Паролі у 2026 році: вичерпний інженерний путівник з автентифікації

Виконавче резюме

Парольна автентифікація вмирає повільно й неприглядно. Станом на квітень 2026 року **22% усіх підтверджених витоків даних починаються з компрометації облікових даних** (Verizon DBIR 2025), середня вартість витоку сягає **$4,88 млн** (IBM Cost of a Data Breach 2024), а медіанний час виявлення credential-based інциденту — **292 дні**. Кожен другий споживач використовує один пароль на двох і більше сервісах (Forbes Advisor 2024), 85% повторюють паролі взагалі (Bitwarden 2024), а глобальний топ-пароль роками залишається `123456`.

Паралельно світ робить найбільший за останні тридцять років зсув автентифікаційної парадигми. **Понад 15 мільярдів акаунтів** уже підтримують passkeys, **3+ мільярди активних passkey-credentials** видані, **1+ мільярд користувачів** мають хоча б один. Microsoft, Google, Apple та Amazon роблять безпарольний вхід стандартом за замовчуванням. Нова редакція **NIST SP 800-63B-4, опублікована 31 липня 2025 року**, офіційно поховала композиційні правила (`Aa1!`-вимоги) та примусову періодичну зміну паролів — практики, які десятиліттями радше шкодили безпеці, ніж її посилювали.

Ця стаття — спроба зібрати в одному документі історію, математику, інженерію зберігання, моделі атак, NIST-стандарти, інструменти для самоаудиту, ринок менеджерів паролів, ландшафт MFA, технологію passkeys, біометричні нюанси, deepfake-загрози, post-quantum-міграцію та практичні чек-листи. Цільова аудиторія — IT-фахівці, розробники з відповідальністю за автентифікацію та технічно підковані користувачі, яким недостатньо порад "придумайте складний пароль".

---

1. Коротка історія: від CTSS 1961 до memorized secrets

Перша комп'ютерна парольна система з'явилася у **1961 році на CTSS (Compatible Time-Sharing System)** в MIT — її створив Фернандо Корбато, майбутній лауреат премії Тюрінга. Паролі зберігалися у відкритому вигляді в звичайному файлі. Вже навесні **1962 року** аспірант Аллан Шерр здійснив перший задокументований витік в історії обчислювальної техніки: він подав заявку на офлайн-друк парольного файлу через картотеку перфокарт, отримав роздруківку й роздав паролі колегам, щоб приховати

перевикористання своєї квоти машинного часу. Інцидент розкрили лише через чверть століття, на 25-річчі CTSS, коли Шерр сам зізнався.

Цей епізод — мікрокосм усієї подальшої історії: **зберігання у plaintext, інсайдер з доступом, мотивація обходу обмежень**. Більшість сучасних інцидентів повторюють той самий шаблон, лише в інших масштабах — RockYou у 2009 році (32 млн plaintext), Adobe у 2013 (153 млн зашифровано слабким симетричним 3DES ECB), LinkedIn у 2012 (167 млн SHA-1 без salt).

У сучасній термінології NIST пароль називається **memorized secret** (запам'ятований секрет) і належить до фактора *знання* (knowledge factor). Терміни змінилися, фундаментальна вразливість — ні: людина має пам'ятати рядок, який атакувальник може відгадати або витягти.

---

2. Термінологія: що саме ми називаємо паролем

Розрізнення трьох понять критично для розуміння рекомендацій.

- **Пароль (password)** — зазвичай 8–20 символів з широкого алфавіту до 94 друкованих ASCII. Оптимізований під ручний набір.
- **PIN-код** — коротка числова послідовність 4–6 цифр. Ентропія 4-цифрового PIN — лише ~13,3 біта; його захист тримається не на криптостійкості, а на **жорстких лімітах кількості спроб** (зазвичай 3–10), після яких пристрій блокується або стирається. Поза лімітами PIN тривіально брутфорситься.
- **Passphrase (парольна фраза)** — довша послідовність слів з пробілами, часто 20+ символів. NIST Rev.4 однозначно фіксує: *"Passwords and passphrases are equivalent. The difference is that the passphrase is longer and can contain spaces."* Тобто це не два різні класи захисту, а одна сутність із різною довжиною й структурою.

Окремо варто розрізняти **три послідовні концепти автентифікаційного процесу**:

1. **Ідентифікація** — заявлення *хто ви* (логін, email, username).
2. **Автентифікація** — *підтвердження* цієї заяви через один або більше факторів: знання (пароль), володіння (ключ, телефон), інгерентність (відбиток, обличчя).
3. **Авторизація** — визначення, *що саме* підтверджена особа має право робити.

Більшість користувацьких помилок народжуються зі змішування цих понять. Біометрія, наприклад, добре розв'язує етап автентифікації *локально на пристрої*, але майже не годиться для серверної ідентифікації віддалених користувачів.

---

3. Математика стійкості: ентропія та чому вона завжди завищена

Теоретична ентропія пароля в бітах обчислюється за формулою:

$$H = L \times \log_2(N)$$

де *L* — довжина пароля, *N* — розмір алфавіту, з якого незалежно й рівноймовірно вибирається кожен символ. Кожен додатковий біт ентропії **подвоює** середню кількість спроб, які потрібні для повного перебору; брутфорс у середньому розкриє пароль за 2^(H−1) спроб.

Конкретні значення для типових алфавітів:

Довжина	**Лише цифри (N=10)**	**Малі літери (N=26)**	**Змішаний регістр + цифри (N=62)**	**Повний ASCII (N=94)**
6	19,9 біт	28,2 біт	35,7 біт	39,3 біт
8	26,6 біт	37,6 біт	47,6 біт	**52,4 біт**
10	33,2 біт	47,0 біт	59,5 біт	65,5 біт
12	39,9 біт	56,4 біт	71,5 біт	**78,6 біт**
16	53,2 біт	75,2 біт	95,3 біт	**104,9 біт**
20	66,4 біт	94,0 біт	119,1 біт	131,1 біт

Критична обмова: ця формула справедлива виключно для *рівномірно випадкових* паролів. Як тільки в гру вступає людина, ентропія обвалюється. Дослідження 34 000 паролів зі зливу MySpace (2006) показало: лише 8,3% поєднували регістри, цифри й символи; середня реальна ентропія — **40,5 біт замість теоретичних 50+**. Словниковий запас середньостатистичного носія мови — 50 000 слів, тобто одне слово додає максимум **~15,6 біт незалежно від його довжини**. L33t-заміни (a→@, o→0, s→$) додають буквально 1–3 біти, бо атакувальні словники й правила (mangling rules) знають усі шаблони напам'ять.

Звідси одна з центральних тез цього звіту: **довжина важить більше за класи символів**. 16-символьний пароль із самих малих літер (75,2 біт) перевершує 8-символьний з повного ASCII (52,4 біт) на понад тисячократу за середньою кількістю спроб.

3.1. zxcvbn та сучасні моделі оцінки стійкості

Композиційні правила типу "обов'язково велика літера + цифра + спецсимвол" не вимірюють реальну стійкість, бо ігнорують патерни. **zxcvbn** Дена Вілера (Dropbox, USENIX Security 2016) — індустріальний стандарт оцінки стійкості, який працює принципово інакше. Замість підрахунку класів символів він *моделює, як саме атакувальник буде вгадувати*: розбирає пароль на токени, розпізнає словникові слова, дати, клавіатурні патерни (`qwerty`, `1qaz2wsx`, `asdfgh`), послідовності, l33t-заміни, поширені імена та прізвища, видає метрику `guess` — оцінку реальної кількості спроб до відгадування.

Бібліотека повертає **score 0–4**:

Score	**Значення**	**Реальний поріг**
0	too guessable	$< 10^3$ спроб
1	very guessable	$< 10^6$ спроб (online attack feasible)
2	somewhat guessable	$< 10^8$
3	safely unguessable	$< 10^{10}$
4	very unguessable	$\geq 10^{10}$ (захищає від offline GPU-атак)

zxcvbn перетворив парольні UI: користувач отримує миттєвий зворотний зв'язок не за формальними правилами, а за реальною стійкістю. Бібліотека портована на Python, Go, Java, Ruby, інтегрована в 1Password, Okta, Bitwarden та незліченні IAM-системи. Її використовує і **генератор паролів ITEZ**: бар оцінки стійкості показує саме `zxcvbn(password).score` з мітками *Дуже слабкий → Дуже надійний*.

