Network: Часть 8. Безопасность

24.10.2026 10:04 edited 24.10.2026 10:50

Эпоха утраченного доверия

В сентябре 1986 года бухгалтерия Лоуренс-Берклийской национальной лаборатории обнаружила в компьютерных счетах расхождение на семьдесят пять центов: некто потреблял машинное время, не платя за него. Системный администратор Клиффорд Столл, вместо того чтобы списать копейки на ошибку округления, потратил десять месяцев на слежку за взломщиком: писал собственные программы перехвата, распечатывал журналы телетайпа и дежурил ночами, пока не выяснил, что из лаборатории, связанной с Министерством энергетики США, данные уходят в Западную Германию — а оттуда в КГБ. Взломщиком оказался хакер Маркус Хесс, книга Столла «Яйцо кукушки» (1989) стала первым бестселлером о компьютерных преступлениях. Семьдесят пять центов обернулись первым публичным доказательством того, что сеть, построенная на доверии, уже кем-то используется как оружие. Сорок лет спустя сетевая безопасность выглядит так. По оценке Cybersecurity Ventures, киберпреступность обходится миру примерно в 10,5 триллиона долларов в год — и, по их же прогнозу, к 2031 году эта цифра вырастет до 12,2 триллиона, средняя стоимость утечки данных достигла рекордных 4,99 миллиона долларов, прибавив 12% за год (IBM, 2026), эксплуатация уязвимостей впервые стала главным способом проникновения — 31% взломов, — оттеснив кражу учетных данных, а человеческий фактор присутствует в 62% инцидентов (Verizon DBIR, 2026), крупнейшая отраженная DDoS-атака достигла 31,4 терабита в секунду (Cloudflare, четвертый квартал 2025-го), свыше 95% загрузок страниц в Chrome идут по шифрованному HTTPS, а удостоверяющий центр Let's Encrypt выпускает около десяти миллионов сертификатов в сутки, в мире используется порядка пяти миллиардов аппаратных passkey-ключей (FIDO Alliance, май 2026-го), и больше половины веб-трафика уже защищено гибридным постквантовым обменом ключами — за два года доля выросла с двух процентов.

Парадокс сетевой безопасности в том, что ее в Интернете никогда не было. Протоколы TCP/IP проектировались в начале 1980-х сообществом из нескольких тысяч исследователей, которые знали друг друга в лицо: IP не проверяет подлинность отправителя, DNS не подписывает ответы, ARP верит любому, TCP предсказуем в своих номерах, а телнет и FTP передают пароли открытым текстом. Безопасность не закладывали не из наивности — из экономии: каждое усложнение стоило бы памяти и процессорного времени, которых у тогдашних машин не было. Когда сеть вышла за пределы доверенного круга университетов и лабораторий, дыры пришлось закрывать на ходу — слоями криптографии, наросшими поверх протоколов, как броня поверх скелета. За сорок лет эта гонка «атака — защита — новая атака» породила целую индустрию: TLS

зашифровал почти весь веб-трафик, Let's Encrypt раздал сертификаты сотням миллионов сайтов, аппаратные ключи и passkey вытесняют пароли, а постквантовая криптография начала внедряться раньше, чем появились квантовые компьютеры, способные ее сломать. И тем не менее принцип не изменился: безопасность — это цепочка, прочная ровно настолько, насколько прочно ее самое слабое звено. Идеальная криптография не спасает от пароля 123456, а миллиардные SOC не спасают от звонка в службу поддержки.

Эта ситуация порождает ряд критических вопросов:

1. **Вопрос о ключах.** Как две стороны, никогда не видевшие друг друга, договариваются об общем секрете по каналу, который прослушивается? Ответ, найденный Диффи и Хеллманом в 1976 году, казался современной магией: общий ключ возникает из обмена публичными параметрами так, что подслушивающий не может его вычислить. Почему это открытие называют поворотным моментом в истории криптографии — и как оно связано с рассекреченными лишь в 1997 году работами британской спецслужбы GCHQ?
2. **Вопрос о доверии.** Откуда браузер знает, что сертификат банка подлинный? Ответ: ему так сказала одна из нескольких сотен корневых удостоверяющих центров, встроенных в операционные системы. Централизованная фабрика доверия держит весь Интернет на честном слове — и история знает, что случается, когда слово нарушают: взлом DigiNotar в 2011 году позволил перехватывать переписку сотен тысяч пользователей Gmail в Иране, а сама компания обанкротилась за считаные недели.
3. **Вопрос о человеке.** Криптографию не ломают в лоб — ее обходят. В 62% инцидентов присутствует человек: фишинг, телефонная социальная инженерия, усталость от MFA, пароли, утекшие из чужих сервисов. Почему атакующим выгоднее позвонить в службу поддержки, чем взламывать шифр, — и что индустрия противопоставляет этому, от passkey до обучения?
4. **Архитектурный вопрос.** Где должна жить защита: в Wi-Fi (WPA3), на сетевом уровне (IPsec), на транспортном и прикладном (TLS)? Чем ниже уровень, тем прозрачнее защита и тем меньше она знает о смысле данных, чем выше — тем гибче, но тем больше мест, где ее забыли включить. Почему Интернет выбрал TLS, хотя IPsec старше и «правильнее», — и почему сегодня безопасность все равно сквозная?
5. **Вопрос гонки.** Каждое поколение защиты порождает новое поколение атак: BEAST, CRIME, POODLE, FREAK, Logjam, Heartbleed — названия уязвимостей TLS читаются как список серийных убийц. Асимметрия безжалостна: защитник обязан закрыть все, атакующему достаточно одной дыры. Почему Log4Shell 2021 года назвали самой массовой уязвимостью десятилетия — и чему научила индустрию история с закладкой в архиваторе xz в 2024-м?
6. **Вопрос о метаданных.** Шифрование скрывает содержимое, но не факт связи: кто, с кем, когда и сколько. Анализ трафика переживает TLS — а Tor, микс-сети и сквозное шифрование мессенджеров пытаются закрыть и этот зазор. Почему спор о «going dark» — доступе спецслужб к переписке — идет тридцать лет, от Clipper Chip до британского Online Safety Act, и почему криптографы каждый раз выигрывают технически, но не всегда политически?

7. **Вопрос о времени.** Перехваченный сегодня зашифрованный трафик можно расшифровать завтра: противники уже сейчас коллекционируют данные в расчете на будущие квантовые компьютеры (harvest now, decrypt later). В августе 2024 года NIST утвердил первые постквантовые стандарты, а браузеры и CDN развернули гибридный обмен ключами раньше, чем квантовая машина взломала хотя бы один RSA-ключ. Почему миграция измеряется десятилетиями и что такое криптографическая гибкость?

Дальше мы пройдем всю сетевую безопасность: от рождения отрасли и целей защиты — через симметричную и асимметричную криптографию, хеши и подписи, инфраструктуру открытых ключей, аутентификацию от паролей до passkey — к TLS, IPsec и VPN, защите Wi-Fi, разведке и атакам «человек посередине», DDoS, вредоносным программам и вымогателям, атакам на цепочку поставок, файерволам и нулевому доверию, веб-безопасности, приватности, экономике и человеческому фактору — и закончим квантовым будущим криптографии. В финале проследим путь одного вторжения — от фишингового письма до отчета об инциденте — и увидим, как все разобранные механизмы складываются в единую оборону. Материал опирается на стандарты IETF и NIST, документированные инциденты и отраслевую статистику 2025–2026 годов.

Рождение безопасности: от всеобщего доверия к обороне

Сеть, в которой все свои

Первые пятнадцать лет Интернет не нуждался в безопасности по той же причине, по которой в ней не нуждался клуб профессоров: все участники были известны друг другу, а злоупотребление означало бы изгнание из сообщества. Сеть описывали как открытое сообщество единомышленников: участников были тысячи, они знали друг друга в лицо, электронная почта не подписывалась, удаленный вход через телнет передавал пароль первым же пакетом, а протоколы маршрутизации верили любому объявлению. Первым предупреждением стал случай Столла в 1986–1987 годах: одиночка, а не спецслужба, мог месяцами читать чужую почту и документы, оставаясь незамеченным, потому что сама идея «мониторинга вторжений» еще не существовала. Столл, по сути, изобрел ее на практике — журнал событий, перехват трафика, ловушку-приманку.

2 ноября 1988 года аспирантом Корнелла Робертом Таппаном Моррисом был запущен червь, задуманный как безобидный эксперимент по измерению размера Интернета. Программа содержала ошибку: вместо одного экземпляра на машину она размножалась повторно и забивала ресурсы. Из примерно шестидесяти тысяч подключенных к ARPANET компьютеров червь поразил около шести тысяч — каждую десятую машину в мире. Ущерб оценивали от ста до девяноста шести миллионов долларов, Моррис получил три года условно, десять с половиной тысяч долларов штрафа и четыреста часов общественных работ, а впоследствии стал профессором MIT и сооснователем Y Combinator. Главным последствием червя стало другое: в том же ноябре 1988-го DARPA профинансировала создание CERT/CC — группы реагирования на компьютерные инциденты при Университете Карнеги — Меллон. Так у Интернета появилась профессия защитника.

От хулиганов к индустрии

Девяностые превратили безопасность из академической экзотики в индустрию. В 1994-м Netscape выпустила SSL — первый массовый протокол шифрования для веба, в 1995-м Тату Илонен после перехвата паролей в сети своего университета за одну неделю написал SSH, а в 1997-м NIST объявил открытый конкурс на замену DES — так началась публичная эра криптографических стандартов. В феврале 2000-го пятнадцатилетний канадец Майкл Калс по кличке Mafiaboy обрушил DDoS-атаками Yahoo, eBay, CNN, Amazon и Dell — суммарный ущерб оценили в миллиарды, а слушания в Конгрессе США впервые поставили вопрос о защите коммерческого Интернета. Май того же года — вирус ILOVEYOU филиппинца Онеля де Гусмана поразил десятки миллионов машин за сутки, де Гусмана не осудили лишь потому, что на Филиппинах не было закона о компьютерных преступлениях, — его приняли через два месяца после эпидемии.

Начало 2000-х показало скорость, с которой сеть стала уязвимой. 25 января 2003 года червь Slammer — UDP-пакет из 376 байтов, бивший по порту 1434 серверов Microsoft SQL, — удваивал число зараженных каждые 8,5 секунды и поразил 90% уязвимых машин планеты за десять минут. Заплатка существовала шесть месяцев, большинство жертв просто не поставили ее. MyDoom в январе 2004-го стал самым быстрым почтовым червем в истории — на пике каждое двенадцатое письмо в мире несло его копию, а совокупный ущерб с поправкой на инфляцию оценивают почти в 38 миллиардов долларов. Эпоха «детских шалостей» заканчивалась: на смену ей приходили организованная киберпреступность, корпоративный шпионаж и государственные операции — от Stuxnet (2010) до NotPetya (2017) и SolarWinds (2020), которые разберем дальше. Ключевой урок первых двух десятилетий: безопасность — это не состояние, а процесс, потому что противник адаптируется быстрее, чем обновляются системы.

