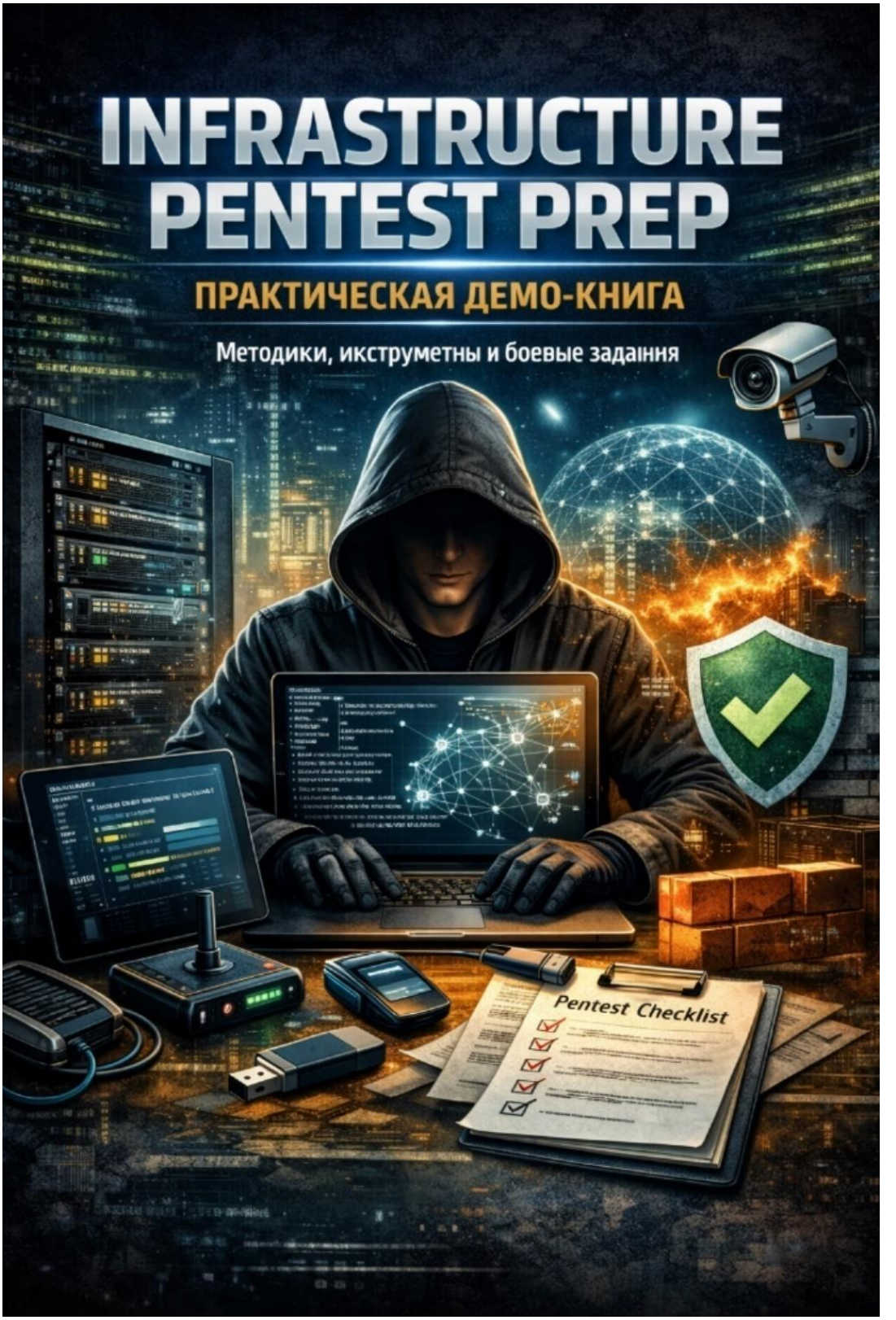

INFRASTRUCTURE
PENTEST PREP
ПРАКТИЧЕСКАЯ ДЕМО-КНИГА
Методики, икструметны и боевые задаіння
Pentest Checklist

# ПРАКТИКА ПЕНТЕСТА

Interview Preparation Book - демо-сборка для подготовки к интервью
на позицию Penetration Tester (Middle / Middle+ / Senior)
с акцентом на инфраструктурный pentest, AD, Linux, сеть и practical thinking


**white2hack | I.P.**
**март 2026 | версия 1.1 (демо, starter edition)**

---

*Учебно-практическая брошюра по методологии, интервью-вопросам, заданиям, репортингу, лабораториям и must-have инструментарию пентестера.*

# ДИСКЛЕЙМЕР И ПРАВИЛА ИСПОЛЬЗОВАНИЯ

Материал собран из **десятков открытых источников, переработан, адаптирован** и собран в единую рабочую брошюру. Здесь есть заимствования идей и формулировок из книг, обучающих курсов, GitHub-репозиториев, официальных документов, методологий и публичных write-up.

**Автор не несет ответственности** за любые последствия использования этого материала. Вся информация дана только в ознакомительных и учебных целях. Это не индивидуальный разбор и не персональная консультация.

Для полноценной консультации под свой техстек, уровень компетенций, рынок РФ/МИР и конкретную вакансию **обращайтесь к автору: Ivan Piskunov | white2hack.**



**В тексте могут встречаться мелкие неточности:** релиз собирался вручную, финальная вычитка и сборка выполнялись автором лично. Автор не позиционирует себя как профессионального pentest-only специалиста, но более 15 лет находится в кибербезопасности, имеет широкий технический бэкграунд и передает практический опыт в рамках своей компетенции.

**Как читать эту книгу:**

Подача рассчитана на новичков и уверенных джунов, но вопросы, задания и комментарии местами тянут на middle+ и senior.

# ОГЛАВЛЕНИЕ



# Что внутри и зачем это читать

- **Практический фокус, не сухая теория.** Книга собрана вокруг того, что действительно спрашивают на интервью и что реально делает пентестер в работе: разведка, enumeration, privilege escalation, lateral movement, фиксация findings и логика атаки.

- **Актуальный стек знаний на 2025/2026 год.** Материал опирается на современные подходы, best practices и живые источники: PTES, NIST SP 800-115, OWASP WSTG, OWASP Top 10:2025, HackTricks, PortSwigger, HTB, THM и другие практические базы.

- **Много hands-on контента.** Внутри - практические задания, мини-кейсы, фрагменты команд, разборы типовых артефактов, примеры мышления на интервью и задачи, где нужно не угадывать, а реально соображать как инженер.

- **Упор на инфраструктурный пентест.** Отдельное внимание уделено внутренней сети, Windows/AD, Linux, сетевому периметру, типовым векторам компрометации, ошибкам конфигурации и тому, как из “маленькой зацепки” вырастает полноценная цепочка атаки.

- **Подготовка именно к собеседованию, а не только к лаборатории.** Здесь собраны вопросы, на которых часто сыпятся кандидаты: что говорить, как мыслить вслух, как не выглядеть “человеком, который просто запускал тулзы”, и как показать зрелый инженерный подход.

- **Разбор реальной пользы для бизнеса.** Книга помогает понять не только *как получить доступ*, но и *что это значит для компании*: утечка данных, захват домена, движение по сети, шифрование инфраструктуры, остановка сервисов, удар по деньгам и репутации.

- **Отдельный блок по отчетности и подаче findings.** Не только “нашел уязвимость”, но и как грамотно ее описать, подтвердить evidence, оценить риск, дать внятную рекомендацию и написать отчет так, чтобы его могли читать и инженеры, и менеджмент.

- **Удобный формат для самостоятельной подготовки.** Это не рекламная брошюра и не набор случайных заметок. Это прикладная рабочая сборка: чек-листы, опорные схемы, квизы, глоссарий, подборка тулов, лабораторий и материалов, с которыми можно реально расти до более сильного оффера.

# ЧТО ХОЧЕТ РЫНОК: КРАТКИЙ СРЕЗ ВАКАНСИЙ PENETRATION TESTER В 2025-2026

Эта глава — это статистика по всем job boards, честный публичный срез рынка по свежим вакансиям **Indeed, Monster, ZipRecruiter, Getmatch и Habr Career.** Этого достаточно, чтобы увидеть повторяющиеся требования работодателей и понять, какие навыки реально конвертируются в офферы.

## Мини-snapshot по площадкам (Январь\Март 2026)

**Indeed:** по публичной выдаче видно около 953 вакансий по запросу Penetration Tester; отдельно junior-ролей заметно меньше — порядка 52, а senior — порядка 572. Для кандидата это важный сигнал: рынок в целом живой, но входных ролей ощутимо меньше, чем middle/senior.

**Monster и ZipRecruiter** подтверждают ту же картину: ролей много, но они чаще описываются как зрелые cross-domain позиции — web, API, internal/external network, cloud, wireless, thick client, reporting и client-facing communication.

**Habr Career** по публичной выдаче сейчас показывает существенно более узкий рынок: около 5 открытых вакансий по специализации pentester и около 3 вакансий с навыком «Тестирование на проникновение». Это уже не массовая выборка, а скорее сигнал про более точечный и жёсткий рынок.

**Getmatch** даёт полезный качественный срез по РФ-рынку: видно middle, senior и lead-роли, а по отдельным вакансиям просматриваются вилки уровня 150–300 тыс. ₽ и выше; встречаются и cloud/red-team позиции с более высоким уровнем компенсации.

## Что чаще всего требуют на US / EU рынке

**На англоязычном рынке стабильно повторяется одна и та же ось:** web/API testing, internal/external infrastructure pentest, Windows/Linux fundamentals, scripting/automation и сильный

reporting. Для middle/senior поверх этого часто ждут cloud, mobile/thick client, wireless и элементы red team.

Работодатели там смотрят не только на набор техник, но и на то, умеешь ли ты вести **engagement целиком:** планировать подход, формулировать гипотезы, безопасно подтверждать impact, писать remediation-focused findings и говорить с заказчиком понятным языком.

**Практический вывод:** even если ты целишься в junior-позицию, звучать выгоднее как человек, который уже умеет мыслить через scope, evidence, impact и коммуникацию, а не только через список тулов.

## Что чаще всего видно на РФ-рынке

На российском рынке роли чаще выглядят как гибрид infra/web/red-team. В описаниях регулярно всплывают Windows, Linux, AD, Burp Suite, Python, PowerShell, OSINT, внутренняя инфраструктура, сегментация, Wi-Fi, PoC-подтверждение attack path и обязательное написание отчётных документов.

Заметно сильнее ощущается **запрос на хард-скиллы и инженерную насмотренность:** понимание сетевых технологий, сервисных учёток, типовых уязвимостей ОС и ПО, а также умение быстро разложить инфраструктурную задачу по шагам.

**Отдельный паттерн РФ-рынка:** часто ценится не узкий «чисто web» профиль, а широкий offensive security mindset с готовностью работать и по внутреннему периметру, и по внешним сервисам, и по нестандартным инфраструктурным кейсам.

## Как искать работу и как подаваться: US / EU vs РФ

Если целишься в **US/EU** рынок, **готовь не только технику, но и подачу.** Там важны STAR-модель для поведенческих вопросов, culture fit, ясная коммуникация, умение объяснить trade-offs и показать, что ты понимаешь миссию продукта, а не просто хочешь «ломать всё подряд».

Для US/EU полезно заранее **подготовить 4–6 историй:** как ты разбирал неоднозначный finding, как спорил про приоритет

remediation, как действовал в scope-ограничениях, как объяснял риск нетехническому собеседнику и как подходил к незнакомой системе. Это почти так же важно, как знание тулов.

Если целишься в **РФ-рынок, чаще выигрывает сильный технический разговор:** сеть, Windows/Linux, AD, writing/reporting, инструменты, иногда нормативка/регуляторика и умение быстро разложить инфраструктурную задачу на приоритетные шаги. Софт-**скиллы важны, но ими редко закрывают слабую базу.**

**Для обоих рынков правило одно:** не пытайся выглядеть «магом». Сильнее звучит кандидат, который честно говорит о границах, объясняет логику проверки и показывает, как превращает наблюдение в доказуемый finding.

### Практический вывод для читателя

Если времени мало, **качай четыре столпа:** web/API triage, внутренняя инфраструктура (AD/Linux/сеть), базовая автоматизация и writing/reporting. Именно эта связка чаще всего повторяется в вакансиях и лучше всего конвертируется в интервью.

Хороший market-fit для pentester-а сегодня — это не «я знаю 500 CVE». Это сочетание технической базы, дисциплины, понятной коммуникации и умения не теряться в незнакомой среде.