Поряд із zxcvbn в академії використовують три типи генеративних моделей для оцінки guess number:

- **Markov-моделі** — статистично враховують умовну ймовірність наступного символу за попередніми; добре передбачають типові переходи (`qu`, `ing`, `123`).
- **PCFG (Probabilistic Context-Free Grammars)** — розбирають пароль на структурні складові (`L7D2S1` = 7 літер + 2 цифри + 1 символ) і генерують варіанти за ймовірністю кожного граматичного правила.
- **Нейромережі** (LSTM, GAN) — натреновані на корпусах злитих паролів, видають гладкий розподіл імовірностей; PassGAN та подібні моделі стискаються до десятків кілобайт і працюють у браузері.

Усі ці моделі сходяться на одному висновку: *guess number* для людських паролів драматично нижчий за теоретичну ентропію Шеннона. Тому будь-яка серйозна оцінка стійкості (включно з парольним UI) має використовувати моделі, а не формулу `L × log₂(N)`.

---

4. Як зламують паролі у 2025–2026

Сучасний ландшафт атак на автентифікацію стратифікований за вектором, цільністю та необхідною компетенцією зловмисника. Опишу головні класи з прикладами реальних кейсів.

4.1. Brute force — повний перебір

Атакувальник перебирає всі можливі комбінації `N^L` за алфавітом і довжиною. Ефективний насамперед в **офлайн-сценарії**, коли БД хешів уже викрадена: rate-limiting не діє, обмеження — лише швидкість заліза. Online-брутфорс через форму входу елементарно блокується rate-лімітами (5–10 спроб + delay).

4.2. Dictionary attack — словники й правила

Замість сліпого перебору атакувальник використовує підготовлені словники. Класичні:

- **rockyou.txt** — 14,3 млн паролів зі зливу RockYou 2009; досі стандартний словник у Kali Linux.
- **HaveIBeenPwned Pwned Passwords** — 800+ млн унікальних SHA-1 хешів зі справжніх витоків; завантажується пакетом ~16 ГБ.
- **Burnett 10M corpus** — кураційований набір з найпоширеніших паролів.
- **CrackStation Wordlist** — масивний словник з MD5/SHA-1 рейнбоу для legacy-хешів.

До словника застосовуються **mangling rules**: капіталізація першої літери, l33t-заміни, append року (`Password2024`, `Password2025`), додавання спецсимволу. Інструменти Hashcat і John the Ripper мають мову правил, що дозволяє автоматично генерувати десятки варіацій на одне слово.

Базовий приклад атаки в Hashcat (для тестування власних систем):

Атака зі словником і базовим набором правил для bcrypt-хешів

hashcat -m 3200 -a 0 hashes.txt rockyou.txt --rules-file rules/best64.rule

```
# Ось як перевірити slow-hash проти HIBP-словника

hashcat -m 22000 wpa-handshake.hccapx hibp-passwords.txt --workload-profile 4
```

4.3. Rainbow tables — попередньо обчислені ланцюжки

Райдужні таблиці — попередньо обчислені структури hash → reduction → hash, що дозволяють реверсити *unsalted* MD5/SHA-1/LM хеші за час пошуку, а не повного перебору. Вирізняли їх у нульових. Сьогодні **повністю нейтралізовані salt'ом**: при 128-бітовому випадковому salt атакувальнику довелося б побудувати 2^{128} окремих таблиць — обчислювально неможливо. Райдужні таблиці залишаються релевантними лише для legacy-систем без salt (Windows LM, старі MD5-БД).

4.4. Credential stuffing — повторне використання пар

Атакувальник бере мільярдні combolist'и (пари email:password) з минулих витоків і прогонить їх через нові сервіси. Success rate — лише ~0,1%, але при мільярдах спроб це дає колосальну абсолютну кількість компрометацій. **Класичний кейс — 23andMe, квітень–вересень 2023**: 14 000 акаунтів зламали напряму через credential stuffing. Ключовий момент — функція *DNA Relatives* розширила доступ до даних 6,9 млн профілів. Наслідки: settlement $30 млн, штраф ICO £2,31 млн, банкрутство компанії в березні 2025. Один із найдорожчих credential-інцидентів в історії.

4.5. Password spraying — інверсія брутфорсу

Замість багатьох паролів проти одного акаунта — один-два популярні паролі (`Welcome123`, `Summer2024`, `Pa$$w0rd`) проти багатьох акаунтів. Це обходить класичні lockout-механізми, бо на кожен акаунт — лише одна-дві спроби. Microsoft публічно констатує: **понад 97% identity-атак на Microsoft 365 — це password spray або brute force**. У листопаді 2023 саме цим вектором російська група Midnight Blizzard (APT29) скомпрометувала корпоративні акаунти самого Microsoft — атаку маскували residential-проксі.

4.6. Phishing та AiTM (adversary-in-the-middle)

Будь-яка криптостійкість пароля безсила перед фішингом — користувач сам віддає секрет. **AiTM-проксі** (Evilginx, EvilProxy, Tycoon) — поточне покоління інструментів, які проксують увесь трафік між жертвою й справжнім сервісом, перехоплюючи не тільки пароль, а й session cookie *уже після* успішного MFA. Технічно: атакувальник підіймає реверс-проксі на схожому домені, реальний сервіс бачить легітимний логін, жертва бачить справжній інтерфейс, а зловмисник зливає сесійний токен з cookies. **MFA через TOTP, SMS і навіть більшість push-схем не рятує** — рятує лише FIDO2/WebAuthn з origin-binding.

4.7. Соціальна інженерія

Класичний кейс — **Twitter, липень 2020**: vishing-атака (voice phishing) на адмінів дала зловмисникам доступ до інструментів модерації; були зламані акаунти Барака Обами, Ілона Маска, Джеффа Безоса під Bitcoin-скам. Жодна довжина пароля чи серверне хешування тут не рятує — атакувальник обходить технологію через людину.

---

5. Швидкості сучасного заліза: чому вибір алгоритму важить

Уся теорія стійкості замикається на питанні "скільки спроб за секунду може зробити атакувальник". Бенчмарки Hashcat 2024 (Chick3nman) на одній **NVIDIA RTX 4090**:

Алгоритм	**Швидкість на RTX 4090**
MD5	165 ГГц/с
NTLM (Windows)	288,5 ГГц/с
SHA-256	22,7 ГГц/с
bcrypt(cost 5)	184 кГц/с
bcrypt(cost 10)	**14 кГц/с**

RTX 5090 (2025) додає ще ~45% швидкості, MD5 досягає ~240 ГГц/с. Корпоративна рига KnowBe4 з **24× RTX 4090** виходить на **7,25 трильйонів NTLM-хешів на секунду** — теоретичний верхній край споживацьких збірок. Хмарні GPU-ферми (AWS, RunPod, vast.ai) дають аналогічну потужність за лічені долари за годину.

Зведена таблиця Hive Systems (29 квітня 2025, конфігурація 12× RTX 5090, bcrypt cost 10) — час брутфорсу **сучасних бізнес-хешів**:

Довжина	**Лише цифри**	**Малі літери**	**Змішаний регістр**	**Складний (ASCII)**
6	миттєво	2 секунди	3 години	22 хвилини
8	миттєво	3 тижні	місяці	**4 дні – 164 роки**

Довжина	Лише цифри	Малі літери	Змішаний регістр	Складний (ASCII)
10	1 година	роки	століття	**~803 роки**
12	роки	тисячі років	мільйони років	**~3 000 років**
18	століття	нескінченність	нескінченність	трильйони років

Ключовий висновок цього розділу: **bcrypt(cost 10) у 11 мільйонів разів повільніший за MD5**. Тобто питання "як я зберігаю паролі на сервері" впливає на стійкість сильніше, ніж питання "які символи я вимагаю від користувача". Один і той самий 8-символьний пароль ламається за секунди при MD5 і за роки при bcrypt — вибір алгоритму є **множником у 10^6–10^7** до часу атаки.