Цели и модели: что значит защищенная система

Четыре свойства и пятая нога

Классическая формула целей безопасности — триада CIA плюс неотказуемость. **Конфиденциальность** (confidentiality): данные читает только тот, кому они предназначены, — ее обеспечивают шифрование (TLS, VPN) и контроль доступа. **Целостность** (integrity): данные не изменены при передаче или хранении, — ее обеспечивают хеши, HMAC и цифровые подписи. **Доступность** (availability): сервис работает, когда нужен, — против нее направлены DDoS-атаки, вымогатели и даже собственные ошибки конфигурации. **Аутентификация** (authentication): уверенность в том, с кем имеешь дело, — пароли, сертификаты, аппаратные ключи. И **неотказуемость** (non-repudiation): отправитель не может позже заявить «я этого не делал», — ее дает цифровая подпись, имеющая в ряде юрисдикций юридическую силу. На практике свойства переплетены: TLS обеспечивает сразу конфиденциальность, целостность и аутентификацию сервера, а доступность остается на совести других механизмов.

Принцип Керкгоффса и слабое звено

В 1883 году голландский криптограф Огюст Керкгоффс сформулировал принцип, ставший фундаментом всей современной криптографии: стойкость системы должна опираться исключительно на секретность ключа, а не на секретность алгоритма. Алгоритм все равно узнают — украдут, восстановят по реализации, опубликуют, как случилось с рассекреченными в 1997 году разработками GCHQ. Отсюда современная норма: AES, RSA, SHA и TLS полностью открыты, миллионы специалистов их изучают, и именно публичность делает их надежными — безопасность через неясность (security through obscurity) не работает, потому что «неясное» рано или поздно становится ясным. Второй фундаментальный принцип: безопасность — цепочка. HTTPS на сайте не спасает, если пароль администратора «123456», идеальное шифрование бесполезно, если ключ сгенерирован слабым датчиком случайных чисел или лежит в коде приложения. Атакующий не выбирает самое крепкое звено — он ищет самое слабое, и защита обязана учитывать все звенья сразу: людей, процессы, конфигурации, протоколы и математику.

Модель атакующего и цепочка убийства

Профессиональная защита начинается не с инструментов, а с модели противника: кто атакует (скрипт-кидди, организованная группа, инсайдер, государственная спецслужба), какими ресурсами обладает и какова его цель — кража, разрушение, шпионаж, вымогательство. От ответа зависит бюджет и архитектура: от банковского трояна защитит одно, от противника уровня Stuxnet — совсем другое. Практической моделью атаки служит kill chain — последовательность шагов вторжения, формализованная Lockheed Martin в 2011 году и развитая в матрице MITRE ATT&CK (2013):

```
разведка → точка входа → закрепление → боковое перемещение → достижение цели → вывод данных
```

Разведка собирает поверхность атаки (attack surface): IP-диапазоны, DNS-записи, открытые порты, версии ПО, сотрудников в соцсетях. Точка входа — уязвимый сервис, инъекция, фишинг. Закрепление — бэкдор, новый пользователь, планировщик задач. Затем движение внутри сети от веб-сервера к базе и к домену, и наконец цель: данные зашифрованы, украдены или уничтожены. Ценность модели в том, что защите не обязательно выигрывать каждый этап: достаточно порвать цепочку в любом месте — и чем больше независимых барьеров на пути, тем дороже атака. Отсюда принцип эшелонированной обороны (defense in depth): файервол, сегментация, шифрование, MFA, мониторинг и обучение не дублируют друг друга зря — они рассчитаны на то, что предыдущий слой уже провален.

Где живет защита в стеке уровней

Безопасность не принадлежит одному уровню модели: на канальном уровне работает WPA3 для Wi-Fi, на сетевом — IPsec и фильтрация пакетов, на транспортном и прикладном — TLS, выше — пароли, токены, подписи транзакций. Выбор уровня — компромисс: защита ниже прозрачнее (приложениям не нужно ничего знать) и универсальнее, но грубее,

защита выше понимает смысл данных (кого аутентифицировать, что подписывать), но каждое приложение должно внедрить ее само — и обязательно найдется то, которое забудет. История распорядилась так, что Интернет сделал ставку на прикладной TLS, а не на сетевой IPsec: последний оказался слишком сложен для массового внедрения и прижился в VPN-туннелях, тогда как TLS зашифровал почти весь трафик браузеров, почты и API. Современный ответ на вопрос «где делать безопасность» — везде сразу, потому что сквозная система защищает не уровень, а путь данных от источника до потребителя.

Симметричная криптография: один ключ на двоих

От Шеннона до DES

Научную основу шифрования заложил Клод Шеннон в секретном докладе 1945 года, опубликованном в 1949-м как «Теория связи в секретных системах»: он доказал, что абсолютно стойкий шифр существует — это одноразовый блокнот Вернама, где ключ длиннее сообщения и никогда не используется дважды, — и что все остальные шифры стойки лишь настолько, насколько трудно их взломать. Абсолютная стойкость оказалась непрактичной: ключ нужно передать так же безопасно, как и само сообщение. Весь двадцатый век криптография решала одну задачу — сделать ключи короткими и переиспользуемыми, не потеряв стойкость.

Первым массовым ответом стал DES. В начале 1970-х IBM разработала шифр Lucifer под руководством Хорста Файстеля, в 1973 году Национальное бюро стандартов США объявило открытый конкурс, и в ноябре 1976-го усовершенствованный Lucifer утвердили как федеральный стандарт с длиной ключа 56 бит. История DES — это и первая криптографическая драма: АНБ настояло на укорачивании ключа и на модификации S-блоков, что породило двадцатилетние подозрения в закладке. Подозрения оказались напрасными — наоборот, в 1990-м Эли Бихам и Ади Шамир опубликовали дифференциальный криптоанализ, а IBM признала, что знал о нем еще в 1974-м: правки АНБ сделали шифр устойчивее. Но 56 бит было мало. В 1997 году энтузиасты проекта DESCHALL под руководством Роке Версера взломали ключ распределенными вычислениями тысяч добровольцев за 96 дней, в 1998-м EFF построила специализированную машину Deep Crack за 250 тысяч долларов — она нашла ключ за 56 часов, в январе 1999-го в паре с distributed.net — за 22 часа 15 минут. Стандарт, защищавший банковские транзакции двадцать лет, публично умер. В 2005 году DES официально исключили из списка одобренных стандартов.

AES и режимы работы

На смену NIST устроил первый полностью открытый криптографический конкурс: в 1997 году приняли заявки от всего мира, в 1999-м отобрали пять финалистов (MARS от IBM, RC6 от RSA Security, Rijndael, Serpent, Twofish), а 2 октября 2000 года победителем объявили бельгийский Rijndael Джоан Димена и Винсента Реймена. В ноябре 2001-го он стал стандартом AES (FIPS 197): блоки по 128 бит, ключи 128, 192 или 256 бит, никаких секретных S-блоков и никакого участия спецслужб в дизайне. AES выдержал четверть

века атак: практического взлома полного числа раундов нет до сих пор, а с 2010 года инструкции AES-NI в процессорах Intel сделали шифрование почти бесплатным — гигабайты в секунду на ядро.

Один и тот же шифр можно применять по-разному, и режимы работы оказались опаснее самого алгоритма. Режим ECB шифрует каждый блок независимо: одинаковые блоки дают одинаковый шифротекст, и знаменитый Tux-пингвин на иллюстрации режима ECB остается узнаваемым силуэтом — ECB запрещен для всего, кроме учебных примеров. Режим CBC сцепляет блоки и требует уникального вектора инициализации, повтор IV или предсказуемый IV рождают атаки вроде BEAST. Хуже того, CBC с проверкой отступов (padding) уязвим к padding oracle: если сервер разными ответами или разным временем отклика выдает, корректен ли padding, атакующий расшифровывает сообщение побайтово, отправляя модифицированные блоки, — так работали атаки Lucky13 (2013) и POODLE (2014), похоронившая SSL 3.0. Современный ответ — режим GCM (Galois/Counter Mode): он совмещает шифрование с аутентификацией (AEAD), выдает метку целостности и встроен в каждый набор шифров TLS 1.3. Урок режима GCM прост: нельзя сначала шифровать, а потом «дошивать» целостность отдельным HMAC — нужно использовать конструкции, где они неразделимы.

Асимметричная криптография: ключ, который можно публиковать

«Новые направления в криптографии»

Проблема распределения ключей казалась нерешаемой: чтобы обмениваться шифротекстом, стороны должны заранее, по защищенному каналу, передать друг другу общий ключ. В 1974 году Ральф Меркль, студент Беркли, придумал «головоломки Меркля» — задачу, которую инициатору решить легко, а подслушивающему в квадрат раз дороже, — и послал работу на рецензию, Николас Негопропулос ее отклонил. В 1976-м Уитфилд Диффи и Мартин Хеллман опубликовали статью «Новые направления в криптографии» с первым практическим протоколом обмена ключами: стороны обмениваются степенями общего основания по открытому каналу, каждая возводит полученное число в свою секретную степень — и независимо приходят к одному и тому же общему секрету. Подслушивающий видит лишь два публичных числа, чтобы получить секрет, ему нужно решить задачу дискретного логарифмирования, экспоненциально сложную. Диффи и Хеллман предложили и саму концепцию цифровой подписи. В 2015-м они получили премию Тьюринга, Меркля в списках не оказалось, но его приоритет общепризнан. А в 1997 году британское правительство рассекретило материалы GCHQ: Клиффорд Кокс придумал схему с открытым ключом еще в 1973-м — за четыре года до открытой публикации RSA. Спецслужба знала ответ раньше, но не могла им воспользоваться: вычислительной мощности не хватало.

RSA и факторизация

В 1977 году Рон Ривест и Ади Шамир из MIT вместе с Леонардом Адлеманом построили первую практическую систему с открытым ключом — RSA. В ее основе другая трудная задача: умножить два больших простых числа легко, а разложить произведение обратно на множители (факторизовать) — практически невозможно. Из пары простых чисел получается приватный ключ, их произведение входит в публичный. Тот же механизм работает и как подпись: «зашифровать» хеш документа приватным ключом может только владелец, а проверить — любой. В честь шифра учредили задачу RSA-129: за разложение 129-значного числа обещали 100 долларов. Считалось, что на это уйдут тысячи лет, в апреле 1994 года шестьсот добровольцев из двадцати стран справились за восемь месяцев, передав фразу «The Magic Words are Squeamish Ossifrage». С тех пор рекорды двигались методично: RSA-768 (232 десятичных цифры) разложили в декабре 2009 года за эквивалент двух тысяч процессорных лет, в феврале 2020-го пал и RSA-250 — число из 250 знаков. Практический вывод: ключи RSA короче 2048 бит считаются ненадежными, 3072 бит — минимальная рекомендация NIST для новых систем, а для сравнимой стойкости на эллиптических кривых (ECC, независимо предложенных Нилом Коблицем и Виктором Миллером в 1985-м, кривая Curve25519 Даниэля Бернстайна, 2006) хватает 256 бит. ECC экономит ключи и время — именно эллиптические кривые (ECDHE) обеспечивают обмен ключами в современном TLS.

Гибридная схема и качество случайности

Асимметричная криптография медленнее симметричной в сотни и тысячи раз, поэтому в реальности ее почти никогда не используют для шифрования данных. Работает гибрид: асимметричный протокол (ECDHE или RSA) лишь договаривается о симметричном сеансовом ключе, а дальше терабайты трафика шифрует быстрый AES-GCM или ChaCha20-Poly1305. Именно так устроен TLS, так же устроены IPsec и SSH: математическая дорогая часть — один раз на рукопожатие, дешевая — на весь сеанс.