## МЕТОДОЛОГИЯ\ФРЕЙМВОРКИ И КАРТА ПОДГОТОВКИ

Хороший pentest на интервью и в жизни - это не про магию а-ля «что-то там в консоли» и не про один любимый tool. Это про повторяемый подход: **scope, recon, enumeration, hypothesis-driven testing, exploitation, impact, reporting и нормальную коммуникацию с заказчиком.**

В 2025/2026 базовый каркас удобно держать в голове через **PTES, NIST SP 800-115 и OWASP WSTG**. Это не религия, а карта местности: помогает не забыть pre-engagement, артефакты, validation findings и финальный репорт.

Для web-направления логично мыслить через OWASP WSTG и OWASP Top 10:2025, но не ограничиваться ими. На интервью сильнее смотрится кандидат, который умеет связать vuln не только с категорией, но и с реалистичным business impact.

**Ось подготовки:** для интервью удобно тренировать 5 мышц: recon/enumeration, web logic, Windows/AD basics, Linux priv-esc mindset и reporting.

## На что опираться в 2026 году

**PTES** - как общий скелет этапов и deliverables.

**NIST SP 800-115** - как внятный guide по планированию, выполнению и анализу результатов технических тестов.

**OWASP WSTG** - как практический web testing framework.

**OWASP Top 10:2025** - как актуальный awareness baseline для web risk discussion.

**HackTricks, PortSwigger Web Security Academy, HTB Academy, TryHackMe** - как ежедневные hands-on источники.

# MUST-HAVE ТУЛКИТ ПЕНТЕСТЕРА

Ниже не список "must install everything". Это ориентир на то, что полезно хотя бы знать, запускать и понимать на уровне задач, сильных сторон и ограничений. **Это базовый минимум!**

| Тул | Зачем знать/что дает |
|---|---|
| **Nmap** | база для discovery, service enumeration, NSE |
| **Rustscan** | быстрый фронт для TCP recon |
| **Masscan** | массовое сканирование больших диапазонов |
| **NetExec / CrackMapExec** | enum и post-auth проверки в Windows/AD |
| **Impacket** | AD/SMB/Kerberos toolkit |
| **BloodHound** | графовая аналитика прав и путей в AD |
| **Responder** | LLMNR/NBT-NS poisoning в лабах |
| **Burp Suite** | основной комбайн для web testing |
| **ffuf** | fuzzing URL, vhost, parameters, files |
| **gobuster** | dir/vhost enumeration |
| **dirsearch** | альтернатива для content discovery |
| **nikto** | быстрый sanity-check на web misconfig |
| **sqlmap** | проверка SQLi при наличии оснований |
| **httpx** | быстрый probing web endpoints |
| **Amass** | attack surface и subdomain enum |
| **theHarvester** | OSINT и поверхностный сбор артефактов |
| **feroxbuster** | параллельный контент-дискавери |
| **wfuzz** | гибкий web fuzzing |
| **wpscan** | специализированный WordPress тестинг |

| | |
|---|---|
| **hashcat** | password auditing в рамках разрешенного scope |
| **John the Ripper** | альтернатива под cracking-аудит |
| **PEASS-ng** | linPEAS/winPEAS для локальной оценки misconfig в лабах |
| **enum4linux-ng** | SMB enum |
| **mimikatz** | знать концептуально и понимать риски; использовать только в разрешенных стендах |
| **Wireshark** | трафик, протоколы, ручной разбор |
| **tcpdump** | быстрый CLI sniffing |
| **Metasploit** | знать, но не превращаться в 'msf-only operator' |
| **searchsploit** | локальный поиск публичных exploit references |
| **Ghidra** | азбука reverse/triage для PE/ELF |
| **CyberChef** | decode/encode/transform артефактов |
| **CrackMapExec / NetExec** | быстрый AD/SMB triage, auth spray, share enum, execution paths и первичная проверка relay/creds-сценариев |
| **BloodHound / SharpHound** | граф доверия и attack path в AD; полезен как карта гипотез, но не как замена ручной верификации |
| **Hashcat** | офлайн-проверка стойкости паролей и демонстрация реального impact слабых creds |
| **nuclei** | массовая валидация типовых misconfig/known issues; findings всё равно нужно подтверждать руками |
| **Wireshark / tshark** | разбор DNS/SMB/Kerberos/LDAP-трафика и доказательство attack path без догадок |
| **Ligolo-ng / Chisel** | pivoting и транспорт в сегментированной сети, когда нужен быстрый и понятный тоннель |
| **Gobuster** | directory/vhost/DNS brute-force; простой и надёжный базовый инструмент |
| **ScoutSuite / Prowler** | cloud posture triage для AWS/Azure/GCP, если интервьюер хочет увидеть, что ты не теряешься в облаке |
| **WhatWeb** | быстрый fingerprinting веб-стека: CMS, фреймворки, веб-сервер, версии, встроенные технологии; полезен на этапе первичного web recon. |
| **testssl.sh** | проверка TLS/SSL: протоколы, cipher suites, криптографические flaws и слабые конфигурации; очень полезен для веба, reverse proxy, VPN-порталов и внешних сервисов. |

| | |
|---|---|
| **EyeWitness** | быстрый triage большого списка хостов/URL: скриншоты веб-панелей, server headers, визуальная приоритизация интересных целей. |
| **Katana** | современный web crawler для сбора endpoint'ов, route'ов и surface area, включая headless crawling для SPA/JS-приложений. |
| **Arjun** | поиск скрытых HTTP-параметров; полезен там, где обычный recon не показывает все input points приложения. |
| **Dalfox** | специализированный automation-scanner под XSS и анализ параметров; хорош как узкопрофильный web-инструмент, а не как "универсальный сканер всего". |
| **OpenVAS / Greenbone** | полнофункциональный vulnerability scanner для authenticated и unauthenticated проверок; полезен для базовой широкой оценки хостов, сервисов и misconfig на сети. |
| **onesixtyone** | быстрый SNMP-scanner для поиска устройств, отвечающих по SNMP, и первичной оценки сетевого оборудования/инфраструктурных узлов. |
| **ike-scan** | discovery и fingerprinting IKE/IPsec VPN-узлов; особенно уместен, когда в scope есть VPN-шлюзы и perimeter-оборудование. |
| **SMBMap** | удобный SMB enumeration tool: shares, permissions, содержимое, доступы; полезен при внутреннем pentest и оценке Windows/SMB-поверхности. |
| **WhatWeb** | быстрый fingerprinting веб-стека: CMS, фреймворки, веб-сервер, версии, встроенные технологии; полезен на этапе первичного web recon. |
| **testssl.sh** | проверка TLS/SSL: протоколы, cipher suites, криптографические flaws и слабые конфигурации; очень полезен для веба, reverse proxy, VPN-порталов и внешних сервисов. |
| | |

**Практический совет:**

На интервью сильнее звучит не длинный список софта, а 5-7 инструментов, которыми ты реально умеешь пользоваться руками + твой mindset

# !!!!ВНИМАНИЕ!!!

Это **демо-версия** полноценного учебного издания по подготовке к интервью и практике в penetration testing.

**В полной версии:**

- **170+ страниц** прикладного материала;
- еще **40 теоретических вопросов**;
- еще **12 практических real-word кейсов**;
- **детальный roadmap по самостоятельному обучению** с разбивкой по неделям и темам;
- **личные рекомендации автора** по развитию навыков;
- дополнительные **sample reports**, приложения и чек-листы;
- расширенные рекомендации по проведению pentest-а и подготовке к собеседованиям (репо в GitHub).

Эта версия нужна, чтобы показать общий стиль, глубину и практическую ценность данного материала. Free version ©

За **полной версией** и персональной консультацией - к автору: **Ivan Piskunov | white2hack**.

# ПРАКТИЧЕСКИЙ АЛГОРИТМ РАБОТЫ

Когда тебя спрашивают ***"как бы ты подошел к тесту?",*** не надо уходить в хаотичный список тулов. Дай последовательный, взрослый ответ, а именно:

- **Уточни scope:** что именно можно трогать, когда, с каких IP, с какими ограничениями. На интервью это маркер зрелости.
- **Собери первичный attack surface:** DNS, web, exposed services, auth points, third-party интеграции, staging/dev, forgotten subdomains.
- **Проведи enumeration глубже**, чем просто banner grabbing. Ищи trust boundaries, auth flows, misconfig, default paths, старые панели, SSO, file upload, secrets exposure.
- **Строй гипотезы:** 'если тут legacy SSO + слабый reset flow, то где может быть account takeover?', 'если SMB открыт, что из этого вытекает для AD pathing?'
- **Подтверждай finding двумя вещами:** техническим evidence и объяснением impact для бизнеса. Без impact report слабее и на интервью, и в жизни.
- **После находки не залипай в одну дыру.** Хороший pentester умеет решать, когда копать глубже, а когда двигаться дальше по coverage.
- **Фиксируй все артефакты сразу:** URL, запросы/ответы, скриншоты, команды, timestamps, test account, воспроизводимость.

## Мини-чек-лист перед интервью

### Как звучать сильнее на интервью

Сильнее звучит кандидат, который строит цепочку мысли: «я бы начал с... потому что... если подтвердится, это даст... дальше я бы проверил...». Интервьюеру нужен не список модных слов, а управляемое мышление.

Хороший тон - честно признавать границы. Фраза «точный флаг команды могу не вспомнить, но логика проверки такая-то» почти всегда лучше, чем самоуверенная чушь.

Красный флаг - романтизация разрушительных действий: «обязательно добить до DA/root», «если вижу upload - сразу shell», «всегда гоняю автоскрипты». Зрелее звучит подход про evidence, impact и безопасную валидацию.

Повтори базовые nmap flags, HTTP triage, common auth flaws, Linux/Windows basics.

Освежи в голове 2-3 реальных или учебных кейса, которые можешь рассказать без воды.

Подготовь короткое объяснение разницы scanner vs pentest, finding vs risk, vuln vs impact.

Умей честно сказать "не знаю", но продолжить рассуждение через гипотезы и безопасную валидацию.

## Roadmap на 6-8 недель: junior -> уверенный junior / junior+

**Недели 1 - 2:** база сети, HTTP, Linux/Windows fundamentals, ручной recon, уверенная работа с Nmap/Burp/httpx/ffuf. Цель - перестать теряться в базовых протоколах и сервисах.

**Недели 3 - 4:** Linux privilege escalation, web auth/authorization, file upload, deserialization/misconfig, writing short findings. Цель - научиться объяснять reasoning chain, а не просто вспоминать термины.

**Недели 5 - 6:** AD/SMB/LDAP/Kerberos triage, basic BloodHound/Impacket, relay surface, pivoting basics. Цель - понимать внутренний pentest не только через «скан и эксплойт».

**Недели 7 - 8:** мини-проекты - 2-3 write-up, 1-2 mock interviews, 5-10 коротких findings в едином стиле, разбор вакансий и целенаправленное закрытие пробелов под рынок.


REAL-WORLD КЕЙСЫ
&
СИСТЕМНОЕ МЫШЛЕНИЕ
ПРИКЛАДНЫЕ НАВЫКИ ДЛЯ PENTEST-А.
ATTACK PATHS
VULNERABILITIES
OBJECTIVES
DATA BREACH
CONFIDENTIAL
ACCESS GRANTED
ANALYSIS REPORT
ЭКСПЛОИТАЦИЯ ОБЪЕКТОВ • АНАЛИЗ И РИСКИ • КРИТИЧЕСКОЕ МЫШЛЕНИЕ

# REAL-WORLD КЕЙСЫ И СИСТЕМНОЕ МЫШЛЕНИЕ

Ниже - не "боевые секреты", а упрощенные собирательные сценарии, похожие на то, что обсуждают на интервью и в отчетах.

## Финтех / bank-like - технический профиль отрасли

У банков и финтеха поверхность атаки редко ограничивается только публичным сайтом. Обычно есть web/mobile front, API gateway, партнерские интеграции, внутренние сервисы антифрода, identity stack, VPN/jump host-инфраструктура, а также отдельные зоны с усиленными регуляторными ограничениями. Часто **часть сервисов написана кастомно,** а часть держится на **коробочных решениях: WAF, IAM, MDM, VPN, secret storage, платежные шлюзы и SIEM**. Для пентестера здесь сильный старт - не «а где CVE», а карта trust zones: пользовательский контур, партнерский контур, админская плоскость, сервисные учётки и control plane.

## E-commerce / логистика / ops - где чаще прячется цепочка

На первый взгляд e-commerce выглядит как чистый web/API, но реальная среда обычно шире: **storefront, CMS, ERP/WMS, кабинеты поставщиков, платежные интеграции, логистические панели, SSO, VPN для подрядчиков, иногда 1C/ERP и внешние API перевозчиков.** Тут часто много подрядчиков и интеграционных швов, а значит - токены, service accounts, shared secrets, исторические доступы и плохо разделённые роли. Для пентестера это одна из отраслей, где misconfiguration и authorization flaw могут дать больший эффект, чем экзотический RCE.