---

6. Зберігання паролів: хешування ≠ шифрування

6.1. Принципова різниця

Хешування — одностороння функція з властивістю preimage-resistance: маючи `hash`, обчислювально неможливо відновити вхідний `password`. **Шифрування ж двостороннє** — наявність ключа дає змогу розшифрувати дані. Для паролів шифрування фатально: ключ стає єдиною точкою відмови, і його витік миттєво компрометує всю БД. OWASP формулює без двозначностей: *"Passwords should be securely hashed using modern, adaptive hashing algorithms — rather than encrypted or stored in plaintext."*

Уроки з історії жорстокі. **RockYou (грудень 2009)** — 32 млн паролів зберігалися у plaintext; файл `rockyou.txt` досі є стандартним словником атак. **Adobe (жовтень 2013)** — 153 млн акаунтів зашифровано симетричним 3DES у режимі ECB зі спільним ключем для всіх записів; до того ж password hint зберігався у plaintext поряд із зашифрованим паролем. Аналітики відновили мільйони паролів, просто корелюючи однакові ECB-блоки з підказками. Хрестоматійний приклад того, чому **encryption ≠ hashing**.

6.2. Salt: непомітний, але необхідний

Salt — унікальне випадкове значення (мінімум 16 байтів), яке додається до пароля перед хешуванням і зберігається в БД поруч із хешем. Дві ключові ролі:

1. **Унікалізація** — два користувачі з однаковим паролем мають різні хеші. Атакувальник не може зразу побачити, що 100 акаунтів мають той самий пароль.

2. **Анулювання rainbow tables** — для кожного можливого salt'а довелося б будувати окрему таблицю. При 128-бітному salt це 2^{128} таблиць, фізично неможливо.

Важливо: **salt не секретний**. Його зберігають відкрито в БД поряд із хешем. Безпека salt полягає виключно в унікальності.

6.3. Pepper: захист від суто-БД-витоків

Pepper — додатковий секретний рядок, спільний для *всіх* паролів системи, який зберігається **окремо від БД** (HSM, environment variable, окремий vault). Hash тоді обчислюється як `hash(salt ‖ password ‖ pepper)` або через HMAC: `HMAC(pepper, salt ‖ password)`.

Цінність pepper: якщо атакувальник викрав *тільки БД* (типовий сценарій SQL-інжекції чи compromised backup), він не зможе брутфорсити хеші без pepper. Це **defense-in-depth**, який не замінює salt і не дозволяє послаблювати інші захисти, але ускладнює офлайн-атаку.

Обмеження: ротація pepper потребує примусової зміни всіх паролів, бо старі хеші стають невалідними. Тому pepper рідше використовують у малих проектах і частіше у великих банківських/державних системах.

6.4. Чотири алгоритми, які OWASP рекомендує у 2025

Алгоритм	**Рік**	**Тип**	**Рекомендовані параметри**
Argon2id *(перший вибір)*	2015	memory-hard	m=19 MiB, t=2, p=1
scrypt	2009	memory-hard	$N=2^{17}$, r=8, p=1
bcrypt	1999	CPU-bound	cost ≥10, обмеження 72 байти
PBKDF2 *(FIPS-сумісний)*	2000	CPU-bound	≥600 000 ітерацій HMAC-SHA-256

Argon2id — переможець Password Hashing Competition 2015, стандартизований у RFC 9106 (2021). Гібридний режим між Argon2i (стійкий до side-channel) та Argon2d (стійкий до GPU) — балансує обидва клас атак. Memory-hard означає, що алгоритм потребує великого обсягу швидкої пам'яті на ітерацію, що нейтралізує перевагу GPU/ASIC (вони мають багато паралельних ядер, але обмежений швидкий on-chip RAM).

scrypt (Колін Персіваль) — попередник Argon2 з аналогічною ідеєю. Ще активно використовується, але Argon2id вважається кращим вибором для нових систем.

bcrypt — заснований на Blowfish, перевірений 25 роками експлуатації. Має дві обмежувальні особливості: вхід обрізається після 72 байтів (довгі passphrase ефективно укорочуються), і алгоритм — CPU-bound, тому сучасні GPU дають велику перевагу. Все ще прийнятний для legacy й нових систем, де Argon2 недоступний, але cost має бути ≥10.

PBKDF2 — старий, "офіційний" вибір. Сертифікований під FIPS-140, що робить його обов'язковим у багатьох федеральних/банківських системах. Не memory-hard, тому повільно "дорослий" — OWASP підвищив рекомендований мінімум до **600 000 ітерацій HMAC-SHA-256** саме через прогрес GPU.

6.5. Загальна схема правильного зберігання

```
register(username, password):
    salt = csprng_bytes(16)
    hash = argon2id(password, salt, m=19MiB, t=2, p=1)
    db.store(username, salt, hash)

verify(username, password):
    record = db.fetch(username)
    candidate = argon2id(password, record.salt, ...)
    return constant_time_compare(candidate, record.hash)
```

Два важливі додаткові пункти:

- **Constant-time compare** — звичайний `==` для байтових рядків може витікати інформацію через тайминг (рання відмова при першому розбіжному байті). Криптографічні бібліотеки мають окрему функцію (`hmac.compare_digest` у Python, `crypto.timingSafeEqual` у Node.js).
- **Tunable cost** — параметри Argon2/bcrypt мають зростати з часом. Раз на 1–2 роки переоцінюйте, чи не настав час підняти cost factor. Деякі системи перехешовують паролі при наступному успішному вході з посиленими параметрами.

---

7. NIST SP 800-63B-4: офіційний поворот галузі

31 липня 2025 року NIST опублікував фінальну редакцію SP 800-63B-4 — стандарту digital identity guidelines, який де-факто диктує політики для всіх федеральних систем США та задає тон комерційному сектору в усьому світі. Зміни щодо паролів у Rev.4 — кульмінація восьмирічного перегляду, який почався з Rev.3 у 2017 році під впливом досліджень Лорі Крейнор (FTC, 2016 — *"Time to rethink mandatory password changes"*) та Кормака Герлі (Microsoft Research).

Зведена таблиця основних вимог Rev.4:

Вимога	**Правило**
Мінімум, коли пароль — єдиний фактор	**15 символів (SHALL)** — нова вимога Rev.4
Мінімум для інших випадків	8 символів
Максимум підтримки	≥64 символи, повний Unicode, пробіли
Композиційні правила (`Aa1!`)	**ЗАБОРОНЕНО вимагати**
Періодичні зміни за графіком	**ЗАБОРОНЕНО** — тільки при компрометації
Перевірка за breach-blocklist	**Обов'язкова**
Підказки (password hints)	**Заборонені**
Секретні питання	**Заборонені**
Truncation пароля	Заборонено
Rate limiting	Обов'язково

7.1. Чому скасували композиційні правила

Раціонал NIST і академічних досліджень CMU CUPS lab: композиційні правила створюють **передбачувані шаблони**. Користувач, якого змушують додати велику літеру, цифру й символ, не вигадує справжньо випадкову комбінацію — він конвертує `password` у `Password1!`. Атакувальні словники з mangling rules давно знають усі типові трансформації, тому ентропія додається мінімально, а usability страждає сильно: користувачі забувають варіації, записують паролі на стікерах, повторно використовують одну формулу на всіх сервісах.

7.2. Чому скасували періодичну ротацію

Аналогічно з примусовою зміною щокварталу: якщо `Password1` уже скомпрометовано, то `Password2` буде вгадано за лічені секунди. Дослідження UNC та CMU показали, що при примусовій ротації **75% користувачів обирають передбачувано трансформовані попередники**. NIST тепер вимагає змінювати пароль *тільки* при підозрі компрометації або відповідно до сповіщення системи моніторингу витоків.