Фундамент всей асимметрии — случайность: ключи, векторы инициализации, nonce должны быть непредсказуемыми, и генератор случайных чисел становится самым слабым звеном. В 2008 году выяснилось, что из-за ошибки в пакете Debian OpenSSL два года выдавал предсказуемые ключи — сотни тысяч серверов пришлось перевыпускать сертификаты. В 2010-м Sony использовала одну и ту же константу вместо случайного числа в подписи прошивки PlayStation 3 — защита консоли рухнула за неделю. В 2017-м уязвимость ROCA в библиотеке Infineon породила сотни тысяч RSA-ключей, восстанавливаемых из публичной части, — дефект нашли в национальных ID-картах Эстонии. Вывод отраслевой нормы: криптография ломается не математикой, а реализацией — и потому реализации проверяют аудитом, а случайность берут из аппаратных генераторов.

Хеши, HMAC и цифровые подписи

Хеш-функции: отпечаток данных

Криптографический хеш превращает сообщение произвольной длины в короткий фиксированный отпечаток, и делает это так, что по отпечатку нельзя восстановить исходник (стойкость к прообразу), нельзя найти два разных сообщения с одним отпечатком (стойкость к коллизиям), а изменение одного бита входа меняет примерно половину битов выхода. Хеши — универсальный клей безопасности: на них держатся цифровые подписи, хранение паролей, контроль целостности загрузок, блокчейны и Certificate Transparency.

История хешей — хроника падений. MD4 и MD5 Рональда Ривеста (1990 и 1992) казались незыблемыми, пока в 2004 году китайский математик Ван Сяоюнь не продемонстрировала построение коллизий MD5 за минуты: два разных документа с одинаковым отпечатком. Практическое применение нашли быстро — в 2012-м червь Flame использовал коллизию MD5, чтобы подделать подпись Microsoft и распространяться через Windows Update. SHA-1 (1995) продержался дольше: 23 февраля 2017 года исследователи CWI Амстердам и Google объявили SHAttered — первую реальную коллизию, стоившую эквивалента шести с половиной тысяч процессорных лет, браузеры к тому моменту уже отключали сертификаты с SHA-1, а PDF-подписи на SHA-1 потеряли силу. Сегодняшняя норма — SHA-256 из семейства SHA-2 и его преемник SHA-3 (стандарт FIPS 202, август 2015-го, построенный на конструкции Keccak, не родственной SHA-2, — страховка на случай, если и SHA-2 когда-нибудь падет). Для паролей обычные хеши не годятся: они слишком быстрые, и миллиарды подстановок в секунду на GPU перебирают словарные пароли мгновенно. Поэтому пароли хешируют медленными адаптивными функциями с солью — bcrypt (Провос и Мазьер, 1999), scrypt (2009), а сегодня предпочтителен Argon2, победитель Password Hashing Competition 2015 года (RFC 9106).

HMAC и подпись

Хеш без ключа доказывает лишь то, что данные не изменились случайно, — подделать новый корректный хеш злоумышленнику тривиально. HMAC (RFC 2104, 1997) добавляет в конструкцию секретный ключ: код аутентичности может вычислить и проверить только его знающий. HMAC применяется в IPsec, в подписи JWT-токенов, в протоколах аутентификации и внутри TLS 1.2 как код целостности записей.

Цифровая подпись соединяет хеш с асимметричной криптографией: отправитель вычисляет хеш документа и «подписывает» его приватным ключом, получатель проверяет подпись публичным ключом и сверяет хеш. Подпись решает две задачи — аутентификацию отправителя и целостность документа, — а также обеспечивает неотказуемость: приватный ключ был только у подписанта, и отказаться от авторства он не может. Подписями защищены обновления программного обеспечения, документы PDF,

сообщения почты (DKIM), транзакции и сертификаты PKI. Те же эллиптические кривые сделали подписи компактными: Ed25519 подписывает и проверяет в разы быстрее RSA при 128-битном уровне стойкости.

PKI: фабрика доверия

Сертификаты и цепочки

Асимметричная криптография решает вопрос «как передать ключ», но не решает вопрос «чей это ключ». Любой может опубликовать публичный ключ и назвать его ключом банка. Ответ — инфраструктура открытых ключей (PKI): удостоверение личности для ключей. Стандарт X.509 (ITU-T, 1988, интернет-профиль RFC 5280) описывает сертификат: публичный ключ владельца, его доменное имя, срок действия, издатель и подпись издателя. Издателем выступает удостоверяющий центр (CA), чей собственный сертификат подписан вышестоящим центром, а корни — сами собой (самоподписанные) и предустановлены в операционных системах и браузерах. Проверка проста: браузер получает сертификат сайта, строит цепочку до доверенного корня, сверяет подписи на каждом шаге, срок действия и совпадение доменного имени. В мире несколько сотен доверенных корневых CA — от гигантов вроде DigiCert, Sectigo и Google Trust Services до государственного удостоверяющего центра России, Let's Encrypt выдает сертификаты бесплатно и автоматически.

Слабость системы очевидна: доверие транзитивно, и любой из сотен центров может выдать сертификат на чужой домен — браузер его примет. История подтверждала это многократно. В марте 2011-го через итальянского реселлера Comodo были выпущены подложные сертификаты для google.com, yahoo.com и Skype. Летом 2011-го голландский CA DigiNotar был взломан: злоумышленники выпустили сертификаты для доменов Google и использовали их для перехвата переписки примерно трехсот тысяч иранских пользователей Gmail, компания обанкротилась через два месяца. В 2017-м Google объявил о поэтапном лишении доверия сертификатов Symantec после того, как сотрудник выпустил тестовые сертификаты для google.com, — Symantec продал бизнес удостоверяющего центра DigiCert за 950 миллионов долларов. Реакцией индустрии стали Certificate Transparency (RFC 6962, 2013): публичные журналы, куда CA обязаны вносить каждый выпущенный сертификат, — подложный выпуск теперь видно всему миру, записи CAA в DNS (RFC 8659), разрешающие домену указывать список уполномоченных CA, и переход к короткоживущим сертификатам — в 2025 году CA/Browser Forum проголосовал за сокращение максимального срока жизни сертификата до 200 дней с марта 2026-го, до 100 дней с 2027-го и до 47 дней с 2029-го: украденный ключ станет бесполезен за недели.

Отзыв, pinning и автоматизация доверия

Отдельная проблема — отзыв: сертификат скомпрометирован до истечения срока, и его нужно объявить недействительным. Списки отозванных сертификатов (CRL) громоздки, протокол OCSP заставляет браузер спрашивать у CA в реальном времени, раскрывая ему историю просмотров и замедляя соединения, на практике браузеры проверяют отзыв

непоследовательно, а OCSP stapling перекладывает задачу на сервер. Полноценного решения нет — отрасль ставит на короткий срок действия как на главный механизм отзыва. Дополняет картину certificate pinning: приложение заранее «прибивает» ожидаемый ключ, как делали мобильные банки, стандарт HPKP для веба провалился (слишком легко «выстрелить себе в ногу») и был удален из браузеров к 2019 году.

Let's Encrypt изменил экономику доверия радикально. Запущенный в декабре 2015 года Internet Security Research Group центр автоматизировал выдачу через протокол ACME: сертификат запрашивается и продлевается скриптом, без писем и оплаты. Результат: десять лет спустя центр выпускает порядка десяти миллионов сертификатов в сутки, его сертификат используют почти две трети всех веб-сайтов (W3Techs, октябрь 2026-го), а доля HTTPS-загрузок в Chrome превысила 95%. В январе 2026-го Let's Encrypt добавил шестидневные сертификаты — шаг в будущее, где ключи живут дни, а доверие полностью автоматизировано. Альтернативная модель — web of trust PGP, где пользователи подписывают ключи друг другу без единых центров, — технически элегантна, но осталась нишевой: людям удобнее делегировать доверие институту, пусть и несовершенному, чем строить его самостоятельно.

Крипто-войны: государство против математики

PKI выросла в разгар крипто-войн — тридцатилетней борьбы государств за контроль над шифрованием. В 1993 году АНБ предложило Clipper Chip: шифр с аппаратным escrow-ключом для прослушки по решению суда, в 1994-м Мэтт Блейз показал, что механизм контроля ломается, и к 1996-му проект тихо умер. До 2000 года сильная криптография считалась в США боеприпасом: экспорт PGP был уголовным преступлением, и Фил Циммерман, опубликовав его в 1991-м, три года провел под следствием — после чего издал исходный код книгой в MIT Press, формально использовав конституционную защиту печати. Даниэль Бернстайн в 1995-м подал иск, доказывая, что криптографический код — это речь, в 1999-м апелляционный суд с ним согласился, и экспортные правила либерализовали. Проиграв войну за контроль, государства перешли к войне за бэкдоры — спор о «going dark» продолжается до сих пор, и каждый раунд (Apple против ФБР в 2016-м, британский Online Safety Act, австралийский закон 2018 года) криптографы выигрывают по одному аргументу: бэкдор для «хороших» нельзя сделать недоступным «плохим».

Аутентификация: пароли, билеты и аппаратные ключи

Пароль: старейшее и слабейшее звено

Пароль придумали вместе с разделением доступа к компьютерам в начале 1960-х в MIT (система CTSS), и шестьдесят лет спустя он остается главным способом войти почти везде — и главной причиной взломов. Статистика безжалостна: в списке NordPass 2025 года первое место снова занял «123456», за ним «admin» и «12345678», такие пароли инструменты подбирают менее чем за секунду. Хуже того, пароли переиспользуют: утекший из одного сервиса пароль открывает десятки других. Проект Have I Been Pwned

Троя Ханта накопил миллиарды записей из утечек — одна только «Коллекция № 1» 2019 года содержала 773 миллиона адресов почты и 21 миллион паролей, — и сервис Pwned Passwords позволяет проверить пароль, не раскрывая его (протокол k-анонимности).

Хранить пароли открытым текстом нельзя, быстрыми хешами — тоже: GPU перебирают миллиарды кандидатов в секунду, а радужные таблицы находят совпадения мгновенно. Правильная практика — медленный адаптивный хеш с индивидуальной солью на пользователя (bcrypt, Argon2), ограничение частоты попыток входа и проверка пароля по базам утечек. Передавать пароль по сети в открытом виде тоже нельзя — отсюда challenge-response: сервер присылает случайное число (nonce), клиент доказывает знание пароля вычислением ответа, сам пароль по каналу не идет, а nonce делает бессмысленным повтор перехваченного ответа.

Kerberos: билеты вместо паролей

В 1980-х в проекте Athena Массачусетского технологического института родилась идея централизованной аутентификации без передачи паролей по сети — Kerberos, названный в честь трехголового пса Цербера. Схема с центром распределения ключей (KDC), разбитым на сервер аутентификации и сервер выдачи билетов, работает так: пользователь один раз доказывает знание пароля и получает билет-билетодатель (TGT), а затем по TGT получает сервисные билеты для конкретных служб — каждый зашифрован ключом своей службы и снабжен меткой времени против повторов. Пароль вводится один раз за сессию, службы доверяют KDC, повторы отсекаются. Протокол пятой версии (RFC 1510, 1993, современный профиль RFC 4120, 2005) стал основой аутентификации в Windows 2000 и остается ядром доменов Active Directory до сих пор — каждая корпоративная сеть Windows фактически сеть Kerberos. Его же и атакуют по-крупному: «золотой билет» (компрометация главного ключа домена дает власть над всем лесом), kerberoasting (перебор хешей сервисных билетов), атаки с понижением версии.