## Health-tech / медицина - почему здесь выше цена ошибки

В медицине и health-tech часто смешиваются классические web-приложения, внутренние порталы, терминальные среды, интеграции с лабораториями, imaging/PACS, VPN-доступ для подрядчиков, **старые Windows-сегменты и специализированные устройства.** Часть систем кастомна, но огромный пласт обычно составляют vendor solutions и плохо обновляемые интеграции.

Пентестеру приходится работать с ограничениями на exploitation, чувствительностью к downtime и повышенной ценностью evidence. Сила здесь не в громком exploitation, а в умении построить аккуратную attack path-модель.

Личный кабинет, расписание, CRM, лабораторные результаты, мобильный frontend. Ценность - **персональные данные, медкарты, страховые сведения, доступность сервиса.**

### Что спрашивать до заключения контракта или старта проекта

**Полезно заранее выяснять:**

- scope включает только публичный периметр или есть internal segment;
- есть ли VPN/VDI/jump host; что живет в облаке, а что on-prem;
- кто обслуживает сеть, IAM, endpoint и SOC;
- что кастомно, а что коробка; есть ли production-only зоны и насколько жёстки ограничения на exploitation.

Эти вопросы помогают заранее понять, в каком типе среды ты будешь силён и где интервьюеру полезно показывать зрелое понимание среды, а не просто набор техник.

**Хороший тон:**

Не драматизируй. Не каждая misconfig тянет на critical. Но и не обесценивай мелочи: пара "low" иногда собирается в очень неприятный chain.

## ЧТО ПРОИСХОДИТ ПОСЛЕ КОМПРОМЕТАЦИИ

В CTF часто достаточно взять root или domain admin и довольным уйти пить кофе. В реальном инциденте это только середина истории.

Ниже - **high-level картина** того, зачем злоумышленникам вообще нужен доступ и чем это бьет по бизнесу. Без операционных подробностей, но с фокусом на risk thinking.

- кража чувствительных данных и дальнейший шантаж/вымогательство;
- массовая выгрузка клиентской базы, документов, договоров, медицинских записей, исходников;
- саботаж: удаление логов, отключение сервисов, порча конфигов и scheduled jobs;
- ransomware-сценарий: шифрование файловых шар, виртуалок, бэкапов, остановка бизнес-процессов;
- компрометация доверенных каналов: рассылка фишинга от лица компании, захват почтовых ящиков, подмена реквизитов;
- дальнейшее развитие доступа: переход в облако, CI/CD, PAM, секрет-хранилища, VPN, почту и системы мониторинга.

Для pentester-а это важно по двум причинам: во-первых, помогает объяснять findings языком бизнеса; во-вторых, не дает скатываться в "я нашел shell, значит задача закрыта".

**Что ценят интервьюеры:**

Умение перевести технический доступ в бизнес-последствия: простой сервиса, утечка, fraud, регуляторные риски, потеря доверия, рост затрат на восстановление.

# ИНФРАСТРУКТУРНЫЙ PENTEST: БАЗА БЕЗ ВОДЫ

Если говорить честно, в 2026 на собеседованиях по pentest очень часто побеждает не тот, кто помнит больше названий CVE, а **тот, кто уверенно мыслит по инфраструктуре:** видит trust boundaries, понимает AD-цепочки, не теряется в Linux-базе и умеет объяснить, зачем вообще бизнесу опасна конкретная misconfig.

Поэтому ниже - короткий, но плотный блок именно про **инфраструктурное мышление** (mindset этичного хакера\пентестера). Это не "магическая шпаргалка на все случаи", а каркас, который помогает говорить взросло, собранно и по делу.

## Windows / Active Directory

Для Windows/AD зрелый подход начинается с понимания управленческой плоскости: домен, роли, доверия, сервисные учётки, GPO, системы бэкапа, VDI/jump hosts, relay-поверхность и контрольные узлы вроде CA, hypervisor и backup. Интервьюеру обычно важно услышать не только «Kerberoasting/BloodHound», а приоритет: что даёт visibility, что даёт путь к control plane, что безопасно проверять руками и как быстро отделить шум от по-настоящему ценной гипотезы.

Начинай не с "как бы эскалироваться", а **с карты доверия:** домены, forest, tiering, jump hosts, админские группы, сервисные аккаунты, где лежат админские сессии.

На интервью сильнее звучит связка: SMB/LDAP/Kerberos/WinRM + что ты хочешь из этого понять. Не просто "запущу тул", а **"проверю валидные пути к lateral movement и делегированию прав".**

AD любят спрашивать через **бытовые сценарии:** слабый low-priv user, service account без контроля, stale admin session, уязвимая delegation-модель, LAPS/GPO/ACL misconfig, relay-поверхность.

Если получил первичный доступ, думай не только про DA. Ищи data access, control plane, возможность устойчивого удержания, точки скрытого влияния на бизнес-процессы.

## Linux / *nix

В Linux/*nix зрелость часто проявляется в том, как ты смотришь на сервисы и операционные паттерны, а не только на local privesc. SSH, sudoers, cron/systemd timers, capabilities, docker/socket access, NFS/Samba shares, backup scripts, CI/CD runners, kubeconfigs, секреты в env/config-файлах - всё это обычно интереснее, чем «сразу гонять linpeas и надеяться». Сильный кандидат объясняет, где искать реальные следы управления, автоматизации и накопленного техдолга.

Частая ошибка кандидатов - сразу уходить в linpeas и "авось что-то выпадет". На сильном интервью **лучше показать логику**: sudo -l, capabilities, writable paths, cron/systemd, SUID/SGID, контейнеры, secrets в history/env/config.

**Ценится понимание контекста:** root не всегда нужен. Иногда достаточно доступа к backup-пути, kubeconfig, cloud credentials или internal API token.

**Умей объяснить, почему мир Linux-пентеста** - это не только privesc, но и опора для pivoting, data collection, persistence и abuse доверия между сервисами.

## Проверка сетевых устройств и security appliances

Когда говорят про инфраструктурный pentest, сетевые устройства часто незаслуженно упрощают до «ну это же просто железка». На практике routers, switches, firewalls, VPN-шлюзы, reverse proxies, load balancers, mail gateways и NAC/WAF — это узлы, которые либо дают сильную точку наблюдения, либо открывают **путь к management/control plane.**

Смотреть на них полезно в двух плоскостях. **Первая — data plane:** как ходит трафик, где терминируется TLS/SSL, где живут ACL/NAT/segmentation, как организован VPN-доступ и через какие устройства можно обойти изоляцию. **Вторая — management plane:** web-admin, API, SSH, SNMP, TACACS+/RADIUS, локальные учётки, backup/export config, diagnostic bundles, сертификаты и firmware/update-механика.

**Типовые верхнеуровневые вопросы:** exposed admin interfaces, local admin без MFA, старые cipher suites, небезопасный экспорт конфигов, AAA-интеграции с AD/LDAP/RADIUS, уязвимые VPN-порталы, неправильная сегментация management-сети, наследие в виде Telnet/SNMPv2c и чрезмерные права сервисных аккаунтов.

Для switches и маршрутизаторов важен не только баннер версии, но и контекст: **VLAN design, trunk/access-порты, inter-VLAN routing, out-of-band management,** доступ подрядчиков и реальная граница между user VLAN и административным контуром. Даже если ты не делаешь глубокий network exploitation, умение правильно читать такую среду очень ценится на интервью.

## Что полезно уметь проговаривать на интервью

**Зрелый ответ звучит так:** сначала определить роль устройства, потом понять, где management plane, затем проверить exposure, auth и интеграции, и только после этого решать, есть ли смысл углубляться в эксплуатацию. Интервьюеру важнее модель мышления, чем редкий one-liner под конкретный вендор.

**Хорошо работает формулировка:** «я бы отдельно оценил, может ли компрометация этого узла привести к обходу сегментации, к перехвату трафика, к захвату VPN-плоскости или к доступу к PKI/IAM/control plane». Это сразу показывает взрослый offensive mindset.

## Граница advanced-уровня: reverse engineering протоколов и deep packet analysis

У специалистов очень высокого уровня эта тема уходит дальше обычного pentest: ***reverse engineering проприетарных протоколов, packet-level analysis, написание собственных парсеров/декодеров, исследование firmware и нестандартных control-plane сценариев.*** Здесь уже нужны глубокие знания сетевого стека Windows/Linux, бинарных форматов и того, как реально работает транспорт на уровне пакетов.

Для этой брошюры это скорее ориентир, чем обязательный baseline. Детально сюда не уходим, но важно отметить: именно такие навыки

отличают очень сильного инфраструктурного специалиста от просто уверенного практика.

## Красные флаги на интервью

Ниже - не просто список «что плохо говорить», а сигналы, по которым интервьюер пытается понять, есть ли у кандидата инженерная дисциплина.

Красные флаги обычно возникают там, где человек романтизирует эксплуатацию, игнорирует scope/ограничения, не умеет разделять hypothesis/evidence и путает владение инструментом с пониманием среды.

- Кандидат не уточняет scope и ограничения. Это почти всегда тревожный звонок.
- Ответы сводятся к перечню тулов без объяснения, что именно ими проверяется.
- Человек драматизирует каждую низкую misconfig как critical - это выглядит незрело.
- Кандидат не умеет отделять finding от impact и не может объяснить бизнес-последствие.
- Есть obsession на destructive exploitation: сразу "выкатил эксплойт", "уронил сервис", "скинул LSASS", хотя вопрос был про безопасную валидацию.
- В ответах нет reporting-мышления: нет evidence, reproducibility, remediation logic.
- Человек упирается в CTF-подход: root/DA как финальная цель, без понимания, что в реальном проекте ценность - в доказуемом риске и аккуратном deliverable.

# БОЕВЫЕ ИНТЕРВЬЮ-ВОПРОСЫ

Ниже - стартовая пачка вопросов, которые реально проверяют не только знания, но и мышление. Ответы вынесены отдельными подзаголовками, чтобы их можно было сворачивать.

## ВОПРОСЫ ОБЩЕГО ТОЛКА

### Вопрос 1. Что такое penetration testing и чем он отличается от vulnerability scanning?

***Ответ / разбор***

Пентест - это управляемая имитация атаки в согласованном scope. Scanner в основном помогает находить признаки уязвимостей по сигнатурам и конфигам, а pentest проверяет реальные paths exploitation, сочетание слабостей, impact и устойчивость защит.

### Вопрос 2. Какие стадии ты считаешь обязательными даже для маленького пентеста?

***Ответ / разбор***

Pre-engagement, recon, enumeration, validation/exploitation в рамках scope, impact assessment, аккуратный reporting и debrief. Пропуск pre-engagement и debrief - типичная ошибка джунов.

### Вопрос 3. Почему enumeration часто важнее exploitation?

***Ответ / разбор***

Потому что хорошие находки обычно рождаются не из 'магического эксплойта', а из качественного понимания поверхности атаки: кто с кем разговаривает, какие доверия есть, где старые endpoints и как устроен auth flow.

### Вопрос 4. Что ты будешь делать первым делом, если видишь внешний веб-периметр компании?

***Ответ / разбор***

Соберу attack surface: поддомены, технологии, входные точки, auth, API, upload, robots.txt, JS, dev/stage endpoints, CDN/WAF-поведение, default панели и reset/login flows.

### Вопрос 5. Что такое false positive и как ты его отрабатываешь?

***Ответ / разбор***

Не тащу находку в отчет до внятного подтверждения. Ищу воспроизводимость, несколько независимых признаков, валидирую на test data и четко фиксирую ограничения, если impact частично предположительный.

### Вопрос 6. Чем BOLA/IDOR опаснее, чем кажется на первый взгляд?

***Ответ / разбор***

Потому что это не просто 'чужой объект видно'. Это часто прямой путь к горизонтальному доступу, обходу бизнес-логики, утечке PII и эскалации в аккаунт с более высокой ценностью.

### Вопрос 7. Что бы ты искал в Windows/AD среде на интервью-задании?

***Ответ / разбор***

Shares, SMB signing, local admin reuse, weak ACL, Kerberoast/AS-REP roast conditions, delegation issues, legacy protocols, доступы сервисных аккаунтов, misconfig GPO, paths в BloodHound.

### Вопрос 8. Чем плоха позиция 'я все делаю через Metasploit'?

***Ответ / разбор***

Она палит отсутствие глубины. На сильном интервью ожидают понимание протоколов, manual validation, логики эксплуатации и умения объяснить, почему конкретный шаг вообще имеет смысл.

### Вопрос 9. Что важно при проверке file upload?

***Ответ / разбор***

Server-side validation, content-type spoofing, extension handling, storage path, execution context, image processing, SSRF/path traversal around upload pipeline, public accessibility и object permissions.

### Вопрос 10. Как отличить сильный pentest report от слабого?

***Ответ / разбор***

Сильный report читается и технарем, и менеджером: там есть ясный риск, evidence, воспроизводимость, влияние на бизнес, приоритет фикса и адекватные рекомендации без воды.