7.3. Чому ввели обов'язковий breach-blocklist

Жоден набір правил не захистить від ситуації, коли користувач вибрав пароль, який уже фігурує в HIBP. NIST тепер вимагає від systems перевіряти кожен новий пароль проти blocklist скомпрометованих — найпростіша реалізація через HIBP API з k-anonymity (див. розділ 13).

7.4. Зворотна сумісність

Rev.4 не вимагає миттєвої зміни вже існуючих паролів — лише при наступній установці нового або при підозрі компрометації. Це гасить хвилю незадоволення під час впровадження, але водночас означає, що legacy-паролі продовжують жити роками.

---

8. Як створити справді надійний пароль

8.1. Diceware: метод, що не старіє

Запропонований Арнольдом Райнхолдом у 1995 році **Diceware** — найкраще рішення проблеми "як вибрати високоентропійний пароль, який можна запам'ятати". Алгоритм:

1. Беремо стандартний словник Diceware (7776 слів = 6^5, кодованих п'ятизначними числами 11111–66666).
2. Кидаємо 5 фізичних кубиків для кожного слова.
3. Беремо 5–7 слів, з'єднуємо пробілами або дефісами.

Кожне слово додає **$\log_2(7776) \approx 12{,}9$ біт** ентропії. Шість слів = ~77,5 біт — достатньо для захисту від офлайн-брутфорсу навіть на майбутніх ASIC. Сім слів (~90 біт) — практично непробивний бар'єр.

EFF Large Wordlist (2016) — модернізована версія Райнхолда: 7776 слів, обрані для легкої вимови, відсутності образливих термінів і мінімальної мультизначності. Рекомендований сучасний словник.

Приклад згенерованої passphrase:

correct horse battery staple example phrase

(Це канонічний приклад з XKCD #936; не використовуйте його буквально, бо він у словниках.)

8.2. XKCD #936: чому "Tr0ub4dor&3" — гірше за чотири слова

Комікс Рендалла Манро з 2011 року став культовою ілюстрацією. `Tr0ub4dor&3` (11 символів з l33t-замінами та спецсимволом) має приблизно 28 біт ентропії — вгадується за 3 дні офлайн. `correct horse battery staple` (4 слова, 28 символів з пробілами) — 44 біти — вгадується за 550 років.

Людський мозок запам'ятовує осмислену фразу драматично краще, ніж криптомесиво з заміною символів. Парадокс старих політик: вони змушували вибирати важко-запам'ятовувані й одночасно слабкі паролі.

8.3. Якщо потрібен випадковий пароль — використовуйте CSPRNG

Коли passphrase не годиться (наприклад, як master-password менеджера, ключ API, технічний секрет), використовуйте криптографічно стійкий генератор. У JavaScript це `crypto.getRandomValues(new Uint32Array(length))`. **Не використовуйте `Math.random()`** — він не CSPRNG і його послідовність передбачувана.

Генератор паролів ITEZ робить саме так:

```
const array = new Uint32Array(length);

crypto.getRandomValues(array);

for (let i = 0; i < length; i++) {

    password += allowedCharacters.charAt(array[i] % allowedCharacters.length);

}
```

Зауваження: операція `array[i] % allowedCharacters.length` має невеликий modulo bias (символи з початку алфавіту трохи частіше), але для алфавітів 26–94 на 32-бітному рандомі bias незначний. Для криптографічних застосувань, де bias недопустимий, використовують rejection sampling.

8.4. NordPass Top-20 2025 і чому вони не працюють

Глобальні топ-20 паролів за версією NordPass (вересень 2024 – вересень 2025, спільно з NordStellar):

123456, admin, 12345678, 123456789, 12345, password,

Aa123455, 1234567890, Pass@123, admin123, Admin@123,

111111, 123123, qwerty123, 1234567, qwerty,

1q2w3e4r, Abcd@1234, Welcome@123, P@ssw0rd

Близько **78% з них зламуються менш ніж за секунду**. Зверніть увагу: навіть `P@ssw0rd`, `Pass@123`, `Welcome@123` — паролі, які формально проходять композиційні правила (велика літера + цифра + спецсимвол) — у топі. Це підтверджує тезу NIST: формальні правила не корелюють зі стійкістю.

8.5. Найгірша звичка: повторне використання

Bitwarden World Password Day 2024: **85% респондентів повторюють паролі**. Forbes Advisor: 78% користуються однаковим паролем на 2+ сайтах, 52% — на 3+. SpyCloud: **64% користувачів зі скомпрометованими паролями використовують подібні паролі на інших акаунтах**.

Саме повторне використання робить credential stuffing атакою №1 на споживачів. Один витік маленького форуму компрометує банківські акаунти, корпоративні поштові скриньки, державні портали — всюди, де жертва використала ту саму пару email:password. Менеджер паролів — єдиний практичний спосіб ліквідувати цю звичку, бо людина фізично не здатна пам'ятати сотні унікальних 16-символьних рядків.

---

9. Менеджери паролів: математична необхідність

9.1. Чому менеджер — не опція

Аргумент простий: середня людина має ~300 онлайн-акаунтів (Dashlane 2025), безпечний пароль має бути 16+ символів і унікальним для кожного сервісу. Запам'ятати 300 × 16 випадкових символів — за межами людської пам'яті. Менеджер паролів робить дві речі:

1. **Зберігає** довгі унікальні паролі за зашифрованим vault.

2. **Автозаповнення** — і це важливіше, ніж зазвичай помічають: менеджер не вставить пароль на підроблений домен, навіть якщо він візуально ідентичний (зустрічаються phishing-домени з заміненими символами unicode). Це **неочевидний anti-phishing захист**.

9.2. Порівняння провідних менеджерів 2025–2026

Менеджер	Модель	Особиста	Сімейна	Вихідний код	Self-host	Особливості
Bitwarden	Хмара / self-host	Free (повноцінно) або Premium $1,65/міс	6 users, $3,99/міс	Open-source GPL	Так	AES-256, безлім. паролі/пристрої, TOTP у Premium
1Password	Хмара	$2,99/міс	5 users, $4,99/міс	Закритий	Ні	Унікальний 34-сим. Secret Key + майстер-пароль, Watchtower, Travel Mode
KeePassXC	Локальний	Безкоштовно	—	Open-source	N/A	.kdbx-файл, AES-256/ChaCha20, синк через Dropbox/Syncthing
Proton Pass	Хмара (Швейцарія)	Free або Plus $1,99/міс	6 users, $4,99/міс	Open-source	Ні	E2E, аудит Cure53, hide-my-email аліаси, інтеграція з Proton

Менеджер	Модель	Особиста	Сімейна	Вихідний код	Self-host	Особливості
Dashlane	Хмара (США)	$4,99/міс (free-план закрито 16.09.2025)	10 users, $7,49/міс	Закритий	Ні	Argon2, вбудований Hotspot Shield VPN, dark-web моніторинг

Усі хмарні менеджери (окрім KeePass, який локальний) працюють на **zero-knowledge архітектурі**: майстер-пароль ніколи не залишає клієнта, сервер бачить лише зашифрований vault-blob. Дешифрування відбувається на пристрої через KDF (Argon2id/PBKDF2) від майстер-пароля.

9.3. Що особливого в 1Password

Secret Key — 34-символьний випадковий рядок, генерований один раз при реєстрації. Зберігається в Emergency Kit (PDF-файл, який рекомендують роздрукувати) і додається до майстер-пароля при дешифруванні vault'а. Це означає, що навіть якщо атакувальник дізнається майстер-пароль (наприклад, через keylogger), він *усе одно* не зможе розшифрувати vault без Secret Key — той зберігається лише на пристроях користувача. Ефективно це двофакторна автентифікація на рівні дешифрування, а не лише входу.