Многофакторность и passkey

Когда пароль все равно крадут — фишингом, подбором по утечкам, перехватом SMS, — добавляет силу второй фактор: то, что у пользователя есть (телефон, токен), или то, чем он является (биометрия). Эффективность измеряется жестко: по оценке Microsoft, MFA блокирует более 99,9% автоматических атак на учетные записи. Классы второго фактора неравноценны: SMS перехватывается SIM-свопингом, push-уведомления выматываются MFA fatigue (так взяли Uber в 2022-м), а аппаратные ключи устойчивы и к фишингу, потому что подпись привязана к конкретному домену.

Именно на этом принципе построены стандарты FIDO2/WebAuthn — рекомендация W3C с 4 марта 2019 года. Приватный ключ создается и хранится в защищенном элементе устройства и никогда его не покидает, сервер хранит лишь публичный ключ, вход — это подписанная криптографическая задача, привязанная к домену, а не переиспользуемый секрет. Passkey, анонсированные Apple, Google и Microsoft в мае 2022-го, — та же технология с синхронизацией через облако: ключ живет в менеджере паролей платформы и переносится между устройствами. Внедрение идет стремительно: к маю 2026-го FIDO

Alliance оценивал число используемых passkey в пять миллиардов, осведомленность о технологии — в 90%, а долю успешных входов через passkey — примерно в 36% против 26% годом ранее. Для машинного доступа парадигма иная: OAuth 2.0 (RFC 6749, 2012) делегирует права без передачи пароля, OpenID Connect добавляет слой идентификации, а JWT-токены (RFC 7519, 2015) несут подписанные утверждения — при условии, что разработчик проверяет алгоритм и подпись, а не принимает их на веру: ошибки проверки JWT — классика взломов API.

Идея, объединяющая OAuth, JWT и API-токены, старше их всех: capability-based security. Вместо вопроса «кто ты?» система спрашивает «что тебе разрешено?» — и носителем разрешения становится сам токен, неотчуждаемый и непеределываемый билет на конкретное действие. Владелец ссылки на документ в облаке не «авторизуется» — он предъявляет capability, сервису не нужно знать его личность. Отсюда и сила, и слабость подхода: утекший токен дает ровно те права, которые в него зашили, — поэтому токены делают короткоживущими, узкоскоупными и отзываемыми, а модель минимальных привилегий из корпоративных сетей переезжает в API.

TLS: протокол, который зашифровал Интернет

От SSL к TLS 1.3

В 1994 году команда Netscape во главе с Тахером Эль-Гамалем спроектировала SSL — «розетку безопасности» между прикладным и транспортным уровнями, чтобы платежи и пароли в раннем вебе не ходили открытым текстом. SSL 2.0 (1995) оказался дырявым, SSL 3.0 (1996) переработали Алан Фрейер, Пол Кохер и Фил Карлтон, а в 1999-м IETF стандартизовал его наследника — TLS 1.0 (RFC 2246). Дальше шли версии: TLS 1.1 (2006), TLS 1.2 (2008, RFC 5246) — на долгие годы основной протокол Интернета, — и TLS 1.3 (RFC 8446, август 2018), радикальная чистка: рукопожатие ускорили до одного round-trip (и 0-RTT для повторных сессий), вырезали все устаревшие алгоритмы (RSA-обмен ключами, CBC, RC4, SHA-1, все версии ниже 1.2), оставили только AEAD-шифры и сделали perfect forward secrecy обязательной через (EC)DHE. TLS 1.3 шифрует даже часть рукопожатия, включая сертификат сервера, — подслушивающий не видит, какой сайт вы открываете, кроме имени в SNI (его закрывает развивающийся стандарт ECH).

Рукопожатие за один шаг

Современный TLS 1.3 выглядит так. Клиент в ClientHello сразу посылает список поддерживаемых шифров, случайное число и — что главное — свои параметры ECDHE (key share), не дожидаясь ответа сервера. Сервер в ServerHello выбирает шифр, присылает свою половину ECDHE, свой сертификат X.509 и подпись параметров обмена своим приватным ключом. Обе стороны независимо вычисляют общий секрет (ECDHE) и из него — сеансовые ключи AES-GCM или ChaCha20-Poly1305. Клиент проверяет цепочку сертификатов до доверенного корня, имя домена и срок действия, обмен Finished подтверждает, что обе стороны видят одинаковое рукопожатие, — любое вмешательство посередине здесь ломается. Дальше прикладные данные идут записями TLS: каждая

запись шифруется, снабжается скрытым счетчиком последовательности (защита от повторов) и меткой аутентификации. Вся схема — наглядная демонстрация гибридной криптографии: дорогая асимметрика один раз, дешевая симметрика на весь трафик.

Perfect forward secrecy (PFS) заслуживает отдельного слова. Без нее сеансовый ключ шифровался публичным ключом сервера: укради приватный ключ сервера — и весь записанный за годы трафик расшифровывается задним числом. С ECDHE сеансовый секрет каждый раз рождается заново и не зависит от долгоживущего ключа сервера: компрометация ключа не вскрывает прошлые сессии. Именно поэтому в TLS 1.3 PFS обязательна, а в 1.2 ее приходилось включать осознанно.

Хроника падений и уроков

TLS как протокол ломают редко — ломают окружение и старые версии. BEAST (сентябрь 2011, Дуонг и Риццо) эксплуатировала предсказуемые IV в CBC TLS 1.0, CRIME (2012) восстанавливала cookies по изменению длины сжатого трафика, POODLE (октябрь 2014, Google) добила SSL 3.0 через padding oracle, FREAK и Logjam (март и май 2015) заставили серверы понижать соединение до экспортных 512-битных параметров Диффи — Хеллмана, наследие крипто-войн: Logjam работала против 8,4% миллиона крупнейших сайтов. Самая знаменитая — Heartbleed (объявлена 7 апреля 2014-го, CVE-2014-0160): ошибка в реализации heartbeat-расширения OpenSSL позволяла читать до 64 килобайтов памяти сервера — приватные ключи, пароли, сессии. Баг жил в коде два года, уязвимыми оказались около полумиллиона серверов (примерно 17% SSL-хостов), а нашли его независимо Нил Мехта из Google и финская Codenomicon. Heartbleed стал символом главной проблемы отрасли: критическая инфраструктура Интернета держалась на проекте с мизерным бюджетом и горсткой разработчиков, после него OpenSSL получил финансирование и аудит, а индустрия усвоила цену «невидимых» зависимостей.

Не менее поучительны ошибки конфигурации. RC4 долго считался спасением от BEAST, пока атаки 2013 года не привели к запрету стандартом RFC 7465 (2015). Массовый отключение старых версий тянулось годами. Клиентская сторона ломается проще: одна строчка verify=False в коде отключает проверку сертификата — и MITM с любым самоподписанным сертификатом читает весь трафик, исследования раз за разом находили такой паттерн в заметной доле мобильных приложений. Другой вектор — trust store: если в систему установлен чужой корневой CA (корпоративный прокси, малварь или рекламный модуль Superfish, который Lenovo в 2015 году предустанавливала на свои ноутбуки), любые поддельные сертификаты принимаются молча. Ответами стали HSTS (RFC 6797, 2012) — заголовок, запрещающий браузеру говорить с сайтом по HTTP и убивающий TLS stripping, — и постепенное выдавливание HTTP из веба.

Состояние на 2026 год

Цифры фиксируют победу шифрования по умолчанию: свыше 95% загрузок страниц в Chrome идут по HTTPS (в развитых странах до 99%), около 87% всех веб-сайтов имеют валидный сертификат (W3Techs, октябрь 2026-го), TLS 1.3 поддерживают примерно три четверти крупнейших сайтов, а к осени 2026-го Chrome сделал режим «всегда

защищенное соединение» включенным по умолчанию. Let's Encrypt выпускает около десяти миллионов сертификатов в сутки, HTTP/3 перенес TLS 1.3 внутрь QUIC поверх UDP, соединив шифрование с транспортом. За тридцать лет TLS прошел путь от «экзотики для банков» до невидимого стандарта: пользователь 2026 года, открывая незащищенный HTTP-сайт, получает предупреждение браузера — полная инверсия ситуации 1990-х, когда шифрование было исключением.

IPsec и VPN: защита на уровне IP

Стандарт, опередивший внедрение

Пока прикладной уровень только примерялся к шифрованию, IETF готовил защиту на самом сетевом уровне. В августе 1995 года вышел первый набор RFC по IPsec (1825–1829), в 2005-м его переработали (RFC 4301 и соседи). Идея амбициозна: безопасность встроена в сам IP, прозрачна для приложений и работает «из коробки» для всего трафика. IPsec состоит из двух протоколов: AH (Authentication Header, протокол 51) подтверждает целостность и подлинность пакета, но не шифрует, ESP (Encapsulating Security Payload, протокол 50) шифрует полезную нагрузку и аутентифицирует ее. Два режима: транспортный — защищается содержимое пакета между конечными узлами, туннельный — целиком исходный пакет упаковывается в новый, и так строятся VPN. Ключами управляет IKE (Internet Key Exchange, современная версия IKEv2 — RFC 7296): по сути, тот же обмен Диффи — Хеллмана плюс сертификаты, что делает настройку IPsec отдельной инженерной дисциплиной.

IPv6 требовал IPsec в обязательном порядке, и это требование позже отменили как невыполненное: стандарт оказался слишком гибким — десятки опций, режимов и алгоритмов породили несовместимые реализации и репутацию «протокола, который невозможно настроить дважды одинаково». Массовый Интернет выбрал TLS, а IPsec занял свою нишу, где он незаменим: site-to-site VPN между офисами и дата-центрами, защита магистральных каналов, корпоративный удаленный доступ.

От PPTP к WireGuard

История VPN для людей — история последовательного отказа от сломанного. PPTP (1996, консорциум Microsoft) позволял офисным клиентам «звонить» в корпоративную сеть через Интернет, но к 1998-му Брюс Шнайер с коллегами показали фатальные дефекты его аутентификации и шифрования, тем не менее PPTP дожил в настройках роутеров до наших дней, и отключать его рекомендуют до сих пор. L2TP/IPsec заменил криптослой, OpenVPN (Джеймс Йонан, 2001) добавил гибкости на пользовательском TLS-подобном протоколе и стал де-факто стандартом удаленного доступа на двадцать лет.

В 2015–2016 годах Джейсон Доненфелд представил WireGuard — полную противоположность IPsec по философии: около четырех тысяч строк кода вместо сотен тысяч, фиксированный современный набор алгоритмов (Curve25519 для ключей, ChaCha20-Poly1305 для шифрования, BLAKE2 для хешей), никаких опций, которые можно

неправильно настроить. Линус Торвальдс назвал его «произведением искусства» на фоне OpenVPN и IPsec, а в марте 2020-го WireGuard включили в ядро Linux. Сегодня это стандартный выбор для новых VPN — от корпоративных mesh-сетей вроде Tailscale до серверов потребительских провайдеров.

Отдельная реальность — потребительские VPN-сервисы, обещающие анонимность. Технически они дают шифрование «последней мили» до сервера провайдера и подмену видимого IP-адреса, но весь трафик после этого идет открыто к провайдеру VPN, который видит больше, чем домашний оператор, — исследования раз за разом находили у массовых VPN-приложений утечки DNS, встроенные трекеры и журналы, несмотря на политику «no logs». VPN защищает от слежки в кафе и от провайдера, но не от самого VPN-провайдера: доверие не исчезает, оно переезжает.