## ВОПРОСЫ НА ПРОВЕРКУ МЫШЛЕНИЯ И ТЕХ НЮАНСЫ

### Вопрос 11. Что делать, если у тебя есть только low-priv доменный пользователь, а среда большая и шумная?

***Ответ / разбор***

Сначала не бегу "ломать домен". Я бы спокойно подтвердил границы видимости: доступные shares, auth к SMB/LDAP/WinRM, наличие сервисных записей, интересные группы, типовые misconfig по ACL/GPO, следы админских сессий и общую форму графа доверия. В сильном ответе важно показать restraint: минимальный шум, аккуратная проверка гипотез, понимание, что даже без DA можно найти чувствительные данные, ключи, сервисные токены и реалистичный путь к chain.

### Вопрос 12. Чем Kerberoasting отличается от AS-REP Roasting и почему интервьюеры любят этот вопрос?

***Ответ / разбор***

Потому что вопрос проверяет, понимаешь ли ты Kerberos-модель, а не просто видел модные слова. Kerberoasting связан с сервисными аккаунтами и TGS, AS-REP Roasting - с аккаунтами без pre-auth. На интервью лучше объяснить не механику атаки по шагам, а логику риска: слабый пароль у сервисной или legacy-учетки может превратить "обычную" конфигурационную слабость в путь к escalated access.

### Вопрос 13. Почему SMB signing, NTLM relay и общая relay-поверхность до сих пор важны в инфраструктурных тестах?

***Ответ / разбор***

Потому что relay - это не про "старый мем", а про реальную поверхность доверия в enterprise. Когда часть протоколов и служб

допускает небезопасные сценарии аутентификации, атакующий может превращать сетевую доступность и принужденную аутентификацию в расширение контроля. Сильный кандидат не просто называет relay, а объясняет условия, ограничения и как это переводится в риск для AD-окружения.

## Вопрос 14. Когда имеет смысл вспоминать LLMNR/NBT-NS/mDNS poisoning на интервью, а когда лучше не притягивать за уши?

***Ответ / разбор***

Имеет смысл тогда, когда есть разговор про внутреннюю сеть, legacy-хосты, слабую сегментацию и типовые протокольные поверхности. Не стоит звучать как человек, который любую сеть сводит к responder-style истории. Лучше показать здравый смысл: "проверю, релевантно ли это среде, и не буду делать вид, что один паттерн объясняет всю инфраструктуру".

## Вопрос 15. Как ты подходишь к Linux privilege escalation так, чтобы это выглядело профессионально, а не как запуск набора автоскриптов?

***Ответ / разбор***

Говорю через приоритеты. Сначала контекст: кто я, где я, что уже доступно. Затем быстрые ручные проверки: sudo -l, writable paths, capabilities, cron/systemd, SUID/SGID, интересные конфиги, env, history, секреты и контейнерный контекст. Автоэнумерация допустима, но как помощник, а не замена мышлению. Это обычно нравится интервьюерам.

## Вопрос 16. В чем разница между pivoting, port forwarding и SOCKS tunneling, если объяснять без академической воды?

***Ответ / разбор***

Port forwarding - это, грубо говоря, "протянуть" конкретный порт через уже доступную точку. SOCKS - более гибкий транспортный прокси для целого набора соединений. Pivoting как интервью-

термин шире: это сам подход к расширению зоны досягаемости через скомпрометированную систему. Сильный ответ показывает, что ты понимаешь, когда нужен точечный доступ, а когда нужен устойчивый маршрут в соседний сегмент.

## Вопрос 17. Почему BloodHound полезен, но почему опасно строить весь ответ только вокруг него?

***Ответ / разбор***

Потому что BloodHound хорошо визуализирует attack paths и отношения в AD, но он не заменяет понимание среды и не является "истиной в последней инстанции". На интервью лучше сказать так: это хороший инструмент для приоритизации и объяснения цепочек, но findings все равно нужно подтверждать вручную и интерпретировать с учетом контекста.

## Вопрос 18. Что бы ты проверил в сценарии "есть VPN и доступ к jump host, но дальше все закрыто"?

***Ответ / разбор***

Сначала - что именно реально "закрыто". Видимость сети, DNS, WinRM/SMB/RDP/SSH reachability, локальные артефакты на jump host, токены, сохраненные сессии, доверенные соединения, инженерные средства администрирования, прокси-настройки, split tunneling, маршруты и правила сегментации. Очень часто бизнес-риск прячется не в экзотике, а в плохо контролируемой точке административного доступа.

## Вопрос 19. Как объяснить интервьюеру разницу между "есть доступ к одному серверу" и "есть доступ к control plane"?

***Ответ / разбор***

Один сервер - это еще не обязательно катастрофа. Но если через него можно влиять на управление, развертывание, учетные записи, резервные копии, CI/CD, доменные политики или другие доверенные механизмы, риск качественно меняется. На хорошем

интервью важно уметь различать просто компрометацию узла и компрометацию точки, которая управляет остальными.

## Вопрос 20. Как подтвердить серьезный impact, не переходя в разрушительные действия?

***Ответ / разбор***

Через безопасное доказательство. Покажи, что доступ реален, данные достижимы, управление возможно, chain правдоподобен и вред достижим без "демонстрации пожара". Скриншоты, sanitized evidence, ограниченное чтение, контрольные артефакты, описание потенциального blast radius и четкое пояснение business impact обычно гораздо ценнее, чем шумный destructive demo.


PRACTICAL TASKS
& HANDS-ON CASES
ЗАДАНИЯ, КЕЙСЫ И СНИППЕТЫ КОДА
ПОШАГОВЫЕ ЭКСПЛОИТЫ • ОБХОД СИСТЕМ • БОЕВОЙ КОД

# ПРАКТИЧЕСКИЕ ЗАДАНИЯ

Задачи ниже являются безопасными и учебными, но очень часто встречающиеся на интервью в том или ином несколько измененном виде. Логика такая же, как на хорошем screening: тебе дают кусок вывода, артефакт, запрос или конфиг и смотрят, как ты мыслишь.

## Задание 1. Sanitized nmap output

**Ниже дан фрагмент сканирования. Что ты выделишь как приоритет?**

```
22/tcp      ssh     OpenSSH     8.9
80/tcp      http    nginx       1.22
443/tcp     https   nginx       1.22
445/tcp                         microsoft-ds
8080/tcp http-proxy Jenkins
```

***Решение / комментарий***

Ответ: сразу интересны 445/tcp и 8080/tcp. SMB - потенциальный pivot в Windows-мир, Jenkins - возможный CI attack surface, secrets exposure, build logs, script console, agents, outdated plugins. 80/443 идут в базовую web-проверку, но 8080 и 445 обычно дают более жирную развилку.

## Задание 2. HTTP response triage

**Есть ответ API: 200 OK, тело содержит user_id, role, company_id, internal_notes. Что проверишь дальше?**

```
GET /api/v2/profile?id=1042
```

***Решение / комментарий***

Ответ: проверка BOLA/IDOR через соседние идентификаторы, вертикальный доступ к internal_notes, consistency между UI и API, authz на уровне объекта, массовое перечисление, leakage через mobile/web клиенты.

## Задание 3. Linux privilege escalation thought process

**Ты получил shell от low-priv user в учебном стенде. Что смотришь без фанатизма? Куда лезть дальше?**

```
id; sudo -l; find / -perm -4000 -type f 2>/dev/null
```

***Решение / комментарий***

Ответ: sudo rights, SUID/SGID, writable service files, timers/cron, env leaks, credentials in home/service configs, container escapes only если есть явные признаки. На интервью важно не перечислить 200 техник, а показать порядок проверки и приоритизацию.

## Задание 4. PowerShell artifact

**Что видно из их текущего фрагмента? У тебя 3 минуты на анализ.**

```
Get-ChildItem Env: | ? { $_.Name -match 'KEY|TOKEN|SECRET' }
```

***Решение / комментарий***

Ответ: оператор ищет потенциальные секреты в переменных окружения. Для pentester-а это ценная быстрая проверка на exposed creds, API tokens, cloud keys в CI/Windows-сервисах. Обязательно помнить про scope и аккуратность с чувствительными данными.

## Задание 5. Burp / web logic

**В запросе reset password ссылка активна 24 часа и не инвалидируется после использования. Что это?**

```
POST /auth/reset/confirm
```

***Решение / комментарий***

Ответ: weakness в password reset workflow. Риски – replay, account takeover, race conditions, reuse links from logs/mailboxes. Дополнительно проверяются rate limits, token entropy, binding to user/session/device.

## Задание 6. Docker mini-lab

**Контейнер запущен с volume на /var/run/docker.sock. Почему это красный флаг?**

```
docker run -v /var/run/docker.sock:/var/run/docker.sock app
```

***Решение / комментарий***

Ответ: доступ к docker.sock часто означает фактический root на хосте через управление контейнерами. На интервью важно объяснить impact и remediation: не монтировать сокет без крайней необходимости, использовать rootless/isolated runners и минимальные привилегии.

## Задание 7. BloodHound graph interpretation

**Условие:** в sanitized-графе видно, что user helpdesk01 входит в группу Account Operators, имеет local admin на SRV-APP-02, а на SRV-APP-02 регулярно логинится svc-backup. Дополнительно известно, что svc-backup состоит в Backup Operators на нескольких серверах. Вопрос: что здесь выглядит как перспективный attack path и что надо проверить в первую очередь?

***Решение / комментарий***

Здесь важно не фантазировать, а аккуратно описать цепочку доверия. Перспективный путь - использовать control над SRV-APP-02 как точку для проверки артефактов и сессий svc-backup, затем оценить, во что конвертируются права Backup Operators и есть ли выход на более чувствительные системы. На интервью ценится формулировка "сначала подтвержу факты: тип доступа, реальная сессия, хранимые секреты, reachability, ограничения среды".

## Задание 8. SMB signing и relay opportunity

**Условие:** внутренний Windows-хост отвечает по SMB, SMB signing на части сегмента не enforced, есть несколько принтеров и старых серверов, LDAP signing включен не везде. Вопрос: почему это настораживает пентестера даже без "готового эксплойта в кармане"?

***Решение / комментарий***

Потому что это признак relay-поверхности и слабой протокольной гигиены. Даже без немедленной эксплуатации такой набор говорит, что аутентификационные потоки и сервисное доверие могут быть использованы для расширения доступа. Хороший ответ подчеркивает условия релевантности, ограничения и необходимость безопасной проверки без лишнего шума.

## Задание 9. Linux sudoers excerpt

**Условие:** пользователь deploy может выполнять без пароля /usr/bin/tar и /usr/bin/systemctl status *.service. Вопрос: какая строка выглядит опаснее и почему? Что должен увидеть зрелый кандидат?

***Решение / комментарий***

Опаснее обычно правило с tar, если оно допускает сценарии выхода за ожидаемую операцию и дает путь к выполнению команд или обходу ограничений. systemctl status само по себе не всегда страшно. Сильный кандидат не кричит "root за 2 секунды", а объясняет, что риск зависит от точной конфигурации, пути запуска и фактических ограничений sudoers.

## Задание 10. Pivoting через jump box

**Условие:** так, у тебя есть shell на jump-хосте в сегменте админов, прямого доступа к серверам баз данных нет, но с jump-хоста они доступны. Вопрос: какое интервью-мышление здесь ждут, если не просят конкретную команду?

***Решение / комментарий***

Ждут, что ты опишешь сам подход: оценить reachable services, понять, нужен ли точечный port forward, SOCKS или другой способ безопасно исследовать сегмент, не ломая сеть и не генерируя лишний шум. То есть показать понимание транспортного слоя и план действий, а не только набор CLI-флагов.

## Web/API triage: четыре практических сниппета с разбором

### Сниппет 1. IDOR/BOLA на GET

```
GET /api/v1/invoices/10428 HTTP/1.1
Host: app.example.test
Authorization: Bearer eyJ...
```

**Разбор:** сначала проверяешь, действительно ли объект принадлежит текущему пользователю, затем меняешь идентификатор на соседний или предсказуемый и смотришь не только на код ответа, но и на структуру JSON, размер, наличие чужих полей и косвенные признаки утечки.

На интервью лучше проговаривать, что BOLA/IDOR — это не «угадай ID», а нарушение object-level authorization. Сильный finding здесь строится вокруг boundary между пользователями/арендаторами и реального impact на доступ к данным.

### Сниппет 2. Mass assignment / role field

```
POST /api/v1/profile HTTP/1.1
Content-Type: application/json
{"displayName":"ivan", "role":"admin", "isInternal":true}
```

**Разбор:** Проверка делается ступенчато: сначала безопасное поле-маркер, потом логически чувствительное поле. Ты смотришь, принимает ли сервер лишние атрибуты, отражает ли их в ответе и меняется ли поведение модели после update/binding.

Правильная формулировка на интервью: проблема не в магическом role=admin, а в отсутствии allow-list/strict binding на стороне сервера.