9.4. Кейс LastPass 2022 — чому не всі менеджери однакові

Інцидент LastPass — найпотужніший аргумент за уважний вибір менеджера. Хронологія:

- **Серпень 2022** — атакувальник скомпрометував ноутбук одного з чотирьох senior DevOps-інженерів LastPass, які мали доступ до ключів дешифрування S3-бекапів. Початковий вектор — вразливість Plex media server, який інженер використовував удома.
- **Грудень 2022** — компанія публікує, що викрадено зашифровані vaults 25+ млн користувачів **плюс незашифровані URL сайтів, email, IP, білінг-адреси**. URL'и тут критичні: атакувальник бачить, що жертва має акаунти на криптобіржах і йде брутфорсити саме ці vaults у першу чергу.
- **2023** — KrebsOnSecurity пов'язує крадіжки криптовалют на $35+ млн у 150+ жертв з LastPass-vault'ами; жертви мали seed-фрази в записках.
- **2025** — додаткові $150 млн крадіжок, штраф ICO UK £1,2 млн (листопад 2025), class-action settlement $24,5 млн.

- **Грудень 2025** — TRM Labs звітує про продовжувані дешифрування vault'ів російськими акторами.

Ключова відповідальність LastPass — **слабкі дефолтні KDF-параметри** історично (5000 ітерацій PBKDF2 у старіших vaults), що зробило офлайн-брутфорс vault'ів технічно реальним для слабких майстер-паролів. Усе ще активна загроза три роки потому.

Консенсус експертів 2025–2026:

- Для приватності й бюджету — **Proton Pass або Bitwarden**
- Для максимального UX — **1Password**
- Для повного офлайн-контролю — **KeePassXC**
- **LastPass більше не рекомендується нікому**

9.5. Майстер-пароль і 2FA на самому менеджері

Менеджер захищає сотні акаунтів — отже, його власна безпека критична. Дві обов'язкові практики:

1. **Майстер-пароль — passphrase 6+ слів Diceware** або довгий випадковий рядок, який ви запам'ятали.
2. **2FA на сам менеджер** — бажано FIDO2 hardware key (YubiKey), як мінімум TOTP. SMS — категорично ні.

Якщо застосовуєте Bitwarden self-hosted, не нехтуйте резервним копіюванням vault'у та шифруванням бекапів окремим ключем.

---

10. Багатофакторна автентифікація: не всі фактори рівні

MFA комбінує два і більше факторів *з різних категорій*. Microsoft публікувала: **>99,9% скомпрометованих акаунтів не мали MFA**. Google — додавання recovery-телефону блокує 100% автоматизованих ботів, 99% bulk-фішингу, 66% таргетованих атак. Але числа стосуються переважно *автоматизованих* атак; сучасні AiTM-проксі обходять TOTP, SMS і push.

10.1. SMS-коди — нижній поверх

У NIST SP 800-63B-4 SMS офіційно отримав статус **RESTRICTED authenticator** — не заборонений, але з обмеженнями і вимогою попереджати користувача про ризики. Причини:

- **SIM swapping** — атакувальник переконує оператора зв'язку перенести номер на свою SIM. Класичний кейс: захоплення акаунта **@jack** (CEO Twitter Джек Дорсі) 30 серпня 2019 групою Chuckling Squad через SIM swap. У 2019 році було задокументовано крадіжки $2,4 млн у криптовалюті за участю співробітників AT&T і Verizon.
- **Атаки на протокол SS7** — старий телекомунікаційний протокол, через який можна перехоплювати SMS.
- **VoIP-переадресація** — для віртуальних номерів.
- **2024**: Authy повністю закрив desktop-клієнт 19 березня; Twilio (компанія-власник Authy) пережила витік телефонних номерів користувачів.

SMS прийнятний як другий фактор лише там, де нема альтернативи, і ніколи — для критичних систем (банкінг, корпоративний email, менеджер паролів).

10.2. TOTP — RFC 6238

TOTP (Time-based One-Time Password) — HMAC-SHA1 від спільного секрету (shared secret) і поточного 30-секундного вікна. Стандартизований у RFC 6238. Клієнти — Google Authenticator (з cloud-backup з 2023), Microsoft Authenticator, open-source альтернативи Aegis (Android) та Ente Auth. 2FAS, Bitwarden Premium, 1Password — інтегровані варіанти в менеджерах паролів.

Переваги перед SMS: працює офлайн, не вразливий до SIM swap і SS7. Обмеження: shared secret міститься на пристрої, тому крадіжка пристрою + слабкий PIN дають доступ. І головне — **TOTP не phishing-resistant**: AiTM-проксі ретранслює 6-значний код у реальному часі.

10.3. Push-нотифікації та MFA fatigue

Push-схеми (Microsoft Authenticator, Duo, Okta Verify) показують користувачу запит "Підтвердити вхід?" і чекають "Approve". Зручно, але породило техніку **MFA fatigue / prompt bombing**: атакувальник, який має пароль, надсилає 30–100 push-запитів вночі або під час робочого дня; рано чи пізно жертва натисне "Approve" з утоми, помилки чи в комбінації з vishing-дзвінком "це служба IT, підтвердьте".

Хрестоматійний кейс — **Uber, 15 вересня 2022**: Lapsus$-хакер купив креди підрядника на даркнеті, надіслав 30+ push, потім написав у WhatsApp нібито від IT і добився "Approve". Доступ дав VPN, Slack, AWS, G Suite, HackerOne. У 2022 той самий вектор уразив Cisco, Microsoft, Nvidia, Okta, Samsung, Twilio, MGM Resorts. **Cloudflare вистояв — бо мав FIDO2 hardware-ключі, які не вразливі до prompt bombing**.

Пом'якшення: **number matching** замість простого approve. Користувач у застосунку має ввести число, що показане на екрані входу, тобто має бачити, *куди саме* підтверджує вхід. Microsoft зробив number matching обов'язковим у 2023 році.

10.4. FIDO2 hardware keys — золотий стандарт

Hardware authenticator з FIDO2/WebAuthn — єдиний широко розгорнутий метод MFA з near-100% phishing-resistance.

Як це працює технічно:

1. Приватний ключ зберігається в апаратному secure element фізичного ключа (YubiKey, Titan Security Key, SoloKey, Nitrokey).
2. При реєстрації браузер генерує пару ключів і прив'язує її до origin (домену) сайту.
3. При логіні сервер надсилає challenge; ключ підписує його приватним ключем; сервер перевіряє підпис публічним ключем.
4. **На фішинговому домені ключ просто відмовиться підписувати** — origin не збігається з зареєстрованим. Це фундаментальний захист, відсутній у TOTP/SMS/push.

Сертифікація NIST AAL3 — найвищий рівень.

Ціни (квітень 2026):

- YubiKey 5 NFC — ~$55
- YubiKey Security Key (тільки FIDO2/U2F) — $25–30
- YubiKey Bio (відбиток) — $98
- Titan Security Key (Google) — $30–35
- SoloKeys / Nitrokey (open hardware) — $25–60

Forrester TEI 2024 (на замовлення Yubico): зниження успішного phishing і credential theft на **99,9%**, зменшення helpdesk-тікетів на 75%, 3-річний ROI 203%. У листопаді 2024 YubiKey 5 FIPS отримав сертифікацію FIPS 140-3.

Практична порада: купуйте мінімум **два ключі** — один основний, один резервний, зареєстровані на ваші критичні акаунти. Втрата єдиного hardware-ключа без резерву може заблокувати доступ.

10.5. Recovery codes

При налаштуванні MFA сервіс зазвичай видає 8–10 одноразових recovery codes. Їх треба зберігати:

- У менеджері паролів в окремому захищеному сховищі, або
- Роздрукувати і покласти в фізичний сейф.

Категорично не зберігайте recovery codes в email або plaintext-нотатках у хмарі. Це обнуляє всю безпеку MFA.