Zero trust: VPN уходит со сцены

Классический VPN воплощает устаревшую модель «замка со рвом»: попал внутрь туннеля — ты доверенный и видишь всю сеть. Атака на одного удаленного сотрудника или украденные VPN-учетки (так начинались многие громкие взломы, включая Colonial Pipeline) превращают «ров» в парадный вход. Ответ — архитектура нулевого доверия: доступ дается не «в сеть», а к конкретному приложению, каждый запрос аутентифицируется и авторизуется заново, с проверкой состояния устройства, — VPN исчезает как концепция. Об этом подробнее в разделе о защите сети.

Беспроводные сети: от WEP к WPA3

WEP: сломанный до рождения

Стандарт Wi-Fi 802.11 (1997) обещал «эквивалентную проводной сети безопасность» — WEP, Wired Equivalent Privacy. Обещание не выполнилось ни в одной букве: WEP использовал потоковый шифр RC4 со статическим ключом и 24-битным вектором инициализации, который повторялся за часы на загруженной сети. В августе 2001 года Скотт Флурер, Шай Мантин и Ади Шамир опубликовали атаку FMS, эксплуатирующую слабые ключи RC4: по достаточному количеству пакетов ключ восстанавливался статистически. К середине 2000-х инструменты вроде aircrack-ng автоматизировали взлом до минут, а в 2005-м ФБР публично взломало WEP за три минуты. Миллионы сетей остались беззащитны: обновить прошивку старых точек доступа было нечем.

Промежуточным решением стал WPA (2003) с протоколом TKIP — «заплатка» поверх того же железа, полноценной защитой стал WPA2, утвержденный в стандарте 802.11i в июне 2004 года: AES в режиме CCMP и честное рукопожатие с генерацией свежих ключей (4-way handshake). WPA2 держал оборону тринадцать лет, став синонимом «защищенного Wi-Fi».

KRACK и WPA3

16 октября 2017 года Мэйти Ванхов из KU Leuven представил KRACK (Key Reinstallation Attack): ошибка оказалась не в шифре, а в самом рукопожатии WPA2 — при потере пакета атакующий мог заставить клиента переустановить уже использованный ключ шифрования, обнулить nonce и читать трафик. Уязвимы были практически все реализации стандарта — от Windows и Linux до iPhone и Android, мир увидел, что сертифицированный, проверенный годами протокол может содержать дефект в ядре. Патчи разошлись за месяцы, но урок остался: формальная верификация протоколов — не роскошь (позже рукопожатие WPA3 проверяли именно формальными методами, и все же нашли обходные пути).

WPA3, объявленный Wi-Fi Alliance 25 июня 2018 года, заменил парольное рукопожатие протоколом SAE (Simultaneous Authentication of Equals, «Dragonfly»): словарь перебора по перехваченному трафику невозможен, каждая сессия получает уникальные ключи (forward secrecy для Wi-Fi), добавлен защищенный режим для открытых сетей. И почти сразу — апрель 2019-го — Ванхов вместе с Эйялом Роненом представили Dragonblood: атаки понижения до WPA2, утечки по побочным каналам и восстановление пароля в режиме перехода. Ошибки исправили, стандарт дозрел, но мораль той же природы, что и с TLS: уязвимости живут не в математике, а в протокольных состояниях и реализациях. К 2026 году WPA3 обязателен для сертификации новых устройств Wi-Fi Alliance, однако в парке техники господствует смешанный режим WPA2/WPA3, а корпоративные сети переходят на WPA3-Enterprise с 802.1X и RADIUS. И отдельная вечная угроза беспроводного мира — «злой двойник» (evil twin): открытая точка доступа с именем любимого кафе, которая видит весь незашифрованный трафик, против нее работает только повсеместный TLS и VPN, но не сам Wi-Fi.

Атаки: разведка, перехват и человек посередине

Поверхность атаки и инструменты разведки

Любая атака начинается с разведки: злоумышленник изучает поверхность атаки — все, что система показывает миру. Whois и DNS рассказывают о доменах и инфраструктуре, журналы Certificate Transparency — о поддоменах, поисковики — о забытых тестовых стендах. Сканер портов nmap, написанный Гордоном Лайоном (Fyodor) и впервые опубликованный осенью 1997 года в журнале Phrack, опрашивает хосты и по ответам восстанавливает открытые порты и версии служб — а версия Apache или nginx тут же сверяется с базами уязвимостей. Поисковик Shodan (Джон Матерли, 2009) пошел дальше: он непрерывно сканирует весь Интернет и индексирует сами ответы устройств — камеры, роутеры, промышленные контроллеры с паролями по умолчанию доступны любому желающему без всякого сканирования. Защита от разведки частична: минимизация открытых портов, скрытие версий, геоблокировка админок, — но полностью спрятать сервер в публичном Интернете невозможно, и модель безопасности обязана исходить из того, что атакующий знает топологию.

Перехват трафика: от хаба к ARP-спуфингу

Проще всего читать чужой трафик там, где среда общая. В эпоху концентраторов (hub) каждый пакет видели все порты, и сниффер — программа в promiscuous-режиме — собирал пароли телнета, FTP и SMTP напрямую, анализатор Ethereal (Джеральд Комбс, 1998), ставший в 2006-м Wireshark, родился именно из этой практики. Коммутируемые сети (switch) посылают кадр только получателю — но протокол ARP, переводящий IP в MAC-адреса, по-прежнему верит любому ответу без аутентификации. ARP-спуфинг: атакующий рассылает поддельные ARP-ответы «IP шлюза — это мой MAC», и машины жертв направляют весь внешний трафик через него. Так же работают подмена DHCP и rogue-роутеры. Внутри IPv6 ту же роль играют сообщения neighbor discovery. Стивен Белловин описал эти и десятки других фундаментальных дыр TCP/IP еще в 1989 году в статье «Проблемы безопасности в наборе протоколов TCP/IP» — задолго до того, как они стали массовым инструментом.

Классика перехвата — угон TCP-сессии. В Рождество 1994 года Кевин Митник использовал предсказуемость номеров последовательностей TCP, чтобы подделать пакеты от имени доверенного хоста и войти в компьютер Цутому Симуры — охота за Митником кончилась его арестом 15 февраля 1995-го и книгой «Takedown» (2002). Сегодня номера секвенсов криптостойки, но сама идея живет в RST-инъекциях: подделанный пакет с правильными номерами рвет соединение — прием, который годами применяли системы цензуры, включая Великий файрвол, для разрыва «нежелательных» соединений.

DNS-спуфинг и его лечение

Слабейшее звено именования — отсутствие аутентификации ответов DNS. Подменив ответ резолвера, атакующий отправляет жертву на поддельный сервер банка вместо настоящего. В июле 2008 года Дэн Каминский показал, что отравление кэша гораздо практичнее, чем считалось: подобрав 16-битный идентификатор транзаккции birthday-атакой за тысячи запросов, можно подменить запись домена целиком — включая адреса корневых серверов. Индустрия экстренно выпустила согласованный патч (рандомизация исходного порта) на специальной встрече в штаб-квартире Microsoft, а Каминский отказался публиковать детали полгода. Долгосрочным ответом стал DNSSEC (RFC 4033–4035, 2005): подписи зон и цепочка доверия от корня (подписан в июле 2010-го) к доменам (.com — в 2011-м), минус — подписи не защищают от подглядывания, и развертывание растянулось: валидацию в 2026 году проходят десятки процентов мировых резолверов. Приватность закрыли другие механизмы: DNS-over-TLS (RFC 7858, 2016) и DNS-over-HTTPS (RFC 8484, 2018), которые Firefox и Chrome включают по умолчанию, — запросы к сайтам больше не ходят открытым текстом к провайдеру.

Человек посередине против TLS

MITM — универсальная схема: атакующий вклинивается между клиентом и сервером, читая и модифицируя трафик, а каждая сторона уверена, что говорит с другой напрямую. Против голого HTTP работают все перечисленные техники перехвата. Против TLS классический MITM невозможен без одного из трех условий: жертва принимает

поддельный сертификат (установленный чужой корневой CA, отключенная проверка, обманутый пользователь), либо сервер подписывает злоумышленнику настоящий сертификат (скомпрометированный CA — случай DigiNotar), либо используется TLS stripping. Последнее — изобретение Мокси Марлинспайка 2009 года (sslstrip): прокси принимает от клиента обычный HTTP-запрос, сам идет к сайту по HTTPS и отдает жертве незашифрованный ответ, понизив соединение до того, как HSTS успел сработать. Защитный комплекс отработан: HSTS (включая preload-список браузеров), строгая проверка сертификатов, Certificate Transparency, делающая подложный выпуск видимым, и pinning в мобильных приложениях. В 2026 году MITM против TLS — это уже не массовая техника, а целевая операция против доверия: либо социальная, либо государственная, либо через скомпрометированную PKI.

Отдельный класс атак не требует ни перехвата, ни подделки — повтор (replay). Злоумышленник, записавший легитимный зашифрованный пакет, может отправить его снова: открыть дверь чужим ключом-пропуском, повторно списать деньги, проголосовать дважды. Противоядие одно и то же во всех протоколах: свежесть. Одноразовые случайные числа (nonce), метки времени с допустимым окном, счетчики последовательности — все, что делает записанный пакет непригодным для повторного использования. TLS нумерует записи скрытым счетчиком и рвет соединение при перестановке, Kerberos снабжает билеты метками времени, IPsec ведет анти-replay окно номеров, в веб-приложениях ту же роль играют CSRF-токены и одноразовые коды платежей. Правило проектирования протокола: любой шаг, который можно повторить с выгодой, обязан содержать элемент свежести.

DoS и DDoS: война на истощение

Механика отказа

Цель DoS — не украсть, а сломать: сделать сервис недоступным. Самый древний прием — SYN flood: атакующий шлет тысячи SYN-запросов с подставных адресов, сервер резервирует память под полусоединения, очередь переполняется, и легитимные клиенты не проходят. Лекарство придумал Даниэль Бернстайн в 1996-м — SYN cookies: состояние соединения кодируется криптографически в самом номере последовательности SYN-ACK, сервер ничего не хранит до финального ACK. UDP- и ICMP-флуды бьют по пропускной способности, smurf использовал широковещательные ICMP-эхо с подставным адресом жертвы — пока их не запретили. Особый класс — отражение и усиление: маленький запрос с поддельным адресом жертвы вызывает огромный ответ третьей стороны. Коэффициент усиления DNS-запроса — десятки, NTP-команды monlist — 556, а memcached в 2018-м дал рекордные 51 000:1 — именно так 28 февраля 2018 года GitHub пережил атаку в 1,35 Тбит/с, а через день другой жертве досталось 1,7 Тбит/с. L7-атаки изощреннее: HTTP-флуд выглядит как обычные запросы, Slowloris (Роберт Хансен, 2009) держит соединения открытыми, медленно дописывая заголовки, CVE-2023-44487 «HTTP/2

Rapid Reset» (октябрь 2023-го) использовал мгновенную отмену потоков — так Google в августе 2023-го отбил атаку в 398 миллионов запросов в секунду, десятикратно превышавшую прежние рекорды.