Это системная misconfiguration в update-логике, а не случайный трюк с JSON.

### Сниппет 3. Security misconfiguration в ответе reverse proxy

```
HTTP/1.1 200 OK
Server: nginx/1.18.0
X-Upstream-Server: internal-api-03
X-Env: prod-eu-west-1
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true
```

**Разбор:** Здесь важно разделить recon-value и реально опасную misconfiguration. Версия, внутренний hostname и env-тег полезны для картирования среды, но отдельный red flag — комбинация ACAO:* и credentials=true, которую нужно проверять на фактическом browser/client behavior.

**Хороший follow-up:** origin reflection, preflight, cookie mode, чувствительные методы и реальные ограничения фронтенда. Это звучит взрослее, чем просто перечислить заголовки как чеклист.

### Сниппет 4. Suspicious file upload

```
POST /upload HTTP/1.1
Content-Type: multipart/form-data
file=invoice.jpg.php; Content-Type: image/jpeg
```

**Разбор:** Взрослый разбор начинается не с мечты об RCE, а с пайплайна загрузки: приём, валидация extension/content-type/signature, преобразование, storage path, publication path и downstream consumers вроде image processor, AV, queue или object storage.

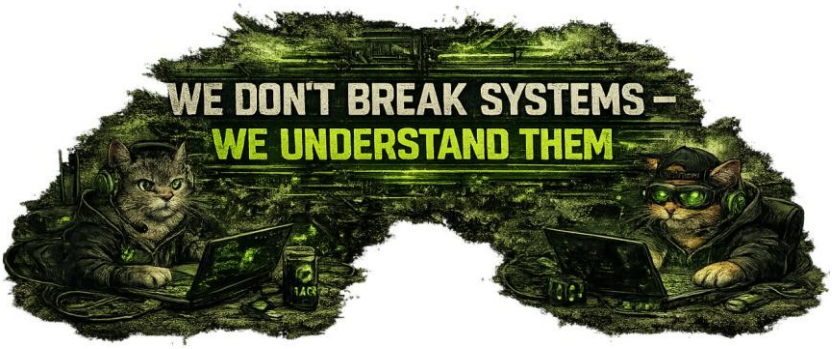
На интервью сильнее звучит кандидат, который умеет объяснить, где именно ломается boundary: в upload validation, в storage policy, в обработчике изображений или в том, как файл затем публикуется/исполняется.

## Kill chain, attack path и lateral movement

Хороший кандидат объясняет эти термины не как набор модных слов, а как способ мыслить о цепочке компрометации. Kill chain помогает видеть этапы атаки, **attack path - конкретную техническую дорогу** от точки входа до ценного актива, а lateral movement - не цель сам по себе, **а механизм расширения reachability и доверия**. На интервью особенно ценится мысль: небольшая misconfiguration может быть слабой сама по себе, но очень сильной в составе цепочки.

Хороший pentester в инфраструктуре мыслит не отдельной уязвимостью, а цепочкой. Примерно так: **initial access -> foothold -> credential access -> privilege escalation -> lateral movement -> access to crown jewels -> business impact.**

На интервью это полезно проговаривать прямо. Такой ответ показывает, что ты понимаешь не только поиск дыр, но и реальную механику инцидента. То есть не "я взял shell и радуюсь" (как в большинстве CTF соревнованиях\комнатах на площадках для практики), а "я вижу, как этот доступ может превратиться в утечку, простой, fraud, шифрование, takeover критичных сервисов".

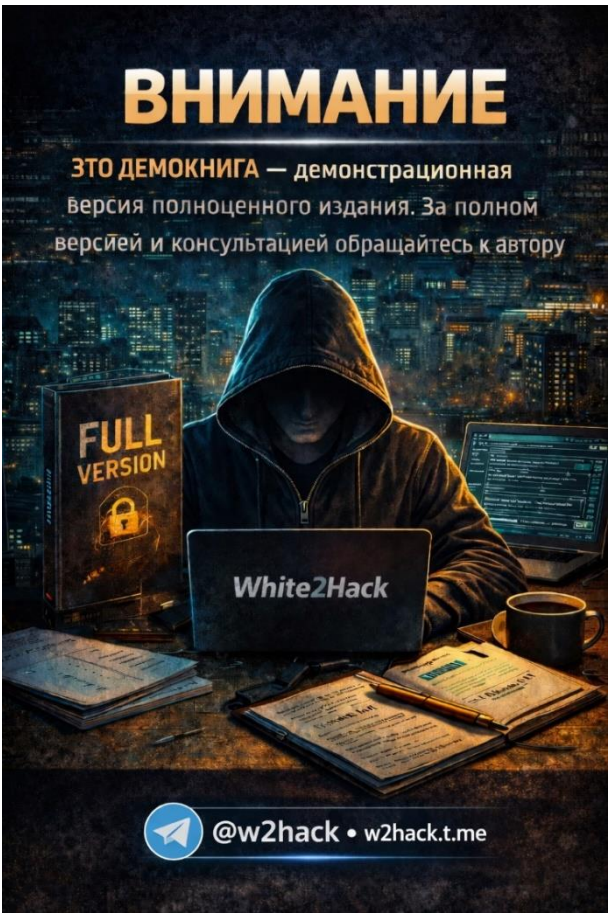


# !!! Attention !!!

Это демо-книга. Перед тобой демонстрационная версия полноценного учебного издания для самостоятельной базовой подготовки к интервью на позицию pentester-а.

За полной версией, персональным разбором, адаптацией под стек и конкретную вакансию обращайтесь к автору - Ivan Piskunov | white2hack.

**Demo notice:** В финальной полной версии обычно больше вопросов, больше практических заданий, дополнительные схемы, разборы кейсов и точечные рекомендации под уровень кандидата.




КВИЗ
ПРОВЕРКА ЗНАНИЙ ПО PENTEST-А
PENTEST QUIZ
What tool do you need to scan for:
What tool do you need to scan for
Nmap
BloodHound
Hydra
ВОПРОСЫ • СЦЕНАРИИ • ПРАВИЛЬНЫЕ ОТВЕТЫ • ОБЪЯСНЕНИЯ

# КВИЗ ДЛЯ САМОПРОВЕРКИ

Быстрый self-check перед интервью. Сначала отвечай сам, потом разворачивай разбор. Это самодиагностика поэтому не подсматривай раньше времени!

## Quiz 1. Какой инструмент чаще всего берут для ручного web testing?

A) Burp Suite

B) Nessus

C) Nuclei

D) Splunk

***Правильный ответ / почему так***

**Правильный вариант: A**

Смотри не только на букву, но и на reasoning. На реальном интервью почти всегда просят пояснить, почему именно этот вариант, какие есть исключения и где это может быть не так однозначно.

## Quiz 2. Что важнее после finding:

A) громкое название

B) exploit meme

C) подтвержденный impact и evidence

D) CVSS в вакууме

***Правильный ответ / почему так***

**Правильный вариант: C**

Смотри не только на букву, но и на reasoning. На реальном интервью почти всегда просят пояснить, почему именно этот вариант, какие есть исключения и где это может быть не так однозначно.

## Quiz 3. Что из этого ближе к object-level authorization flaw?

A) SQL backup открыт по FTP

B) чужой invoice читается по изменению ID

C) self-signed cert

D) banner leakage

***Правильный ответ / почему так***

**Правильный вариант: B**

Смотри не только на букву, но и на reasoning. На реальном интервью почти всегда просят пояснить, почему именно этот вариант, какие есть исключения и где это может быть не так однозначно.

### Quiz 4. Какая практика делает report сильнее?

A) скрин без контекста

B) один PoC на словах

C) reproducible steps + artifacts + business impact

D) больше CAPS LOCK

***Правильный ответ / почему так***

**Правильный вариант: C**

Смотри не только на букву, но и на reasoning. На реальном интервью почти всегда просят пояснить, почему именно этот вариант, какие есть исключения и где это может быть не так однозначно.

### Quiz 5. Что обычно НЕ стоит делать сразу после первой интересной находки?

A) фиксировать evidence

B) оценить scope

C) потерять полдня в tunnel vision

D) проверить соседние attack paths

***Правильный ответ / почему так***

**Правильный вариант: C**

Смотри не только на букву, но и на reasoning. На реальном интервью почти всегда просят пояснить, почему именно этот вариант, какие есть исключения и где это может быть не так однозначно.

### Quiz 6. Для AD triage какой набор выглядит разумно?

A) Burp + sqlmap

B) BloodHound + Impacket + NetExec

C) Wireshark + Photoshop

D) только Metasploit

***Правильный ответ / почему так***

**Правильный вариант: B**

Смотри не только на букву, но и на reasoning. На реальном интервью почти всегда просят пояснить, почему именно этот вариант, какие есть исключения и где это может быть не так однозначно.

## Quiz 7. Что выглядит сильнее на интервью: "я запускаю linpeas" или "сначала проверяю sudo, capabilities, writable paths и только потом автоэнумерацию"?

***Правильный ответ / почему так***

Второй вариант. Интервьюеры обычно хотят увидеть мышление, а не рефлекс на одну утилиту.

## Quiz 8. Если после внешнего foothold можно достучаться до CI/CD, backup или IAM control plane, как обычно меняется риск?

***Правильный ответ / почему так***

Риск растет резко, потому что компрометация control plane почти всегда опаснее компрометации одного узла.

## Quiz 9. Что важнее в хорошем finding: громкий technical detail или связка evidence + impact + remediation?

***Правильный ответ / почему так***

Связка evidence + impact + remediation. Без этого finding плохо продается и для заказчика, и на интервью.

## Quiz 10. Какой ответ взрослее: "нужно обязательно добивать до DA" или "нужно понять, достаточно ли текущего доступа для доказуемого риска"?

***Правильный ответ / почему так***

Второй. В реальном pentest цель - не трофей, а понятный и безопасно подтвержденный риск.

### Quiz 11. Что чаще является red flag: признать, что не знаешь точный флаг команды, или делать вид, что знаешь и нести чушь?

***Правильный ответ / почему так***

Второе. Честность плюс адекватное рассуждение почти всегда лучше, чем уверенная фантазия.

### Quiz 12. При обсуждении SMB/NTLM/LDAP что обычно важнее: знание модного слова или понимание, какие trust-отношения это открывает?

***Правильный ответ / почему так***

Понимание trust-отношений и реальной attack surface.

### Quiz 13. Какой ответ про BloodHound лучше: "это серебряная пуля" или "это хороший графовый инструмент, но findings нужно подтверждать"?

***Правильный ответ / почему так***

Второй. Это зрелее и ближе к реальной практике.

### Quiz 14. Что более профессионально после первой интересной находки: углубляться в нее до упора или оценить, не важнее ли сейчас расширить картину среды?

***Правильный ответ / почему так***

Сначала оценить приоритет и контекст. Иногда одна уязвимость стоит глубокого копания, иногда важнее не потерять общий attack path.

### Quiz 15. Какой подход к segmentation звучит взрослее: "сеть есть сеть" или "важно понять, какие зоны доверяют друг другу и через что это доверие реализовано"?

***Правильный ответ / почему так***

Второй. Именно так обычно и ищут путь к реальному impact.

### Quiz 16. Если кандидат не умеет объяснить разницу между finding, exploitability и business impact, это хороший или плохой сигнал?

***Правильный ответ / почему так***

Плохой. Это означает, что отчетность и приоритизация у него могут проседать.

### Quiz 17. Что обычно ценнее для junior->middle перехода: редкая экзотика или уверенная база по AD, Linux, web-logic и reporting?

***Правильный ответ / почему так***

Уверенная база. Именно она чаще всего и приводит к хорошему офферу.

### Quiz 18. Если на интервью спрашивают про pivoting, чего обычно ждут первым делом?

***Правильный ответ / почему так***

Не конкретный one-liner, а понимание сетевой досягаемости, транспорта и безопасного плана действий.

### Quiz 19. Какой ответ лучше: "любая низкая misconfig - ерунда" или "низкие находки тоже важны, если собираются в chain"?

***Правильный ответ / почему так***

Второй. Отдельно мелочь может быть low, но в цепочке - уже совсем другой разговор.

## Quiz 20. Что чаще всего отличает сильного pentester-кандидата от просто "любителя CTF"?

***Правильный ответ / почему так***

Умение мыслить через scope, evidence, impact, reporting и реальные бизнес-последствия, а не только через получение shell.


КАК ПИСАТЬ ОТЧЕТ
ИНСТРУМЕНТЫ, ПРАКТИКИ И ЛУЧШИЕ ПРИНЦИПЫ
PENTEST REPORT
PENTEST REPORT
EVIDENCE И ВИДЕО • ЧЕКЛИСТЫ И РЕКОМЕНДАЦИИ • КОММУНИКАЦИЯ

# КАК ПИСАТЬ ОТЧЕТ ПОСЛЕ ПЕНТЕСТА

Сильный report - это не свалка скриншотов и не "proof we hacked it". Это документ, по которому бизнес понимает риск, а инженерная команда может воспроизвести проблему и пофиксить ее без догадок.

