes: self-intro, salary range, вопросы, 3 достижения. Сделай дыхание 4-6: вдох 4, выдох 6, 2–3 минуты.
Если не знаешь ответ	Скажи: “I have not used this exact setup in production, but I would approach it this way...” и покажи ход мышления.
Если давят глубиной	Не фантазируй. Уточни assumptions, проговори trade-offs, покажи troubleshooting logic.
Если спрашивают salary	Не называй минимальную цифру. Говори диапазон, основанный на role scope, market, location, remote/onsite, benefits.

Фраза: Salary phrase
“Based on the role scope and my hands-on experience with AppSec/DevSecOps automation, I am targeting a range around [X–Y]. I am flexible depending on total compensation, remote setup, responsibilities and growth path.”

Не продавай нужду. Продавай способность закрыть задачу. Даже если тебе очень нужна работа, в коммуникации должно звучать не “возьмите меня”, а “я понимаю вашу боль и могу быть полезен вот так”. Не унижайся, не выпрашивай, не ставь себя в полицию «мальчик для битья». Иногда, могут быть кризисные ситуации и приходится затыкать себе рот, стискивать зубы что бы выжить. Но жить стоит по другому. Не будь в долгу. Не ищи помощи опираясь на благородство, сам уже сегодня предлагай то что можешь, и найдется тот кто обязательно поможет тебе за помощь оказанную ему. Деньги – лишь инструмент.

Вопросы, которые стоит задать hiring manager / team lead:

- Как сейчас устроен vulnerability triage и кто владеет remediation?
- Какие security checks уже встроены в CI/CD, а где ещё gaps?
- Что будет считаться успешным результатом через первые 90 дней?
- Сколько product teams / repos / services входит в scope роли?
- Как команда измеряет false positives, SLA и developer adoption?

11. 30-дневный план перепаковки профиля

Не надо ждать идеального момента. За месяц можно заметно поднять качество подачи.

Дни	Что сделать	Выходной артефакт
1–3	Собери inventory: роли, проекты, tools, scope, достижения, scripts, reports, dashboards	Raw experience map
4–7	Выбери 6–8 сильных technical stories и распиши по STAR	STAR bank
8–12	Перепиши CV bullets через Action + Scope + Evidence + Impact	CV v1
13–16	Собери skills section под 2–3 target roles	Targeted CV variants
17–20	Обнови LinkedIn: headline, about, experience, featured projects	LinkedIn profile
21–24	Сделай маленький portfolio: GitHub snippets, sanitized report, writeup, checklist	Evidence pack
25–27	Отрепетируй self-intro и 5 answers вслух	Interview script
28–30	Подайся на 15–25 релевантных вакансий и веди tracker	Application tracker

Минимальный evidence pack для hands-on кандидата:

- □ 1 sanitized security report или writeup без чувствительных данных;
- □ 1 automation script / pipeline snippet / detection rule / checklist;
- □ 1 diagram или short note: как ты строишь triage/remediation workflow;
- □ 1 список tools с контекстом: не просто “Burp”, а “Burp for web/API testing, auth/session checks, PoC validation”.

Про личный GitHub

Не надо выкладывать рабочие секреты, внутренние PoC и чужой код. Можно делать форки под свои задачи, но не копипасить. Можно показать безопасные шаблоны: parser для scanner reports, пример CI policy, checklist, demo dashboard, notes по ASVS, lab writeup. Для hands-on роли это часто сильнее, чем очередная фраза “good team player”. Или попытка выдать себя за black hat. Работай честно, делись искренне, помогай от сердца.

12. Финальный чеклист: «продай» себя дороже, оставаясь технарем

Перед отправкой CV или перед интервью пройди этот короткий чеклист:

- □ Я могу объяснить свою специализацию за 45 секунд без воды.
- □ В моём CV видно не только tools, но и scope: apps, repos, findings, pipelines, assets.
- □ У меня есть минимум 5 STAR stories по реальным technical tasks.
- □ Я умею говорить “я сделал” там, где это мой вклад, и “команда сделала” там, где это командный результат.
- □ Я не преувеличиваю опыт, но и не обесцениваю себя.
- □ Я показываю automation, triage, validation, documentation и collaboration with dev teams.
- □ Я могу объяснить, чем моя работа снижала risk, ускоряла feedback loop или улучшала coverage.
- □ У меня есть questions к работодателю, которые показывают зрелость hands-on engineer.
- □ Я знаю свой salary range и не называю самую низкую цифру из страха.

Финальная мысль
Можно оставаться человеком, который работает руками. Можно быть executor. Но не обязательно быть дешёвым executor. Упакуй свой technical output так, чтобы рынок видел не “ещё одного спеца по тулзам”, а инженера, который умеет закрывать реальные security gaps.

Важная информация