---

11. Passkeys і WebAuthn: безпарольне майбутнє вже тут

11.1. Що таке passkey технічно

Passkey — це FIDO-credential на основі асиметричної криптографії, стандартизований у WebAuthn (Web Authentication API від W3C) і CTAP2 (Client to Authenticator Protocol). Коротко:

1. **Реєстрація**: пристрій генерує пару ключів (зазвичай ECDSA на P-256 або EdDSA). Приватний ключ зберігається в апаратному secure element — Secure Enclave (iOS), Android Keystore / StrongBox, TPM (Windows), або в зовнішньому YubiKey. Публічний ключ і credential ID надсилаються на сервер.
2. **Прив'язка до origin**: ключ криптографічно прив'язаний до домену сайту. Браузер перевіряє origin перед запитом підпису.
3. **Логін**: сервер надсилає challenge; пристрій просить user verification (Touch ID, Face ID, Windows Hello PIN, або PIN на YubiKey); після верифікації приватний ключ підписує challenge; сервер перевіряє підпис публічним ключем.
4. **Приватний ключ ніколи не передається**. Витік БД сервера = публічні ключі = неможливість реверсити приватні. Жодних паролів для крадіжки.

11.2. Чому passkey фундаментально кращий за пароль + MFA

Чотири властивості, кожна з яких розв'язує конкретний клас атак:

Властивість passkey	**Атака, яку нейтралізує**
Origin-binding	Phishing, AiTM-проксі (Evilginx, EvilProxy)
Asymmetric crypto, no shared secret	Credential stuffing, breach-induced compromise
Hardware-bound private key	Keylogger, malware, token theft
User verification (biometric/PIN)	Fізична крадіжка пристрою

Жоден інший масово розгорнутий метод не закриває одночасно всі чотири.

11.3. Стандарти і портабельність

- **WebAuthn Level 2** — W3C Recommendation від 8 квітня 2021, поточна офіційна версія, що підтримується всюди.
- **WebAuthn Level 3** — Candidate Recommendation Snapshot від 13 січня 2026; додає Signal API, Conditional Create, JSON Serialization, Related Origins.
- **Credential Exchange Format (CXF/CXP)** — прийнято як industry-standard draft у серпні 2025. Розв'язує головну проблему перших років passkeys: неможливість мігрувати credentials між Apple, Google, Microsoft, 1Password, Bitwarden. CXP дозволяє експортувати/імпортувати passkeys із зашифрованої контейнерної структури.

11.4. Типи authenticators

- **Platform** — вбудовані в пристрій: Touch ID, Face ID на iPhone/Mac, Windows Hello (камера або PIN), Android biometric.
- **Roaming** — зовнішні ключі, що підключаються через USB, NFC або BLE: YubiKey, Titan Security Key.

Поза цим — поділ за моделлю синхронізації:

- **Device-bound** — приватний ключ ніколи не залишає одного пристрою (типово roaming-ключі типу YubiKey). NIST AAL3, прийнятно для регульованих галузей.
- **Synced** — ключ end-to-end зашифровано синхронізується через хмару власника платформи (iCloud Keychain, Google Password Manager, Microsoft Authenticator) або сторонній менеджер (1Password, Bitwarden). Кращий UX і recovery, але NIST AAL2.

11.5. Адопція 2025–2026: вибуховий зріст

FIDO Alliance Passkey Index (жовтень 2025) — зведена статистика:

- **>15 млрд акаунтів** уже можуть використовувати passkeys
- **>3 млрд активних passkeys** видані
- **>1 млрд користувачів** мають хоча б один
- **93% success rate** passkey-входів проти 63% для інших методів
- Час входу **8,5 секунд vs 31,2 секунди** (-73%)
- Зменшення helpdesk-тікетів на **81%**
- Awareness серед споживачів: **75% у 2025 проти 39% у 2023**
- 48% топ-100 сайтів підтримують passkeys (вдвічі більше за 2022)

Платформенні цифри:

- **Google**: 800 млн акаунтів з passkeys, **>2,5 млрд passkey sign-ins**, +352% зростання за рік; success rate +30%, час входу -20%.
- **Microsoft**: ~1 млн passkey-реєстрацій на день (травень 2025); **98% sign-in success vs 32% для паролів**; 3 секунди vs 69 секунд для password+MFA; passwordless як default для нових Microsoft accounts з травня 2025.
- **Amazon**: 175 млн користувачів з passkey, 6× швидше за паролі.
- **Apple**: iOS 16+ з 2022; iOS 26 (2025) додав credential portability через CXP.
- **Yahoo! Japan (LY Corp)**: 27 млн активних passkey-користувачів, ~50% автентифікацій на смартфонах — passkeys.

State of Passkey Deployment in Enterprise (HID/FIDO, лютий 2025): 87% організацій розгорнули або активно розгортають passkeys.

11.6. Залишкові виклики

- **Recovery** — синхронізовані passkeys розв'язують через хмарне резервування, але це повертає довіру до платформи. Device-bound passkeys потребують реєстрації мінімум двох ключів плюс recovery codes.
- **Cross-platform UX** — досі трохи громіздкий через QR-код hybrid transport, але CXP робить це краще.
- **Підтримка legacy-сервісів** — більшість сайтів усе ще не підтримує passkeys, тому пароль + менеджер паролів — паралельна реальність на найближчі 3–5 років.

---

12. Біометрія: зручний ключ, але не сам пароль

12.1. Типи біометрії

Capacitive-сенсори відбитків (Apple Touch ID з 2013, більшість Android) вимірюють електричну ємність гребнів шкіри. **Ultrasonic** (Qualcomm 3D Sonic у Samsung Galaxy S10+) будує 3D-мапу пальця звуковими хвилями і стійкіший до spoofing.

Apple Face ID на TrueDepth-камері: IR-flood illuminator + dot projector проектує >30 000 точок, neural engine будує 3D-модель обличчя, що зберігається в Secure Enclave. **Windows Hello Face** — IR-камера, простіша. **Apple Optic ID** (Vision Pro, 2024) — сканування райдужки.

Офіційні показники Apple:

- Face ID FAR (false accept rate): <1 на 1 000 000
- Optic ID: те саме
- Touch ID: <1 на 50 000 (один палець); при реєстрації 5 пальців ризик ~1 на 10 000

Статистика гірша для близнюків і дітей <13 років.

12.2. Принцип: біометрія розблоковує ключ, а не є ключем

Ключова перевага сучасних FIDO/passkey-систем: **біометричний шаблон ніколи не передається серверу**. Він зберігається локально в Secure Enclave / StrongBox / TPM як математична репрезентація (а не сире зображення обличчя/відбитка), і його єдина функція — *розблокувати* криптографічний приватний ключ для підпису WebAuthn-challenge.

Це принципово відрізняє сучасну модель від старих серверних біометричних систем (як voice biometry в кол-центрах банків — ABA Banking Journal у 2025 рекомендує цю модель припинити). Серверне зберігання біометрії — катастрофічна модель: на відміну від пароля, відбиток пальця змінити неможливо. Витік біометричної БД компрометує користувачів довічно.

12.3. Загроза deepfake — від дослідницької до промислової

У Q1 2025 deepfake-відео становили 46% фрод-інцидентів за біометричною верифікацією, фото — 32%, аудіо — 22%. **Voice cloning потребує лише 3 секунд аудіо**; готові deepfake-shopfloor пропонують картинки за $10–50, face-swap софт — за $1000–10 000.

Знакові кейси:

- **Лютий 2024, Гонконг**: фінансовий працівник Arup Group переказав $25 млн після відеодзвінка з deepfake-CFO та підробленими учасниками. Один із найдорожчих deepfake-інцидентів задокументовано.
- **Січень 2024**: робоголос Джо Байдена в праймеріз Нью-Гемпширу, згенерований за ~$1 і 20 хвилин роботи.
- **2023–2024**: понад 300 компаній найняли північнокорейських шпигунів з deepfake-інтерв'ю, вірячи, що це справжні розробники.

Gartner прогнозує: до кінця 2026 року 30% enterprises більше не довірятимуть ізольованій біометрії як єдиному методу автентифікації.

12.4. Регуляторне тло

- **BIPA (Illinois, 2008)** — Biometric Information Privacy Act, вимагає інформованої згоди. Приватні позови з штрафами $1000–$5000 за порушення; Facebook сплатив $650 млн settlement у 2021.
- **GDPR Art. 9** класифікує біометричні дані як sensitive data з особливим режимом.
- **EU AI Act (2024)** регулює real-time remote biometric identification у публічних місцях.