## Структура отчета

**Executive summary:** 1-2 страницы для руководства без лишнего техжаргона.

**Scope и Rules of Engagement:** что входило, что не входило, когда тестировалось и с какими ограничениями.

**Methodology:** какие подходы и источники использовались.

**Findings:** каждая проблема отдельно, с risk rating, evidence, affected assets, reproduction и remediation.

**Appendix:** test accounts, timestamps, списки активов, raw artifacts по необходимости.

## Стиль изложения

**Короткие предложения и ясная логика.** Не пытайся казаться умнее текста.

**Техническая часть** - конкретная и воспроизводимая**.**

**Менеджерская часть** – спокойная, без страшилок, связывай технику с бизнес impact.

**Избегай vague-формулировок вроде** "может привести к серьезным последствиям" без пояснения каких именно.

## Как подтверждать findings

### Мини-примеры findings и короткие шаблоны отчета

Ниже не «идеальный enterprise-report», а короткие образцы, чтобы junior видел форму мысли: **контекст -> evidence -> impact -> remediation.** На интервью полезно показать, что ты умеешь не только найти проблему, но и оформить её так, чтобы инженер понял, что делать дальше.

**Пример 1. Недостаточная сегментация между пользовательским VLAN и сервером резервного копирования.** Во время внутреннего теста из пользовательского сегмента был подтверждён сетевой доступ к backup-серверу по SMB и WinRM. Удалось безопасно подтвердить наличие административной поверхности без разрушительных действий. Риск в том, что компрометация рабочей станции может стать ступенькой к системам восстановления и данным повышенной критичности. Рекомендации: пересмотреть ACL/маршрутизацию между зонами, ограничить административные протоколы, выделить отдельную управляющую плоскость и включить мониторинг нештатного доступа.

**Пример 2. Object-level authorization flaw в API кабинета партнёра.** Аутентифицированный пользователь с ролью оператора мог изменять идентификатор объекта в запросе GET/POST и получать доступ к данным другого партнёра. Уязвимость подтверждена на тестовых сущностях без изменения боевых данных. Бизнес-риск - утечка информации и нарушение целостности партнерского контура. Исправление: server-side проверка владения объектом, отрицательные тесты на уровне API, журналирование попыток доступа к чужим объектам и регрессионные проверки в CI.

**Мини-шаблон finding:** Заголовок -> Краткое описание -> Затронутые активы -> Preconditions -> Evidence -> Impact -> Likelihood / Exploitability -> Recommendations -> Notes / Limitations. Даже если report небольшой, эта структура дисциплинирует и помогает не превращать finding в поток сознания.

## Типовые ошибки в pentest report

***Ошибка №1:*** писать finding как поток сознания. Хороший finding строится вокруг структуры: что подтверждено, где подтверждено, какой impact, какие ограничения проверки и что делать дальше.

***Ошибка №2:*** путать exploitability с business impact. То, что эксплуатировать неудобно, не всегда означает низкий риск; и наоборот, красивый PoC не гарантирует реальной бизнес-значимости.

***Ошибка №3:*** заливать отчет скриншотами без контекста. Скрин сам по себе слабый evidence, если рядом нет запроса/ответа, роли пользователя, актива, timestamp и короткого объяснения, что именно он доказывает.

***Ошибка №4:*** давать remediation в стиле «обновите всё». Сильнее работает конкретика: где именно проверять ACL, какую server-side проверку добавить, какой trust boundary пересмотреть, какой control включить.

***Ошибка №5:*** делать вид, что гипотеза уже доказана. Если часть сценария только предполагается, это нужно честно отделять от подтверждённой части. Такая дисциплина повышает доверие к report и к самому тестировщику.

## Как быстро разбирать незнакомый хост или сервис за первые 15 минут

**1) Сначала определи роль узла:** публичный web, jump host, AD-инфраструктура, middleware, CI/CD, backup, monitoring, VPN, appliance или dev/test-артефакт. Интервьюеру важна именно гипотеза о роли, а не бессвязный список портов.

**2) Собери минимальный контекст:** баннеры, версии, TLS, HTTP-заголовки, DNS-имена, сертификаты, SMB/LDAP/Kerberos-признаки, соседние сервисы, куда ведёт панель входа. Цель - понять trust-отношения и плоскости доступа.

**3) Реши, что проверять руками сразу, а что оставить на потом.** Ручной приоритет: auth boundary, role boundary, default/weak exposure, admin interfaces, file upload, segmentation, service accounts, relay surface.

**4) Не вцепляйся намертво в первую находку**. Сначала оцени, не ведёт ли она к control plane: backup, CI/CD, IAM, MDM, hypervisor, orchestrator, secrets или управляющей сети.

5) Сразу думай, как это попадёт в report: что уже есть как evidence, какой потенциальный impact и какие безопасные шаги нужны для подтверждения гипотезы.

## Что учить «руками», а что знать\понимать концептуально

Ниже не «жёсткая истина», а практичная карта приоритетов. Руками стоит качать то, что часто воспроизводится на интервью и в проекте. Концептуально — то, что помогает не теряться в незнакомой среде и правильно строить гипотезы.

| Что качать руками | Что понимать концептуально | Почему это важно |
|---|---|---|
| Burp / HTTP / API triage | Boundary authN/authZ, object model, data flow | Помогает видеть не только запрос, но и реальную модель доступа. |
| Nmap / httpx / ffuf / recon | Reachability, surface mapping, приоритизация attack surface | Хороший пентестер сначала понимает поверхность и доверие, а не бежит в exploitation. |
| Windows/AD hands-on: SMB, LDAP, Kerberos | Trust paths, relay surface, service accounts, control plane | Даёт возможность мыслить через цепочку компрометации, а не через одну технику. |
| Linux hands-on: sudo, services, cron, capabilities | Операционные паттерны, automation, persistence surfaces | Linux-среды часто ломаются не магией, а накопленным техдолгом и неверной автоматизацией. |
| Writing findings и remediation notes | Разница между evidence, exploitability и business impact | Именно здесь junior часто проигрывает кандидату с более зрелым мышлением. |
| Upload / reset flows / auth logic mini-labs | Pipeline thinking: validation -> storage -> publication | Учит видеть не только баг, но и всю цепочку обработки данных. |
| Базовая автоматизация на Python/Bash/PowerShell | Когда автоматизация ускоряет анализ, а когда маскирует непонимание | Сильный кандидат автоматизирует рутину, а не заменяет скриптом собственную голову. |

**Главная идея таблицы простая**: руками отрабатывай повторяемые навыки и артефакты, а концептуально учись понимать среду, trust-отношения и control plane. Именно это и отличает уверенного кандидата от человека, который просто насмотрелся write-up.

# ЛАБОРАТОРИИ И СТЕНДЫ ДЛЯ САМООБУЧЕНИЯ / ПРАКТИКИ

Ниже подборка практикумов, которые реально помогают вырасти без необходимости сразу покупать дорогую инфраструктуру и обучающие коммерческие курсы. А вот знать английский язык на B1 нужно минимум.

## PortSwigger Web Security Academy

Бесплатно, в браузере, почти эталон для web-практики. Хорош для auth, deserialization, SSRF, request smuggling, race conditions и современного web-mindset.

## TryHackMe

Подходит для структурного входа, guided rooms и базовой практики. Удобно закрывать gaps по enumeration, web, Linux/Windows priv-esc и AD.

## Hack The Box Academy / Labs

Более плотная hands-on среда. Под middle+ хорошо заходят path-based модули и CPTS-ориентированный подход.

## OWASP Juice Shop

Бесплатное insecure web-приложение, покрывающее широкий спектр типовых web flaws. Классика для тех кто учит веб-пентест!

## OWASP WebGoat / Security Shepherd

Нормальная база под объяснение web-категорий и безопасных учебных упражнений. Тоже база!

## GOAD

Один из лучших бесплатных AD-стендов для тренировки attack path analysis и типовых misconfig. По теме изучения безопасности инфраструктуры Microsoft это must have!

## Vulhub / Metasploitable / DVWA

Локальные мини-лабы через Docker/VM для отработки базовых сценариев без лишних затрат. Староваты, но в целом сойдет для базы.

## Мини-лаба дома

**Docker:** DVWA, Juice Shop, WebGoat, Security Shepherd, Vulhub - быстро поднять и снести.

**VM/VirtualBox:** Kali/Parrot + одна Windows VM + GOAD, если хватает железа.

**Браузерные платформы:** PortSwigger Academy, THM, HTB - минимальный friction для регулярной практики.

**Коммерческие варианты:**

Для серьезной прокачки в 2026 чаще всего берут HTB Academy/HTB Labs, платные пути TryHackMe и при необходимости OSCP/CPTS-ориентированную программу.

# БАЗОВЫЕ КНИГИ И РЕСУРСЫ

Ниже - компактная библиотека, которая реально помогает расти. Я бы не пытался читать все подряд: выбери 1-2 книги по web, 1-2 по инфраструктуре/AD и параллельно держи живые labs и wiki-ресурсы.

## Книги

Ниже — базовые книги и официальные ресурсы с прямыми ссылками на Издательства.

- The Web Application Hacker’s Handbook
- Real-World Bug Hunting
- The Hacker Playbook 3
- Practical Binary Analysis
- Black Hat Python, 2nd Edition
- Red Team Field Manual
- Blue Team Field Manual

## Онлайн-ресурсы

Где искать сильные референсы и тренировать насмотренность:

- PortSwigger Web Security Academy
- OWASP Web Security Testing Guide
- OWASP ASVS
- OWASP Cheat Sheet Series
- PentesterLab
- Hack The Box Academy
- TryHackMe
- BloodHound Docs
- Impacket
- NetExec
- Nuclei
- ffuf

- Amass
- PENTESTING-BIBLE
- Awesome Pentest

- ✓ HackTricks - https://hacktricks.wiki/en/index.html
- ✓ PortSwigger Web Security Academy - https://portswigger.net/web-security
- ✓ TryHackMe - https://tryhackme.com/
- ✓ Hack The Box Academy - https://academy.hackthebox.com/path/preview/penetration-tester
- ✓ OWASP Juice Shop - https://owasp.org/www-project-juice-shop/
- ✓ OWASP WebGoat - https://owasp.org/www-project-webgoat/
- ✓ OWASP Security Shepherd - https://owasp.org/www-project-security-shepherd/
- ✓ GOAD - https://orange-cyberdefense.github.io/GOAD/
- ✓ HackTricks / old mirror - https://angelica.gitbook.io/hacktricks
- ✓ Практическая книга по общему подходу - https://ppn.snovvcra.sh/

# РАСШИРЕННЫЙ ГЛОССАРИЙ, СЛЕНГ И АББРЕВИАТУРЫ

Да, не забыл. Это обязательный блок. Его полезно пролистать перед интервью, чтобы не путаться в терминах и не тормозить на базовых словах.