От Mirai к терабитной эре

DDoS — распределенная версия, бьющая из тысяч машин сразу. Индустриализацию обеспечили ботнеты из IoT: в сентябре 2016-го журналист Брайан Кребс пережил 620 Гбит/с от Mirai — червя, перебиравшего стандартные пароли веб-камер и роутеров, 21 октября 2016-го Mirai ударил по DNS-провайдеру Dyn, и пол-Америки на часы потеряло Twitter, Reddit, Netflix и CNN. Авторы — трое студентов из Рутгерса — в конце сентября 2016-го выложили исходники червя на хакерском форуме, и с тех пор Mirai-клоны стали commodity. Хронология рекордов говорит сама за себя: 300 Гбит/с против Spamhaus (март 2013-го, DNS-усиление), 1,7 Тбит/с (memcached, 2018), 5,6 Тбит/с (Cloudflare, октябрь 2024-го), 7,3 Тбит/с (май 2025-го), 29,7 Тбит/с (сентябрь 2025-го, ботнет Aisuru), и, наконец, 31,4 Тбит/с в четвертом квартале 2025-го — атака ботнета Kimwolf, собранного из более чем 1,8 миллиона зараженных Android-устройств, в основном ТВ-приставок без обновлений безопасности, в те же недели Cloudflare отбивал и HTTP-атаки свыше 200 миллионов запросов в секунду. Во втором квартале 2026-го атак мощнее 1 Тбит/с было уже более восьмисот — на 519% больше, чем годом ранее: терабитный трафик стал рутиной.

Защита и экономика

Защита строится на распределении: anycast-сети вроде Cloudflare, Akamai и Google рассеивают атаку по сотням точек присутствия, где трафик проходит через скраббинг — автоматическую фильтрацию аномалий, fingerprinting ботнетов и rate limiting. Для L7-атак добавляются WAF, проверки на ботов и поведенческий анализ. Обратная сторона — экономика: арендовать ботнет для часовой атаки на black market стоит десятки долларов, и «stresser-сервисы» продают DDoS как услугу, ФБР и Европол регулярными операциями (Operation Power Off) закрывают крупнейшие площадки, но их место занимают новые. DDoS остается самой дешевой и доступной цифровой дубиной — от вымогательства у малого бизнеса до геополитики: на пике напряженности 2022–2026 годов атаки на правительственные сайты и банки стали фоновым шумом конфликтов. Полностью проблема не решается никогда: источник трафика — чужие зараженные устройства, а жертва обязана быть доступной. Асимметрия в чистом виде: стоимость атаки падает, стоимость фильтрации растет.

Вредоносные программы: от Creeper до вымогателей

Тридцать лет эволюции

Первой самореплицирующейся сетевой программой стал Creeper (1971, Боб Томас, BBN): он перепрыгивал между мейнфреймами DEC в ARPANET, выводя на телетайп «I'm the creeper, catch me if you can!», — и против него написали Reaper, первую антивирусную программу. Первый вирус для IBM PC, Brain (1986), сочинили пакистанские братья Алви,

чтобы защитить свой медицинский софт от пиратства. Эпидемии 2000–2004 годов — ILOVEYOU, Code Red (июль 2001-го, 359 тысяч зараженных серверов IIS за четырнадцать часов), Nimda, Slammer, MyDoom — показали скорость распространения в полностью связном мире, Conficker (2008–2009) заразил, по разным оценкам, до пятнадцати миллионов машин и заставил Microsoft объявить награду в 250 тысяч долларов и собрать рабочую группу из конкурентов.

Stuxnet (обнаружен в июне 2010-го) перевел вредоносные программы в другую весовую категорию: червь нес четыре zero-day эксплойта Windows, два — к ПО Siemens, три украденных цифровых подписи производителей драйверов и нацеливался на контроллеры центрифуг иранского завода по обогащению урана в Натанзе. Около тысячи центрифуг вышли из строя, операцию приписывают США и Израилю (программа «Олимпийские игры»). Впервые код пересек границу между битами и физическим миром, а подписанный украденными сертификатами червь наглядно показал уязвимость PKI.

Эра вымогателей

CryptoLocker (сентябрь 2013-го) изобрел бизнес-модель: шифрование файлов жертвы с требованием выкупа в биткоинах. Последователей у CryptoLocker хватало, но настоящая эра началась 12 мая 2017 года с WannaCry: червь использовал эксплойт EternalBlue к SMB — инструмент АНБ, украденный и опубликованный группой Shadow Brokers месяцем ранее, — и за день поразил 230 тысяч компьютеров в 150 странах. Пострадали Deutsche Bahn, Telefonica, Renault и британская NHS: около восьмидесяти из 236 госпитальных трестов, тысячи отмененных операций, убыток порядка 92 миллионов фунтов, заплатка Microsoft вышла еще в марте, но ее не поставили. Распространение остановил 22-летний исследователь Маркус Хатчинс, зарегистрировав за 10,69 доллара домен-«выключатель», зашитый в червь. Атрибуция — Lazarus, Северная Корея (официально — США, декабрь 2017-го).

Через шесть недель NotPetya (27 июня 2017-го) замаскировался под вымогателя, будучи чистой машиной разрушения: проникнув через обновление украинской бухгалтерской программы M.E.Doc, он за часы парализовал Maersk (4 тысячи серверов, 45 тысяч компьютеров), Merck, FedEx и Чернобыльскую АЭС, где дозиметрию пришлось мерить вручную. Совокупный ущерб оценивают свыше 10 миллиардов долларов — самая дорогая кибератака в истории, США и союзники публично возложили ответственность на российскую военную разведку, а в октябре 2020-го Минюст США предъявил обвинения шести офицерам подразделения 74455.

К 2020-м вымогатели превратились в корпоративную индустрию с моделью RaaS (ransomware-as-a-service), двойным вымогательством (шифрование плюс угроза публикации украденного) и аффилиатной схемой, где разработчик делит выручку с исполнителями. Colonial Pipeline (7 мая 2021-го, группа DarkSide): остановка 5500-мильного трубопровода, дающего 45% топлива Восточному побережью США, паника на заправках, выкуп 4,4 миллиона долларов биткоинами — из которых ФБР вернуло 63,7 BTC, получив доступ к кошельку. Change Healthcare (февраль 2024-го, BlackCat/ALPHV): вход через Citrix-

портал без MFA парализовал аптечные расчеты по всем США на недели, UnitedHealth заплатил 22 миллиона выкупа, затронуто до 190 миллионов американцев, совокупные потери корпорации превысили 3 миллиарда долларов. MOVEit (май — июнь 2023-го, Cl0p): zero-day SQL-инъекции в файлообменнике дал 2700 организациям и свыше 90 миллионов человек в пострадавших. В 2025-м волна прошла по британской рознице: Marks & Spencer потеряла около 300 миллионов фунтов прибыли после социальной инженерии против сторонней службы поддержки (аффилиаты Scattered Spider), сеть Co-op осталась без части поставок, в августе 2025-го остановился на пять недель Jaguar Land Rover — экономисты оценили эффект для экономики Великобритании до 1,9 миллиарда фунтов, в сентябре атака на ПО Collins Aerospace MUSE сорвала регистрацию пассажиров в Хитроу, Брюсселе и Берлине. Статистика фиксирует индустриализацию: по данным Black Kite, в 2025 году на сайтах утечек вымогателей засветились 7551 организация — на четверть больше, чем годом ранее, Verizon DBIR 2026 помещает вымогательские схемы почти в половину всех инцидентов, а Sophos насчитывает, что шифрование удается примерно в 56% атак — защита научилась срывать чаще, но нападающие перешли на чистое вымогательство с кражей данных, без шифрования вовсе.

Цепочка поставок: доверие как вектор

Самые эффективные атаки последнего десятилетия били не в жертву, а в ее поставщиков. SolarWinds (обнаружен FireEye 8 декабря 2020-го): в обновления платформы Orion, которой пользовались 18 тысяч организаций, включая девять федеральных агентств США, был вшит троян SUNBURST с «периодом сна», операцию приписали российской СВР. Log4Shell (декабрь 2021-го, CVE-2021-44228, оценка CVSS 10,0): крошечная библиотека журналирования Log4j, встроенная в сотни миллионов устройств — от Minecraft до банковского ПО, — выполняла удаленный код из-за JNDI-подстановки в строке лога, уязвимость сообщил Apache 24 ноября исследователь Alibaba Cloud Чэнь Чжаоцзюнь, эксплуатация началась еще до публичного анонса, Tenable назвала ее самой масштабной уязвимостью десятилетия. И xz-utils (29 марта 2024-го, CVE-2024-3094): инженер Microsoft Андрес Фройнд, расследуя полусекундную задержку SSH-входа, обнаружил закладку в архиваторе liblzma, которую два года внедрял авторизовавшийся в сообществе персонаж «Jia Tan» с сетью подставных аккаунтов, давивших на утомленного мейнтейнера. Закладка давала удаленный доступ к sshd на тестовых сборках Fedora и Debian — ее остановили в шаге от стабильных релизов практически всей Linux-инфраструктуры Интернета. Урок всех трех историй одинаков: в мире, где любая программа собирается из тысяч чужих компонентов, доверие к зависимости — это тоже вектор атаки, а репозитории, обновления и CI/CD требуют той же защиты, что и периметр. Отсюда и новая категория OWASP Top 10 2025 — Software Supply Chain Failures.

Защита сети: периметр, сегментация и нулевое доверие

Три поколения файерволов

Первое поколение — пакетные фильтры конца 1980-х: маршрутизаторы, сверяющие каждый пакет с таблицей правил (адреса, порты, протоколы). Работали быстро, но не помнили контекст: пакет, принадлежащий установленному соединению, отличался от нового только флагами TCP, и правила быстро превращались в нечитаемую простыню. Систематизировали практику Билл Чесвик и Стив Белловин в AT&T — их книга «Файерволы и безопасность Интернета» (1994) стала первым учебником отрасли. Второе поколение добавило состояние (stateful inspection): DEC SEAL Маркуса Рэнума (1992) и особенно Check Point FW-1 Гиля Шведа (1993, патент на stateful-фильтрацию) отслеживали таблицу соединений — «ответный трафик разрешен автоматически» стало нормой, и файервол перестал быть фильтром, став диспетчером сессий. Третье поколение — application-level gateway и прокси, разбирающие протоколы до содержимого, а с конца 2000-х — NGFW (Palo Alto Networks, 2007): идентификация приложений независимо от порта, интеграция с IPS и фильтрация по пользователям. В Linux-мире стандартом стали iptables (2001, подсистема netfilter) и его преемник nftables, в облаке ту же роль играют security groups — stateful-фильтры, привязанные к виртуальной машине с первого дня EC2 (2006).

Честное ограничение: файервол не защищает от того, что проходит сквозь разрешенные порты, — от веб-атак, фишинга, вредоносных вложений. Он сужает поверхность атаки, но не заменяет защиту приложений и людей.

IDS, IPS и сегментация

Системы обнаружения вторжений выросли из работы Джеймса Андерсона для АНБ (1980) и модели Дороти Деннинг (1987): анализ аудита на аномалии и сигнатуры. Джин Ким с Юджином Спарфордом написали Tripwire (1987) — контроль целостности файлов, Мартин Роэш в 1998-м выпустил Snort — сетевой IDS с открытым кодом и языком сигнатур, ставший отраслевым стандартом (Sourcefire, купленная Cisco в 2013-м за 2,7 миллиарда долларов), с 2009 года Open Information Security Foundation развивает Suricata с многопоточностью. IPS — тот же анализ, но с блокировкой в реальном времени, цена — ложные срабатывания, рвущие легитимный трафик. Современные наследники — EDR/XDR: агенты на конечных точках с телеметрией поведения, именно они сегодня ловят lateral movement, которое классические IDS пропускают.

Сегментация — второй кит внутренней защиты: VLAN, подсети и микросегментация делят сеть так, чтобы взлом веб-сервера не открывал дорогу в бухгалтерию и к контроллеру домена. DMZ выносит публичные сервисы за внутренний периметр, bastion hosts дают единственную контролируемую точку входа администраторов. Ценность сегментации измеряется не в предотвращенных проникновениях, а в стоимости lateral movement: NotPetya выкосил Maersk целиком именно потому, что плоская корпоративная сеть не знала внутренних границ.