12.5. Висновок розділу

Біометрія сама по собі не замінює паролі і не є їх безпечною заміною. **Безпечна заміна пароля = passkey з біометричним розблокуванням локального hardware-ключа**. Цей синтез поєднує криптографічну стійкість FIDO з ергономікою біометрії, не передаючи жодних чутливих даних.

---

13. SSO, OAuth і чому session-cookie крадуть частіше за паролі

13.1. Стандарти

- **OAuth 2.0** (RFC 6749, жовтень 2012) — фреймворк *авторизації* (делегованого доступу), не автентифікації. Часто плутають із входом, але технічно OAuth 2.0 видає access token для доступу до API від імені користувача.
- **OpenID Connect (OIDC)** — лютий 2014, надбудова над OAuth 2.0, додає автентифікаційний шар через ID Token (JWT).
- **SAML 2.0** — OASIS, березень 2005, XML-based. Досі домінує в корпоративних інтеграціях з Active Directory; у нових розгортаннях поступається OIDC.

Consumer-SSO: **Sign in with Google**, **Sign in with Apple** (запущений на WWDC червень 2019, з ключовою функцією *Hide My Email* через relay-домен apple-id.apple.com), **Microsoft Entra** (ребренд з Azure AD 11 липня 2023).

13.2. Переваги і ризики

Переваги: один акаунт замість сотень (Dashlane 2025: середня людина має 301 credential), централізовані MFA і Conditional Access policies, зниження helpdesk-витрат, єдиний log audit trail.

Ризики — компрометація IdP знищує доступ до всіх RP. І — критично важливо — післяавтентифікаційні session tokens стають головною ціллю.

13.3. Кейс Okta, жовтень 2023

Атакувальник скомпрометував обліковий запис у customer support system Okta через особистий Google-профіль співробітника, синхронізований у корпоративному Chrome. Через support system отримав доступ до завантажених клієнтами **HAR-файлів** (HTTP Archive — записи трафіку з браузера, які клієнти надсилали для діагностики). У HAR-файлах міститься повний дамп HTTP-сесії, включно з session cookies. Атакувальник

використав ці токени для **session hijacking** п'яти клієнтів, серед них **1Password, BeyondTrust, Cloudflare**.

Доступ тривав з 28 вересня по 17 жовтня 2023 року. Скомпрометовано 134 з 18 400 клієнтів. Урок: навіть IdP найвищого рівня вразливий, якщо співробітник зливає корпоративну конкретну сесію через особистий Chrome-профіль.

13.4. Кейс Linus Tech Tips, 23 березня 2023

Хрестоматійна **крадіжка session-cookie через infostealer**. Співробітник LTT відкрив `.exe`-файл, замаскований під PDF-пропозицію спонсорства. Infostealer-malware (типу RedLine, Raccoon) за 30 секунд витяг cookies і tokens з Chrome та Edge, включно з session-tokens YouTube Google Workspace.

Атакувальники обійшли і пароль, і MFA — у них уже була пост-автентифікаційна сесія. Захопили три YouTube-канали, стримили крипто-скам з deepfake Маска, поки доступ не відкликали.

Verizon DBIR 2025 фіксує: у логах infostealer-зливів **30% корпоративно-керованих і 46% некерованих пристроїв** містять корпоративні credentials.

Ключовий висновок: **MFA не рятує, коли вкрали session cookie після успішного входу**. Рішення — короткі сесії, device-binding токенів (DPoP, mTLS), continuous re-authentication для чутливих операцій, моніторинг infostealer-зливів.

---

14. Витоки даних: що формувало галузь

14.1. Основні віхи

- **Yahoo (2013, розкрито 2016)** — 3 млрд акаунтів, найбільший одиночний breach в історії.
- **Equifax (2017)** — 147 млн SSN американців через unpatched CVE Apache Struts; settlement $425 млн.
- **Marriott/Starwood (2018)** — 500 млн, 4 роки прихованого APT-доступу.
- **Collection #1 (2019)** — 773 млн email + 21 млн паролів; запустила широку credential stuffing-епідемію.

2024 рік установив нові рекорди:

- **MOAB ("Mother of All Breaches")** — виявлений Бобом Дяченком і Cybernews 22 січня 2024. **26 млрд записів на 12 ТБ** з 3876 доменів: Tencent QQ 1,4 млрд, Weibo

504 млн, Twitter/X 281 млн, LinkedIn 251 млн, AdultFriendFinder 220 млн, Adobe 153 млн.
- **National Public Data (квітень–серпень 2024)** — 2,9 млрд записів зі SSN, 272 млн унікальних SSN. NPD збанкрутував і повернувся в серпні 2025.
- **RockYou2024 (4 липня 2024)** — компіляція **9 948 575 739 унікальних plaintext-паролів** (~10 млрд) на 145 ГБ, виклав форумер ObamaCare.
- **Snowflake-атаки** — UNC5537 через infostealer-credentials уразила 165+ організацій: AT&T (73 млн), Ticketmaster (560 млн), Santander.

14.2. Have I Been Pwned: де-факто стандарт моніторингу

Have I Been Pwned (HIBP) — створений Троєм Хантом 4 грудня 2013. Станом на квітень 2026 сайт показує **974 pwned websites, ~17,5 млрд pwned accounts**. Сервіс безкоштовний, інфраструктура на Cloudflare.

Найважливіший компонент — **Pwned Passwords API з k-anonymity**. Принцип:

1. Клієнт хешує пароль через SHA-1.
2. Бере перші 5 символів хеш-префікса.
3. Надсилає лише префікс на `https://api.pwnedpasswords.com/range/<prefix>`.
4. API повертає список усіх суфіксів хешів, що починаються з цього префікса, і кількість появ кожного у витоках.
5. Клієнт локально шукає свій повний хеш у списку.

Сам пароль ніколи не покидає браузер — k-anonymity забезпечує неможливість API дізнатися, який саме пароль перевіряли. Метод інтегрований у 1Password, Okta, Firefox Monitor, численні IAM-платформи.

ITEZ використовує точно цей підхід у власному генераторі паролів — функція `checkPassword()` у `passwordHIBP.js`:

```
const hash = await sha1(password);

const prefix = hash.slice(0, 5);

const suffix = hash.slice(5);

const response = await fetch(`https://api.pwnedpasswords.com/range/${prefix}`);

// Шукаємо власний suffix у відповіді — пароль не покидає клієнт
```

Якщо хеш знайдено, користувач отримує попередження `Знайдено у N витоках`. Це канонічна реалізація breach-blocklist, яку NIST Rev.4 робить обов'язковою для всіх систем автентифікації.

14.3. Чек-лист реакції на витік

При сповіщенні від HIBP або інших моніторингових сервісів (Mozilla Monitor, Apple Password Monitoring, 1Password Watchtower):

1. **Перевірити обсяг витоку** — які саме поля скомпрометовано (паролі, email, SSN, токени).
2. **Негайно змінити пароль пошти** — це корінь довіри для reset усіх інших акаунтів.
3. **Увімкнути phishing-resistant MFA / passkey** на всіх критичних сервісах.
4. **Ревокнути активні сесії** в Google/Microsoft/Apple account dashboard.
5. **Ревокнути OAuth-авторизації** третім сторонам.
6. **Заморозити кредит** (для США — Free Credit Freeze Act з 2018, безкоштовно).
7. **Увімкнути IRS IP PIN** — захист від подачі підробленої податкової декларації.
8. **Підписатися на HIBP notify** для майбутніх сповіщень.

---

15. Корпоративні політики, Zero Trust і майбутнє автентифікації

15.1. Zero Trust як мандат

NIST SP 800-207 (серпень 2020) кодифікував архітектуру Zero Trust з принципом *"never trust, always verify"*: continuous verification, least privilege, MFA everywhere, device trust, мікросегментація, assume breach. Раніше імпліцитна довіра до "внутрішньої мережі" — головний фактор поширення інцидентів типу SolarWinds.