## 1. Базовые термины и методология

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Pentest** | Penetration Testing | Тестирование на проникновение | Контролируемая имитация атаки на инфраструктуру, сервис или сеть для поиска уязвимостей и ошибок конфигурации. |
| **Ethical Hacking** | - | Этичный хакинг | Поиск и подтверждение уязвимостей с разрешения владельца системы и в рамках согласованных правил. |
| **Red Team** | - | Красная команда | Формат проверки, максимально приближенный к действиям реального противника. |
| **Blue Team** | - | Команда защиты | Инженеры, отвечающие за детект, мониторинг, реагирование и hardening. |
| **Purple Team** | - | Совместная работа Red и Blue | Разбор атак совместно с защитой, чтобы улучшить детект и реагирование. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Scope** | - | Область работ | Что входит в тестирование: адреса, домены, сегменты, окна времени, ограничения. |
| **Rules of Engagement** | RoE | Правила проведения работ | Фиксируют допустимые действия, запреты, окно теста и порядок эскалации. |
| **PTES** | Penetration Testing Execution Standard | Стандарт выполнения пентеста | Фреймворк этапов пентеста: от pre-engagement до reporting. |
| **NIST SP 800-115** | - | Руководство по security testing | Официальная методическая опора по assessment и security testing. |
| **Threat Modeling** | - | Моделирование угроз | Понимание, кто атакует, зачем, через какие активы и какие цепочки наиболее вероятны. |
| **Attack Surface** | - | Поверхность атаки | Все доступные точки входа: сервисы, домены, VPN, web, почта, AD, API. |
| **Kill Chain** | - | Цепочка атаки | Модель этапов атаки: разведка, доставка, эксплуатация, закрепление, движение по сети, impact. |
| **TTPs** | Tactics, Techniques and Procedures | Тактики, техники и процедуры | Практические паттерны действий противника. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **IOC** | Indicator of Compromise | Индикатор компрометации | Артефакт или признак того, что среда уже была скомпрометирована. |
| **IOA** | Indicator of Attack | Индикатор атаки | Поведенческий признак того, что атака идет прямо сейчас. |
| **POC** | Proof of Concept | Подтверждение работоспособности | Демонстрация того, что уязвимость реально эксплуатируется. |
| **Exploit** | - | Эксплойт | Код, техника или цепочка действий для использования уязвимости. |
| **Payload** | - | Полезная нагрузка | То, что запускается после эксплуатации: shell, stager, beacon и т.п. |
| **False Positive** | - | Ложноположительное срабатывание | Инструмент или тест показал проблему, которой по факту нет. |
| **False Negative** | - | Ложноотрицательный результат | Проблема есть, но инструмент или сценарий ее не увидел. |
| **Risk Rating** | - | Оценка риска | Приоритет finding по сочетанию вероятности и влияния на бизнес. |

## 2. Разведка и enumeration

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Recon** | Reconnaissance | Разведка | Сбор открытой и технической информации о цели до активной эксплуатации. |
| **Passive Recon** | - | Пассивная разведка | Сбор данных без прямого контакта с инфраструктурой цели. |
| **Active Recon** | - | Активная разведка | Прямое взаимодействие с целью: ping sweep, port scan, banner grabbing. |
| **OSINT** | Open Source Intelligence | Разведка по открытым источникам | Поиск доменов, e-mail, артефактов, документации, вакансий, GitHub-следов. |
| **WHOIS** | - | Регистрационные данные домена | Может дать регистратор, контакты, NS, даты, технические связи. |
| **DNS Enumeration** | - | Сбор DNS-записей | Поиск A, MX, TXT, CNAME, NS и понимание структуры инфраструктуры. |
| **Subdomain Enumeration** | - | Поиск поддоменов | Один из базовых этапов внешней разведки. |
| **Banner Grabbing** | - | Снятие баннера сервиса | Получение версии ПО, типа сервиса и дополнительных следов для привязки к CVE. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Service Enumeration** | - | Углубленный сбор сведений о сервисе | Не просто увидеть порт, а понять версию, режимы, аутентификацию и слабые настройки. |
| **Host Discovery** | - | Поиск живых хостов | Обнаружение машин и активных узлов в сети. |
| **Port Scanning** | - | Сканирование портов | База для понимания, что вообще доступно и где копать дальше. |
| **TCP Connect Scan** | - | Полное TCP-подключение | Простой, но более шумный вариант сканирования. |
| **SYN Scan** | - | Полуоткрытое сканирование | Классический быстрый вариант в nmap. |
| **UDP Scan** | - | Сканирование UDP | Позволяет увидеть SNMP, DNS, NTP, TFTP и другие сервисы. |
| **Fingerprinting** | - | Определение стека | Выявление ОС, версий сервисов, middleware и особенностей конфигурации. |
| **Attack Path** | - | Путь атаки | Цепочка шагов от первой находки до чувствительного актива. |
| **Initial Foothold** | - | Первичная опорная точка | Первый реальный доступ в инфраструктуру. |

## 3. Сеть и инфраструктура

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Perimeter** | - | Периметр | Публичная граница инфраструктуры: web, VPN, почта, remote access. |
| **Internal Network** | - | Внутренняя сеть | Сегменты, куда попадают после initial access или во внутреннем тесте. |
| **Segment** | - | Сетевой сегмент | Логически или физически выделенная часть сети. |
| **VLAN** | Virtual LAN | Виртуальная LAN | Логическое разделение сети на отдельные сегменты. |
| **DMZ** | Demilitarized Zone | Демилитаризованная зона | Пограничный сегмент между Интернетом и внутренней сетью. |
| **Firewall** | - | Межсетевой экран | Ограничивает сетевой доступ между сегментами или наружу. |
| **ACL** | Access Control List | Список правил доступа | Правила, разрешающие или запрещающие сетевые обращения. |
| **NAT** | Network Address Translation | Преобразование адресов | Сопоставление внутренних и внешних адресов. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Pivoting** | - | Пивотинг | Использование уже скомпрометированного хоста как транзита в другой сегмент. |
| **Lateral Movement** | - | Горизонтальное перемещение | Движение по внутренней сети после первичной компрометации. |
| **Proxying / SOCKS Pivot** | - | Проксирование через хост | Перенаправление своего трафика через уже взятый узел. |
| **Tunneling** | - | Туннелирование | Передача трафика поверх другого канала: SSH tunnel, ligolo, chisel и т.д. |
| **SMB** | Server Message Block | Сетевой файловый протокол Windows | Шары, IPC, аутентификация, signing, relay и масса misconfig-сценариев. |
| **RDP** | Remote Desktop Protocol | Удаленный рабочий стол | Часто точка входа или признак слабой парольной политики. |
| **WinRM** | Windows Remote Management | Удаленное управление Windows | Легитимный канал доступа, который часто используют и пентестеры. |
| **SSH** | Secure Shell | Удаленная консоль | Базовый протокол доступа к Linux/Unix. |
| **FTP** | File Transfer Protocol | Передача файлов | Иногда встречаются анонимный доступ, reuse credentials и старые конфиги. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **TFTP** | Trivial FTP | Упрощенная передача файлов | Протокол без нормальной аутентификации, часто на старых сетевых устройствах. |
| **NFS** | Network File System | Сетевые каталоги Unix/Linux | Неправильный export может открыть путь к данным или эскалации. |
| **RPC** | Remote Procedure Call | Удаленные вызовы процедур | Всплывает при enumeration Windows и старых сервисов. |
| **SNMP** | Simple Network Management Protocol | Протокол управления сетью | Может выдать массу сведений о сетевых устройствах. |
| **LDAP** | Lightweight Directory Access Protocol | Доступ к каталогам | Используется для работы с AD и другими directory services. |
| **LDAPS** | - | LDAP по TLS | Защищенный вариант LDAP. |
| **Kerberos** | - | Протокол аутентификации | Ключевая тема для AD, тикетов, делегирования и злоупотреблений. |
| **NTLM** | - | Семейство механизмов аутентификации Windows | Часто фигурирует в relay и pass-the-hash сценариях. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **DNS Zone Transfer** | - | Передача DNS-зоны | Если настроено небезопасно, выдает структуру домена почти целиком. |
| **ARP Spoofing** | - | Подмена ARP | MITM-техника внутри локального сегмента. |
| **MITM** | Man-in-the-Middle | Атака 'человек посередине' | Перехват или подмена трафика между двумя сторонами. |
| **PCAP** | Packet Capture | Дамп сетевого трафика | Полезен и для анализа, и для подтверждения findings. |

## 4. Windows и Active Directory

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **AD** | Active Directory | Служба каталогов Microsoft | Центральная тема внутреннего инфраструктурного пентеста. |
| **Domain** | - | Домен AD | Область с общими учетками, политиками и доверительными отношениями. |
| **Domain Controller** | - | Контроллер домена | Ключевой сервер, отвечающий за аутентификацию и хранение базы AD. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **OU** | - | Организационная единица | Логическая группировка объектов в AD. |
| **GPO** | - | Групповая политика | Помогает понять устройство среды и часто скрывает ошибки конфигурации. |
| **SID** | - | Идентификатор безопасности | Уникальный ID объекта в Windows. |
| **RID** | Relative Identifier | Относительный идентификатор | Часть SID, определяющая объект внутри домена. |
| **SAM** | Security Account Manager | Локальная база аккаунтов | Содержит локальные учетные записи и хэши. |
| **LSASS** | Local Security Authority Subsystem Service | Процесс хранения security-контекста | В его памяти часто оказываются учетные данные. |
| **Credential Dumping** | - | Извлечение учетных данных | Снятие хэшей, тикетов, паролей из памяти, реестра или файлов. |
| **Pass-the-Hash** | - | Аутентификация по хэшу | Использование NTLM-хэша вместо знания исходного пароля. |
| **Pass-the-Ticket** | - | Аутентификация по тикету | Использование украденного Kerberos ticket. |
| **Overpass-the-Hash** | - | Переход от хэша к Kerberos TGT | Техника получения TGT на основе NTLM-хэша. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Golden Ticket** | - | Поддельный TGT | Один из самых тяжелых сценариев после компрометации krbtgt. |
| **Silver Ticket** | - | Поддельный сервисный билет | Используется против конкретного сервиса. |
| **Kerberoasting** | - | Крекинг сервисных тикетов | Извлечение service tickets и офлайн-крекинг сервисных учеток. |
| **AS-REP Roasting** | - | Крекинг TGT без pre-auth | Атака на учетные записи с отключенной Kerberos pre-authentication. |
| **SPN** | Service Principal Name | Идентификатор сервиса в Kerberos | Часто нужен при Kerberoasting. |
| **Delegation** | - | Делегирование | Механизм передачи права действовать от имени другого субъекта. |
| **Unconstrained Delegation** | - | Неограниченное делегирование | Опасный режим, открывающий удобные векторы атак. |
| **Constrained Delegation** | - | Ограниченное делегирование | Более безопасный режим, но тоже может быть сконфигурирован криво. |
| **RBCD** | Resource-Based Constrained Delegation | Делегирование по ресурсу | Современная тема abuse-сценариев в AD. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **ACL Abuse** | - | Злоупотребление ACL | Эскалация через избыточные права на объекты AD. |
| **DACL** | - | Список прав доступа | Определяет, кто и что может делать с объектом. |
| **ACE** | - | Запись в ACL | Конкретное право для конкретного субъекта. |
| **BloodHound** | - | Анализатор путей атак в AD | Помогает увидеть цепочки от низких прав до Domain Admin. |
| **SharpHound** | - | Сборщик данных для BloodHound | Собирает граф связей и привилегий в AD. |
| **DA** | Domain Admin | Администратор домена | На CTF это конец, в реальной жизни — только начало impact-фазы. |
| **EA** | Enterprise Admin | Администратор леса | Еще более высокий уровень прав в AD. |
| **Local Admin** | - | Локальный администратор | Сильная привилегия на хосте, но не полный контроль над доменом. |
| **Machine Account** | - | Учетная запись компьютера | Компьютерный объект в AD со своим паролем и контекстом. |
| **Trust** | Trust Relationship | Доверительные отношения | Связи между доменами и лесами. |
| **Forest** | - | Лес AD | Набор доменов с общими trust-отношениями. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **NTDS.dit** | - | База Active Directory | Критичный файл контроллера домена с учетными данными и объектами. |
| **SYSVOL** | - | Общий ресурс политик | Там лежат GPO, скрипты и иногда исторические секреты. |
| **cpassword** | - | Устаревшее хранение паролей в GPP | Исторический источник компрометации через SYSVOL. |
| **Password Spraying** | - | Массовая проверка слабого пароля | Один-два популярных пароля против большого числа учеток. |
| **Brute Force** | - | Полный перебор | Лобовой подбор паролей. |
| **Account Lockout Policy** | - | Политика блокировки учеток | Нужно понимать, чтобы не уронить домен неосторожным подбором. |
| **LAPS** | Local Administrator Password Solution | Уникальные локальные пароли | Решение Microsoft против reuse local admin паролей. |
| **ADCS** | Active Directory Certificate Services | Сертификатная инфраструктура AD | Горячая тема современных AD-атак и интервью. |

## 5. Linux / Unix и privilege escalation

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Privilege Escalation** | PrivEsc | Повышение привилегий | Переход от обычного пользователя к root или другому сильному контексту. |
| **Root** | - | Суперпользователь | Максимальные права в Linux/Unix. |
| **SUID** | - | Запуск от имени владельца | Классическая тема Linux PrivEsc. |
| **SGID** | - | Запуск в контексте группы | Может открывать неожиданные варианты эскалации. |
| **Capabilities** | - | Тонкие привилегии Linux | Иногда одна capability заменяет root почти полностью. |
| **sudo** | - | Делегированное выполнение команд | Важно смотреть, какие бинарники и флаги доступны. |
| **sudoers** | - | Конфиг sudo | Ошибки здесь часто ведут к root. |
| **NOPASSWD** | - | sudo без пароля | В сочетании с опасными бинарями часто равно PrivEsc. |
| **Cron / crontab** | - | Планировщик задач | Writable job, небезопасный скрипт или PATH hijack дают эскалацию. |
| **systemd service** | - | Системный сервис | Слабые unit-файлы и права на бинарники открывают путь к PrivEsc. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **PATH Hijacking** | - | Подмена через PATH | Подсовывание своего бинаря вместо ожидаемого. |
| **Writable Directory** | - | Каталог на запись | Часто становится точкой подмены, загрузки payload или persistence. |
| **World-Writable** | - | Запись для всех | Почти всегда красный флаг. |
| **/etc/passwd** | - | Список пользователей | База локальных учетных записей Linux. |
| **/etc/shadow** | - | Хэши паролей | Один из главных трофеев после PrivEsc. |
| **SSH Keys** | - | Ключи SSH | Могут дать тихий доступ без знания пароля. |
| **Bash History** | - | История команд | Часто выдает пароли, токены, адреса и позорные one-liner'ы. |
| **TTY Upgrade** | - | Нормализация shell | Приведение сырой shell-сессии к интерактивному виду. |
| **Reverse Shell** | - | Обратная оболочка | Хост подключается обратно к атакующему. |
| **Bind Shell** | - | Слушающая оболочка | Командная оболочка открывает порт на целевой машине. |
| **Persistence** | - | Закрепление | Сохранение доступа через ключи, сервисы, cron, пользователей и т.д. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Loot** | - | Полезная добыча | Креды, конфиги, токены, документы, ключи, дампы. |