Zero trust: конец периметра

Модель «замок со рвом» умерла вместе с облаками, удаленной работой и мобильными устройствами: «внутри» и «снаружи» больше не определены. Концепцию нулевого доверия сформулировал Джон Киндерваг из Forrester в 2010 году в докладе с характерным названием «Больше никаких мягких центров»: не доверять ничему по умолчанию, проверять каждый запрос, исходить из того, что сеть уже скомпрометирована. Самую масштабную реализацию проделал Google: после китайской операции Aurora (2009) компания построила BeyondCorp — доступ сотрудников к приложениям без VPN, через шлюз с постоянной перепроверкой личности, устройства и контекста каждого запроса (серия статей 2014–2017 годов). В августе 2020-го NIST выпустил SP 800-207 — эталонную архитектуру zero trust, в США ее внедрение стало требованием к федеральным агентствам после указа 2021 года, а CISA публикует модель зрелости. Принципы просты и безжалостны к legacy: минимальные привилегии на каждый запрос, аутентификация устройства наравне с пользователем, микросегментация, телеметрия всего, неявный отказ по умолчанию. Zero trust — не продукт, а десятилетняя перестройка архитектуры, но направление необратимо: периметр растворился, и каждая сессия стала границей.

Веб-безопасность: прикладной фронт

OWASP Top 10: карта угроз

Прикладной уровень атакуют не криптографией, а логикой. Карта угроз — список OWASP Top 10, публикуемый с 2003 года, свежая редакция представлена 6 ноября 2025 года на Global AppSec и зафиксировала сдвиг отрасли: первое место много лет держит Broken Access Control — ошибки разграничения прав, куда теперь влили и SSRF, второе — Security Misconfiguration, третье заняла новая категория Software Supply Chain Failures, за ними Cryptographic Failures, Injection, Insecure Design, Authentication Failures, Software or Data Integrity Failures, Logging & Alerting Failures и вторая новая категория — Mishandling of Exceptional Conditions. Показательно, чего в списке больше нет отдельной строкой: за двадцать лет индустрия в основном научилась не конкатенировать SQL и экранировать вывод — базовые инъекции ушли в прошлое, уступив место ошибкам архитектуры, прав и зависимостей.

Классика жанра и ее лечение

SQL-инъекция — подстановка кода в запрос через пользовательский ввод — впервые публично описана в 1998 году и до сих пор встречается в диком виде: так в 2017-м Equifax потеряла данные 147,9 миллиона американцев через неустановленную заплатку Apache Struts, выплатив затем 700 миллионов по мировому соглашению. Лечение тривиально известно: параметризованные запросы и ORM, принцип наименьших привилегий для учетки БД. XSS — внедрение скрипта в страницу — прославил червь Samy на MySpace (октябрь 2005-го): за двадцать часов он набрал миллион «друзей», защита — контекстное экранирование, Content Security Policy, флаги cookie HttpOnly и SameSite. CSRF — принуждение залогиненного пользователя к действию на чужом сайте — нейтрализуется

токенами и атрибутом SameSite, который Chrome с 2020 года ставит по умолчанию. SSRF — вынужденные запросы сервера к внутренним ресурсам — прославился взломом Capital One (июль 2019-го): неправильно настроенный WAF позволил через метаданные EC2 забрать роль IAM и вынести данные 106 миллионов клиентов, облачные метаданные (169.254.169.254) с тех пор — первое, что проверяют пентестеры.

Отдельная эпидемия — секреты в коде: ключи API, токены и пароли, закоммиченные в публичные репозитории, GitGuardian считает их утечки миллионами в год, а сканеры секретов встроены в CI всех крупных платформ. Сюда же — debug-режим в проде, админки без rate limiting, дефолтные пароли панелей, открытые бакеты S3: значительная часть «громких утечек» — не взлом, а находка. Набор гигиенических заголовков и настроек давно канонизирован: HSTS, CSP, X-Content-Type-Options, Referrer-Policy, Secure/HttpOnly/SameSite для cookies, Subresource Integrity для сторонних скриптов, обязательная проверка подписи входящих webhook, rate limiting и алерты на аномалии. Скучно, зато именно скучная гигиена закрывает большинство дыр, через которые реально входят.

Приватность: шифрование — еще не анонимность

Метаданные и анализ трафика

Даже идеальный TLS скрывает содержимое, но оставляет метаданные: кто соединяется, когда, как долго, сколько байт. Анализ трафика — старейшая техника разведки, пережившая шифрование: по объему и ритму соединений специалисты восстанавливают, что пользователь смотрит видео, звонит или торгует, а по SNI в незашифрованном ClientHello долго было видно и домены (проблему закрывает Encrypted Client Hello, развертываемый Cloudflare и браузерами). Старое правило разведки гласит: важнее не что написано в письме, а кто, кому, когда и как часто пишет. Новый уровень — машинное обучение по формам трафика: исследования показывают возможность классифицировать сайты и приложения по статистике пакетов даже внутри TLS. Отсюда главный вывод раздела: конфиденциальность содержимого и приватность поведения — разные свойства, и второе криптографией не решается.

Микс-сети и Tor

Ответом на анализ трафика стали микс-сети Дэвида Чаума — статья «Неотслеживаемая электронная почта» (1981) предложила пропускать сообщения через цепочку узлов, каждый из которых знает только предыдущего и следующего. Идея выросла в onion routing: в середине 1990-х Пол Сайвисон, Майкл Рид и Дэвид Гольдшлаг из Военно-морской исследовательской лаборатории США спроектировали схему для защиты разведывательного трафика, а в 2002 году Сайвисон с Роджером Динглдайном и Ником Мэтьюсоном запустили Tor, с 2004-го проект финансировал EFF, с 2006-го он независимый. Клиент строит цепочку из трех узлов, шифруя пакеты слоями, как луковица: никто из узлов не видит одновременно отправителя и содержание, выходной узел знает назначение, но не источник. К 2026 году сетью пользуются около трех миллионов человек в день через примерно девять тысяч волонтерских реле, крупнейший потребитель мостов

и обходов цензуры — Россия. Tor доказал принципиальную возможность анонимности для всех сразу: журналистов, диссидентов — и преступников одновременно, закрытие Silk Road в октябре 2013-го (Росс Ульбрихт, пожизненный срок, замененный помилованием в январе 2025-го) не уничтожило теневые рынки, но и не отменило ценность Tor для тех, кому анонимность — вопрос жизни. Полная анонимность хрупка: ошибки приложений, уникальные параметры браузера и поведенческие паттерны деанонимизируют быстрее, чем ломают криптографию.

Сквозное шифрование и спор о «going dark»

Сквозное шифрование (E2EE) переносит криптографию на концы: провайдер сервиса физически не может прочитать сообщения. Стандарт де-факто — Signal Protocol Мокси Марлинспайка с механизмом Double Ratchet, меняющим ключи на каждое сообщение, в апреле 2016-го его включил WhatsApp для миллиарда пользователей, аналогичные схемы работают в iMessage (с Advanced Data Protection для резервных копий с декабря 2022-го), в Google Messages и частично в Telegram. E2EE закрывает содержание от всех, включая сам сервис, — и потому тридцать лет идет спор о «going dark»: в феврале — марте 2016 года ФБР судилось с Apple, требуя вскрыть iPhone стрелка из Сан-Бернардино, и отозвало иск, заплатив внешним подрядчикам около 1,3 миллиона долларов за разблокировку, Британия в Online Safety Act 2023 года и Австралия в 2018-м принимали законы, обязывающие «помогать» спецслужбам, в ЕС периодически всплывает идея «контроля сообщений» (chat control). Криптографы неизменно отвечают: ослабление E2EE для избранных — это бэкдор для всех, а клиентское сканирование (Apple отказалась от него в 2021-м под давлением сообщества) противоречит самой идее сквозного шифрования. К 2026 году спор не решен ни в одну сторону: E2EE стоит, законы копятся, а приватность окончательно стала политическим, а не только техническим свойством протоколов.

Экономика безопасности и человеческий фактор

Цена взлома и цена защиты

Безопасность — рынок с измеримыми ценами. Совокупный ущерб киберпреступности Cybersecurity Ventures оценивает в 10,5 триллиона долларов за 2025 год с прогнозом 12,2 триллиона к 2031-му — больше, чем оборот мирового наркобизнеса и треть ВВП США. Цена отдельного инцидента растет: средняя стоимость утечки в мире достигла 4,99 миллиона долларов — рекорд и плюс 12% за год (IBM, июль 2026-го), в США — более чем вдвое выше мировой. Новая строка расходов — искусственный интеллект: каждое четвертое злонамеренное вторжение 2026 года использует ИИ (рост на 56% за год) и обходится в среднем в 6 миллионов долларов, а организации с «теневым ИИ» — неконтролируемым использованием сотрудниками внешних моделей — платят в среднем 5,39 миллиона против 4,99. Жертв тоже больше: по подсчетам ITRC, только за первое полугодие 2026-го в США зафиксировано 1803 компрометации и 471,2 миллиона уведомлений пострадавших — уже больше, чем за весь 2025 год.

Профицит спроса рождает дефицит кадров. По исследованию ISC2 2024 года, в отрасли работали 5,5 миллиона специалистов при разрыве в 4,8 миллиона — рекордном в истории наблюдений, в 2025 году ISC2 отказался публиковать оценку разрыва, поскольку практики важнее количества стали называть нехватку конкретных навыков: реагирования на инциденты, безопасности облаков, инженерии. Отрасль отвечает автоматизацией (SOC на ИИ-агентах), аутсорсингом MSSP и — с другой стороны баррикад — багбаунти: программы вознаграждений Google, Microsoft, Apple и платформы вроде HackerOne выплатили исследователям суммарно сотни миллионов долларов, превратив «белую» охоту за уязвимостями в профессию. Нормы раскрытия уязвимостей (coordinated disclosure), каталог CVE (MITRE, 1999), шкала CVSS и каталог эксплуатируемых уязвимостей CISA — инфраструктура, сделавшая рынок информации о дырах относительно цивилизованным.

Человек как вектор атаки

Главная мишень современной атаки — не шифр и не файервол, а человек. Verizon DBIR 2026 фиксирует человеческий элемент в 62% взломов, социальная инженерия — третья по распространенности схема (16%), а телефонный претекстинг вырос на 40% за год. Хрестоматийный пример — MGM Resorts, сентябрь 2023-го: атакующие нашли сотрудника в LinkedIn, позвонили в службу поддержки от его имени, получили сброс учетки — и за десять дней простоя казино потеряло около 100 миллионов долларов. Та же схема Scattered Spider в 2025-м легла в основу атак на Marks & Spencer, Co-op и Jaguar Land Rover. ФБР (IC3) оценивает потери американцев за 2024 год в 16,6 миллиарда долларов, из них деловой фишинг и BEC — 2,8 миллиарда. Наследие Кевина Митника — от взломщика, взятого в 1995-м, до консультанта, умершего в июле 2023-го, — сводится к формуле: «человека взломать дешевле, чем систему». Ответ индустрии сместился с инструктажей на архитектуру: FIDO-ключи и passkey, которые нельзя передать по телефону, потому что привязаны к домену, симуляции фишинга без наказаний, минимальные привилегии, при которых компрометация сотрудника не равна компрометации сети. И парадокс экономики: безопасность по-прежнему платит за скучное — патчи, MFA, сегментацию, бэкапы, — тогда как взлом платит за исключение.