## 6. Уязвимости, misconfig и эксплуатация

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Vulnerability** | - | Уязвимость | Не только CVE, но и ошибка конфигурации, логики или прав. |
| **Misconfiguration** | - | Небезопасная конфигурация | Часто встречается чаще, чем 'магические' 0-day. |
| **CVE** | Common Vulnerabilities and Exposures | Идентификатор публичной уязвимости | Стандартная ссылка на известную проблему. |
| **CVSS** | Common Vulnerability Scoring System | Система оценки тяжести | Базовая метрика severity. |
| **Exploitability** | - | Эксплуатируемость | Насколько реально использовать проблему в конкретной среде. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Exposure** | - | Открытость актива | Насколько проблема доступна атакующему. |
| **Weak Credentials** | - | Слабые учетные данные | Банально, но до сих пор одна из топовых причин компрометации. |
| **Default Credentials** | - | Дефолтные логин/пароль | Особенно часто на сетевых железках и старых системах. |
| **Anonymous Access** | - | Анонимный доступ | Отсутствие аутентификации там, где она должна быть. |
| **Null Session** | - | Анонимная сессия | Классическая тема SMB/RPC enumeration. |
| **Unpatched System** | - | Необновленная система | Отсутствие security-патчей и исправлений. |
| **Legacy Protocols** | - | Устаревшие протоколы | SMBv1, старый TLS, LLMNR/NBNS и другие исторические артефакты. |
| **SMB Signing** | - | Подпись SMB-трафика | Если не требуется, relay становится существенно реальнее. |
| **NTLM Relay** | - | Релей NTLM-аутентификации | Перенаправление чужой аутентификации на другой сервис. |
| **LLMNR / NBT-NS Poisoning** | - | Подмена локального резолва | Часто используется для захвата NTLM-хэшей. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Credential Reuse** | - | Повторное использование учеток | Одна и та же пара логин/пароль на разных системах. |
| **Password Reuse** | - | Повтор пароля | Особенно опасен между workstation, server и админскими учетками. |
| **RCE** | Remote Code Execution | Удаленное выполнение кода | Один из самых опасных классов уязвимостей. |
| **Command Injection** | - | Внедрение команд ОС | Дает удаленное выполнение системных команд. |
| **Deserialization** | - | Небезопасная десериализация | Может приводить к RCE, но требует понимания контекста приложения. |
| **Arbitrary File Read** | - | Чтение произвольных файлов | Часто ведет к ключам, конфигам и секретам. |
| **Arbitrary File Write** | - | Запись произвольных файлов | Может приводить к закреплению или RCE. |
| **Directory Traversal** | - | Выход за пределы каталога | Получение доступа к чужим путям через манипуляцию путями. |
| **Information Disclosure** | - | Раскрытие информации | Версии, логи, конфиги, ключи, дампы, backup-файлы. |

## 7. Учетные данные, сессии и секреты

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Credentials / Creds** | - | Учетные данные | Пароли, хэши, токены, ключи, сертификаты. |
| **Hash** | - | Хэш | Не шифрование, а одностороннее преобразование данных. |
| **Salt** | - | Соль | Дополнительное значение для усложнения офлайн-крекинга. |
| **Token** | - | Токен доступа | Может быть равноценен паролю в рамках конкретного сервиса. |
| **Bearer Token** | - | Предъявительский токен | Кто предъявил, тот и получил доступ. |
| **API Key** | - | Ключ API | Секрет для программного доступа к сервисам. |
| **Secret** | - | Секрет | Общий термин для пароля, ключа, токена, сертификата. |
| **Vault** | - | Хранилище секретов | Система для централизованного безопасного хранения секретов. |
| **Credential Store** | - | Хранилище учетных данных | Менеджеры паролей, секрет-хранилища, локальные stores. |
| **Session** | - | Сессия | Активный контекст аутентифицированного пользователя или сервиса. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Session Hijacking** | - | Захват сессии | Использование чужого session-токена. |
| **Service Account** | - | Сервисная учетка | Часто имеет расширенные права и живет дольше обычных учеток. |
| **Hardcoded Secret** | - | Жестко вшитый секрет | Пароль или токен внутри кода, скрипта или конфигурации. |
| **Plaintext Credentials** | - | Пароли в открытом виде | Снег в июле для пентестера — праздник для атакующего. |
| **Kerberos Ticket** | - | Билет Kerberos | TGT или service ticket, используемый для аутентификации. |
| **TGT** | Ticket Granting Ticket | Билет на выдачу сервисных билетов | Ключевой элемент Kerberos-аутентификации. |
| **TGS** | Ticket Granting Service / service ticket | Сервисный билет | Используется для доступа к конкретному сервису. |

## 8. Логи, артефакты и детект

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Artifact** | - | Артефакт | Любой след активности: лог, файл, процесс, ключ реестра, PCAP. |
| **Evidence** | - | Доказательная база | Скрины, вывод команд, логи, дампы и подтверждения finding'ов. |
| **Telemetry** | - | Телеметрия | Данные, которые собирают EDR, SIEM, логи и системы мониторинга. |
| **Event ID** | - | Идентификатор события | Код события в Windows Event Log. |
| **Syslog** | - | Системное логирование Unix/Linux | Стандартный поток журналов в Unix-мире. |
| **EDR** | Endpoint Detection and Response | Детект и реакция на endpoint | Важный фактор OPSEC и шумности действий. |
| **SIEM** | Security Information and Event Management | Централизованный security-monitoring | Коррелирует события и помогает Blue Team. |
| **Detection Rule** | - | Правило детекта | Условие, по которому появляется alert. |
| **Alert** | - | Оповещение | Сработавший детект или инцидентный сигнал. |
| **OPSEC** | Operational Security | Операционная скрытность | Насколько аккуратно ты работаешь и какие следы оставляешь. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Noisy Technique** | - | Шумная техника | Создает много логов, следов и алертов. |
| **Stealth** | - | Скрытность | Работа так, чтобы не шуметь без реальной необходимости. |

## 9. Отчетность и findings

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Finding** | - | Выявленная проблема | Не просто 'дырка', а оформленный результат с impact, evidence и remediation. |
| **Executive Summary** | - | Резюме для менеджмента | Пишется понятным языком, без лишней технички. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Technical Summary** | - | Техническое резюме | Краткий обзор для инженеров и ИБ-команды. |
| **Impact** | - | Влияние | Что реально может произойти для бизнеса и инфраструктуры. |
| **Likelihood** | - | Вероятность эксплуатации | Насколько практично и реалистично использовать finding. |
| **Remediation** | - | Исправление | Что нужно сделать, чтобы закрыть проблему. |
| **Mitigation** | - | Смягчение риска | Компенсирующие меры, если полное исправление пока невозможно. |
| **Compensating Controls** | - | Компенсирующие меры защиты | Дополнительные меры, уменьшающие риск при наличии проблемы. |
| **Reproduction Steps** | - | Шаги воспроизведения | Должны быть понятными, но не превращаться в боевой гайд. |
| **Affected Asset** | - | Затронутый актив | Конкретный хост, сервис, сегмент, учетка или бизнес-система. |
| **Proof / Validation** | - | Подтверждение | Проверка, что finding реален и воспроизводим. |
| **Severity** | - | Уровень серьезности | Обычно Critical / High / Medium / Low / Info. |
| **Business Risk** | - | Риск для бизнеса | Техническая проблема в терминах потерь, простоя, утечки и |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| | | | репутации. |
| **Recommendations** | - | Рекомендации | Практические действия по исправлению и предотвращению повторения. |

## 10. Практический жаргон и инструменты

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Энум** | Enumeration | Сленговое сокращение | Углубленный сбор техсведений; часто именно 'энум' решает весь кейс. |
| **Залететь в сеть** | - | Получить initial access | Первый реальный вход в целевую среду. |
| **Зацепка** | - | Мелкий, но важный след | То, из чего затем вырастает attack path. |
| **Поднять шелл** | - | Получить shell | Открыть командную оболочку на целевой системе. |
| **Рутануть** | - | Получить root | Получить максимальные права в Unix/Linux. |
| **Пошуметь** | - | Оставить много следов | Создать заметный объем событий и алертов. |
| **Снять креды** | - | Извлечь credentials | Пароли, хэши, токены, тикеты и все, что может дать следующий шаг. |
| **Слить данные** | - | Вывести или получить чувствительные данные | Данные клиентов, документы, бэкапы, базы, секреты. |
| **Шарить шару** | - | Проверять сетевые шары | Поиск доступных SMB/NFS-ресурсов и прав на них. |
| **Брутить** | - | Подбирать пароль | Brute force или словарный подбор. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Спрейить** | - | Делать password spraying | Проверять несколько типовых паролей на большой группе пользователей. |
| **Пивотиться** | - | Делать pivoting | Ходить через уже взятый узел в новые сегменты. |
| **Дойти до DA** | - | Получить Domain Admin | Серьезная стадия компрометации, но не финальная цель в реальной жизни. |
| **Боевой кейс** | - | Реалистичный сценарий | Задача, приближенная к реальной среде, а не только к лабе. |
| **Nmap** | - | Сканер сети и сервисов | База для discovery, scanning и service enumeration. |
| **NetExec / CrackMapExec** | - | Массовая работа по Windows-среде | SMB/LDAP/WinRM enumeration и abuse. |
| **Mimikatz** | - | Работа с credentials и Kerberos | Классический инструмент post-exploitation в Windows. |
| **Responder** | - | Poisoning и захват хэшей | LLMNR/NBNS/MDNS-сценарии в локальном сегменте. |
| **Impacket** | - | Набор Python-инструментов | Работа с протоколами и Windows/AD abuse-сценариями. |
| **Rubeus** | - | Kerberos-операции в Windows | Тикеты, roasting, delegation, abuse. |

| Термин | Полное название / аббревиатура | Перевод / значение | Практический смысл |
|---|---|---|---|
| **Ligolo-ng** | - | Туннели и pivoting | Удобный современный инструмент для выхода во внутренние сегменты. |
| **Chisel** | - | Туннель поверх HTTP/WebSocket | Популярен для reverse/forward tunneling. |
| **Enum4linux / enum4linux-ng** | - | SMB/Windows enumeration | Удобен на ранних этапах enumeration. |
| **Wireshark** | - | Анализатор трафика | Наглядный разбор сетевых сессий и PCAP. |
| **tcpdump** | - | Консольный сниффер | Быстрый захват и анализ трафика без GUI. |
| **LinPEAS / WinPEAS** | - | Подсказчики PrivEsc | Автоматизированный сбор артефактов для поиска путей эскалации. |
| **Metasploit** | - | Фреймворк эксплуатации | Нужно знать, но на интервью важно не выглядеть человеком одной кнопки. |

# !!!!ВНИМАНИЕ!!!

Это **демо-версия** полноценного учебного издания по подготовке к интервью и практике в penetration testing.

**В полной версии:**

- **170+ страниц** прикладного материала;
- еще **40 теоретических вопросов**;
- еще **12 практических real-word кейсов**;
- **детальный roadmap по самостоятельному обучению** с разбивкой по неделям и темам;
- **личные рекомендации автора** по развитию навыков;
- дополнительные **sample reports**, приложения и чек-листы;
- расширенные рекомендации по проведению pentest-а и подготовке к собеседованиям (репо в GitHub).

Эта версия нужна, чтобы показать общий стиль, глубину и практическую ценность данного материала. Free version ©

За **полной версией** и персональной консультацией - к автору: **Ivan Piskunov | white2hack**.

# WHITE2HACK | CYBERSECURITY SINCE 2018

• **Кибербезопасность простыми словами:** ликбез, практические советы, разборы кейсов

• **Книги, гайды, how-to:** подборки материалов и прикладные инструкции, которые реально можно внедрять

• **Аналитика и тренды индустрии:** что происходит в security-мире и куда всё движется, отчеты, оценки, прогнозы

• **Карьера и комьюнити:** развитие навыков, ориентиры по профессиям, события/эвенты

• **Этичный хакинг + защита данных:** приватность, цифровая гигиена, ИБ осознанность

**Telegram: www.w2hack.t.me | @white2hack**