Квантовая угроза и будущее криптографии

Harvest now, decrypt later

В 1994 году Питер Шор показал: квантовый компьютер достаточной мощности решает задачи факторизации и дискретного логарифмирования за полиномиальное время — то есть ломает RSA, Диффи — Хеллмана и эллиптические кривые разом. Практический вопрос не «если», а «когда»: машины на сотнях кубитов существуют (Google Willow, декабрь 2024-го, впервые продемонстрировал снижение ошибок при масштабировании), до криптографически значимых миллионов стабильных кубитов — по оценкам отрасли, от пяти до пятнадцати лет. Опасность в том, что ждать нельзя: противники уже сегодня записывают перехваченный зашифрованный трафик в расчете расшифровать его потом — harvest now, decrypt later. Бельгийская госбезопасность в 2022 году прямо обвинила

китайские APT в сборе зашифрованных данных правительственных сетей именно с таким расчетом, для данных с секретностью десять лет и больше (медицина, гостайна, персональные данные) угроза уже актуальна.

Постквантовая криптография

NIST запустил конкурс постквантовых алгоритмов в декабре 2016-го, в июле 2022-го объявил победителей, а 13 августа 2024-го утвердил первые стандарты: FIPS 203 (ML-KEM, обмен ключами на решетках, он же CRYSTALS-Kyber), FIPS 204 (ML-DSA, подписи, CRYSTALS-Dilithium) и FIPS 205 (SLH-DSA, консервативные подписи на хешах). Внедрение идет быстрее любого криптографического перехода в истории: гибридный набор X25519MLKEM768 — классическая кривая плюс решетка, чтобы не зависеть от одного нового алгоритма, — Chrome включил по умолчанию в 2024-м, Cloudflare развернул его на своей сети, к 2026 году постквантовым обменом ключей защищено более половины веб-трафика — с примерно двух процентов за два года. АНБ в профиле CNSA 2.0 требует от систем национальной безопасности США перехода на PQC к 2030–2035 годам, миграция предприятий измеряется тем же десятилетием — мешают размеры постквантовых подписей (в разы больше RSA), нагрузка на PKI и устройства, которые не обновятся никогда. Ключевое свойство, которое закладывают сегодня, — криптографическая гибкость (crypto agility): возможность сменить алгоритм конфигурацией, а не переписыванием протокола. TLS 1.3 уже сменил алгоритм обмена ключами без единого релиза «TLS 2.0» — и это лучший аргумент, что гибкость проектируется заранее. История безопасности учит: алгоритмы переживают десятилетия, реализации — годы, а системы, не рассчитанные на замену криптографии, — не переживают ничего.

Путь одного вторжения: собираем все вместе

Проследим типовой сценарий атаки на условную компанию — от первого письма до отчета об инциденте — и увидим, как каждый разобранный механизм работает на практике. Все совпадения с реальными инцидентами намеренны: сценарий собран из материалов Verizon, CISA и разборов атак 2023–2025 годов.

Атака начинается с разведки. Аффилиат вымогательской группы изучает поверхность: Shodan показывает VPN-шлюз компании, журналы Certificate Transparency — поддомены и тестовый стенд, LinkedIn — сотрудников ИТ-отдела, Whois — регистратора домена. Составляется цель: не «взломать компанию», а конкретная учетная запись с правами администратора. Дальше — социальная инженерия: фишинг по сотруднику поддержки подрядчика или прямой звонок в help desk с просьбой «сбросить MFA, я потерял телефон» (схема Scattered Spider). Письмо проходит спам-фильтры: домен-двойник свежий, SPF и DKIM у отправителя свои, легитимные, — DMARC не спасает от look-alike домена, а пользователь видит привычное оформление портала. Если MFA нет или это push-уведомление, которое можно «замучить», учетка падает, если сотрудник работает с FIDO2-ключом, привязанным к домену, — атака на этом шаге умирает, и атакующий идет искать другой вход.

Предположим худший случай: доступ получен. Здесь включается эшелонированная защита. Вход с незнакомого устройства в необычной географии должен поднять алерт IdP, conditional access — потребовать дополнительный фактор или заблокировать, EDR на рабочей станции — заметить попытку дампа учетных данных из LSASS. Но допустим, SOC завален ложными срабатываниями и эпизод утонул в шуме. Атакующий закрепляется: регистрирует собственное «доверенное» устройство, включает пересылку почты, создает OAuth-приложение с постоянным доступом — современные техники работают в облаке идентификации, не оставляя файлов на диске. Начинается lateral movement: перечисление групп Active Directory, kerberoasting сервисных учеток, поиск незакрытых шар, доступ к серверу резервных копий. Сегментация решает исход: в плоской сети путь от почтового ящика до контроллера домена занимает дни (так было в NotPetya и Change Healthcare), в сегментированной — упирается в межсетевые экраны между зонами, микро-сегментацию и запрет горизонтальных перемещений рядовым учеткам.

Кульминация — достижение цели. Данные собираются в архив и готовятся к выводу: классика — через разрешенные облачные хранилища или DNS-туннели, чтобы не поднимать алерты DLP. Если защита сработала хотя бы частично, сценарий вырождается в чистое вымогательство без шифрования — тренд 2024–2026 годов: украденное достаточно продать или пригрозить публикацией. Параллельно вторая линия атаки — DDoS как отвлекающий маневр или как наказание за отказ платить: аренда ботнета обходится дешевле, чем один час простоя производства, как показали Jaguar Land Rover и Collins Aerospace. И здесь тоже все из предыдущих разделов: anycast-скраббинг провайдера, rate limiting на WAF, — или парализованная регистрация пассажиров, если защита периметра проектировалась «на спокойные времена».

Финал — обнаружение и реагирование. В лучшем случае атакующего заметил не человек, а поведенческая аналитика: аномальный объем чтения файлового сервера в три часа ночи. Дальше работает runbook: изоляция хостов, отзыв токенов и сессий, ротация секретов (включая ключи TLS и сертификаты — скомпрометированный приватный ключ сервера требует немедленного перевыпуска через ACME), сохранение образов для форензики, юридический контур — уведомление регуляторов и пострадавших в сроки GDPR и местных законов. Post-mortem возвращает нас к началу статьи: почти каждый громкий инцидент разбирается в те же несколько провалов — отсутствие MFA на критичном портале, плоская сеть, неотслеженные секреты, проигнорированные алерты, неустановленные заплатки. Ни один из них не требует от атакующего взлома криптографии: AES и ECDHE не пали ни разу — пали люди, конфигурации и процессы. Именно поэтому безопасность — это система, а не алгоритм.

Итоги

Сетевая безопасность — единственная тема, которая проходит сквозь все уровни: от кадра Ethernet с его доверчивым ARP до политики регуляторов вокруг сквозного шифрования. Все разобранное укладывается в несколько сквозных линий.

Первая линия — доверие как фундамент. Интернет проектировали без замков, и каждый механизм защиты — это способ заново ввести доверие в среду, где его нет: сертификат подтверждает идентичность, подпись — авторство, PKI делегирует доверие институтам, Kerberos — центру, passkey — устройству в руках владельца. Доверие транзитивно и потому хрупко: история DigiNotar, Comodo и Symantec показала, что компрометация одного центра ставит под удар всех, — и индустрия ответила прозрачностью (CT), ограничением полномочий (CAA), короткоживущими сертификатами и автоматизацией Let's Encrypt. Вопрос «кому мы доверяем и что, если это доверие сломается» — главный вопрос любого протокола безопасности.

Вторая линия — математика надежна, реализации нет. За сорок лет не взломан ни AES, ни RSA достаточной длины, ни эллиптические кривые — ломались реализации и окружение: Heartbleed (ошибка bounds-checking в OpenSSL), слабый генератор случайных чисел в Debian, константа Sony вместо nonce, ROCA в чипах Infineon, отключенная проверка сертификатов в мобильных клиентах. Гибридная схема — дорогая асимметрика для рукопожатия, дешевая симметрика для трафика, PFS против записи «на потом» — стала каноном TLS, IPsec и WireGuard. И этот канон уже переписывается: постквантовые алгоритмы на решетках внедряются гибридами — более половины веб-трафика 2026 года защищено X25519MLKEM768, — потому что harvest now, decrypt later делает миграцию срочной задолго до появления квантового компьютера.

Третья линия — гонка без финиша. Каждое поколение защиты порождает новое поколение атак: BEAST, CRIME, POODLE, FREAK, Logjam против старых режимов TLS, KRACK против рукопожатия WPA2, Dragonblood против свежайшего WPA3, Log4Shell против библиотеки журналирования, закладка в xz — против самого процесса разработки открытого ПО. Асимметрия неизменна: защитник обязан закрыть все звенья цепочки, атакующему достаточно одного. Поэтому современная доктрина — не «не пустить», а эшелонированная оборона с презумпцией компрометации: сегментация, минимальные привилегии, мониторинг поведения, быстрые заплатки и crypto agility как страховка от неизвестного будущего.

Четвертая линия — атаки стали индустрией. От одиночек 1980–1990-х (Столл, Моррис, Митник) через червей 2000-х (Slammer, MyDoom) к государственным операциям (Stuxnet, NotPetya, SolarWinds) и корпоративным вымогателям с моделью RaaS, аффилиатами и техподдержкой для жертв. Рекорды DDoS выросли с 300 Гбит/с (Spamhaus, 2013) до 31,4 Тбит/с (Kimwolf, 2025) — терабитные атаки из зараженных ТВ-приставок стали рутиной, а число атак мощнее терабита за квартал выросло в шесть раз. Ущерб измеряется триллионами в год (10,5 триллиона за 2025-й), средняя утечка — почти пятью миллионами долларов, а нападение на один завод или госпиталь останавливает целые отрасли: от заправок Восточного побережья США до вылетов из Хитроу.

Пятая линия — человек остается главным вектором. DBIR 2026: 62% инцидентов с человеческим элементом, телефонная социальная инженерия растет на 40% в год, пароль «123456» по-прежнему первый в мировых рейтингах, а MGM, Marks & Spencer и Change Healthcare пали не через криптографию, а через help desk. Ответ — архитектурный: FIDO2

и passkey (пять миллиардов ключей, 36% успешных входов), которые невозможно выманить звонком, симуляции фишинга, приватность по умолчанию — E2EE в мессенджерах, DoH в браузерах, минимум метаданных. Криптография защищает каналы, но поведение людей защищают только механизмы, не требующие от людей бдительности.

Шестая линия — безопасность стала свойством всей системы. Нулевое доверие растворяет периметр: каждый запрос аутентифицируется заново, каждое устройство проверяется, каждый доступ выдается под задачу. Supply chain превратилась в самостоятельный фронт — OWASP Top 10 2025 закрепил это новой категорией, а SolarWinds, Log4Shell и xz показали цену слепых зон в зависимостях. Приватность отделилась от шифрования: Tor, ECH и споры о «going dark» идут уже не за содержимое, а за метаданные и за право на анонимность как таковую. От семидесяти пяти центов Столла в 1986-м до пяти миллиардов passkey и постквантового TLS 2026-го путь безопасности — это путь Интернета от клуба единомышленников к среде, где доверие не предполагается, а доказываются математикой, протоколами и процессами каждый день, заново, на каждом запросе. Гонка не закончится никогда: в этом ее смысл и в этом — цена открытой сети, которую стоит защищать.