Executive Order 14028 (Байден, 12 травня 2021) зобов'язав FCEB-агенції розробити ZTA-плани. **OMB Memo M-22-09** (26 січня 2022) поставив конкретну ціль: phishing-resistant MFA для всіх співробітників до FY2024.

CISA Zero Trust Maturity Model v2.0 (11 квітня 2023) визначає **5 стовпів** (Identity, Devices, Networks, Applications, Data) **× 4 стадії зрілості** (Traditional → Initial → Advanced → Optimal). **CISA Secure by Design pledge** (травень 2024) підписали 250+ вендорів, зобов'язуючись ліквідувати default-паролі і посилити MFA з коробки.

15.2. Continuous та adaptive authentication

Дискретний "вхід раз і користуйся годинами" поступається continuous authentication. Adaptive engines (Microsoft Entra Conditional Access, Okta AdaptiveMFA, Ping, Beyond Identity, CrowdStrike Falcon Identity) аналізують у реальному часі:

- **Device posture** — чи пристрій оновлений, чи має EDR, чи в Compliance.
- **Geo-velocity** — фізично неможливі переміщення (Київ → Сінгапур за 5 хвилин).
- **Behavioral biometrics** — ритм набору, рух миші, типові години активності.
- **Network reputation** — IP з відомих VPN, residential proxy, TOR-exit nodes.

CAEP / Shared Signals Framework (OpenID, 2024–2025) дозволяє push-revocation сигналів між IdP і RP у реальному часі: коли IdP бачить компрометацію акаунта, він може миттєво сповістити всі сервіси, що використовують цю identity.

AuthZEN (OpenID, 2025) — стандартизація continuous authorization, аналог OIDC для авторизації.

15.3. Decentralized Identity та user-controlled wallets

Парадигма гаманців ідентичності виходить з пілотного режиму:

- **W3C DID Core v1.0** (червень 2022) — стандарт Decentralized Identifiers.
- **Verifiable Credentials v2.0** (2024–2025) — формат криптографічно підписаних credentials.
- **EU Digital Identity Wallet** (eIDAS 2.0, Regulation 2024/1183) — обов'язкова видача гаманців усіма країнами-членами ЄС до 2026 року, €46 млн на пілоти.
- **mDL (ISO/IEC 18013-5)** — mobile driver's license, розгортається в California, Arizona, Colorado, Австралії.
- **OpenID for Verifiable Credentials (OID4VC, OID4VP)** — стандарти видачі та презентації.

NIST SP 800-63C-4 нормативно визнає user-controlled wallet модель.

15.4. AI як зброя обох сторін

Атакуюча сторона:

- Spear-phishing масштабується GenAI — персоналізовані листи в обсязі автоматизованого spam.
- Deepfake-injection на біометричні системи зросла на 200% у 2023, voice-clone атаки +442% у 2025.
- Кейс Arup ($25 млн) стає типовим, не виключенням.

Захисна сторона:

- IBM 2024: **$2,2 млн економії** при extensive AI/automation у SOC.
- UEBA (User and Entity Behavior Analytics) ловить аномалії в логах автентифікації, недосяжні rule-based системам.
- Gen-AI security tools — $167 000 зниження середньої вартості breach.

15.5. Post-quantum криптографія: фундамент змінюється

Поточні passkey (на ECDSA P-256, EdDSA) вразливі до scalar-quantum атак Шора при появі великих квантових комп'ютерів. NIST фіналізував три PQ-алгоритми **13 серпня 2024 року**:

- **FIPS-203** — ML-KEM (Kyber), key encapsulation
- **FIPS-204** — ML-DSA (Dilithium), digital signatures
- **FIPS-205** — SLH-DSA (SPHINCS+), digital signatures

FIPS-206 (FN-DSA, Falcon) очікується у 2025–2026. **CNSA 2.0** від NSA ставить migration timeline до 2030–2035 років для критичних систем.

TLS уже впроваджує hybrid PQ-KEM: **X25519Kyber768** активний у Chrome і Cloudflare з 2023–2024. Це означає, що навіть якщо квантовий комп'ютер з'явиться, минулий перехоплений TLS-трафік не зможе бути дешифрованим.

Загроза **harvest-now-decrypt-later**: атакувальники сьогодні зберігають перехоплені шифровані дані, щоб дешифрувати їх через 10–15 років. Для passkey це означає, що вже сьогодні треба планувати міграцію на PQ-сигнатури. **FIDO Alliance Post-Quantum Working Group** (2024) розробляє PQ-authenticators.

Практичний чек-лист: що робити сьогодні

Для звичайного користувача

1. **Встановіть менеджер паролів** — Bitwarden (безкоштовний), Proton Pass, 1Password або KeePassXC.
2. **Згенеруйте 6-словну Diceware passphrase** як майстер-пароль; запам'ятайте її через mnemonic.
3. **Увімкніть FIDO2 hardware key або passkey на сам менеджер** (мінімум TOTP).
4. **Замініть слабкі/повторні паролі** на 16+ символьні випадкові, згенеровані менеджером.

5. **Увімкніть passkeys на Google, Apple, Microsoft, Amazon, банк** — перші чотири вже підтримують.
6. **Перевірте email на HIBP** — підпишіться на сповіщення.
7. **Не використовуйте SMS як 2FA** там, де є альтернатива.
8. **Категорично не зберігайте recovery codes в email або хмарних нотатках**.

Для системного адміністратора

1. **Аудит зберігання паролів** — переходьте на Argon2id (m=19 MiB, t=2, p=1) або bcrypt cost ≥10.
2. **Впровадьте salt 16+ байтів** на кожен пароль, опційно pepper в HSM.
3. **Інтегруйте HIBP Pwned Passwords API** при реєстрації та зміні пароля.
4. **Видаліть композиційні правила і примусову ротацію** з парольних політик (NIST Rev.4).
5. **Підніміть мінімальну довжину до 15 символів** для single-factor сценаріїв.
6. **Увімкніть rate-limiting** з progressive delay після 5 невдалих спроб.
7. **Розгорніть FIDO2/WebAuthn** як другий фактор для всіх співробітників.
8. **Відключіть SMS як authenticator** для адмінських акаунтів.
9. **Скоротіть session timeout** і впровадьте device-binding токенів.
10. **Моніторте infostealer-зливи** через сервіси типу SpyCloud, Have I Been Pwned for Business.

Для CISO / архітектора безпеки

1. **Розробіть ZTA-roadmap** за CISA Maturity Model v2.0 — Identity pillar першочергово.
2. **Інвестуйте в phishing-resistant MFA** — YubiKey/Titan для всього персоналу. ROI за Forrester TEI — 203%.
3. **Перейдіть на passkeys для customer-facing продуктів** — економія helpdesk 81%, success rate +30%.
4. **Впровадьте CAEP/Shared Signals** для cross-IdP компрометаційних сигналів.
5. **Розпочніть post-quantum migration plan** — інвентаризація залежностей від ECDSA/RSA, перехід на hybrid TLS.
6. **Регулярні tabletop-вправи** з deepfake-інцидентами і MFA fatigue-сценаріями.
7. **Встановіть KPI** — % акаунтів з MFA, % з passkey, медіанний час від компрометації до детекції, обсяг infostealer-логів з корпоративними credentials.

Зведена таблиця: вибір методу автентифікації за сценарієм

Сценарій	**Рекомендований метод**	**Пояснення**
Особистий банк	Passkey + hardware FIDO2	Найвищий клас захисту, банки активно впроваджують
Робоча пошта	Passkey + Conditional Access	Найбільший вектор ROI для організації
Менеджер паролів	Майстер-passphrase + FIDO2	Захищає всі інші акаунти
Особистий email (gmail)	Passkey + recovery codes у сейфі	Корінь довіри, MFA must
Streaming / форуми	Унікальний пароль з менеджера	Низький ризик, але повторне використання компрометує інші акаунти
Корпоративний SaaS	SSO + Conditional Access + passkey	Централізована політика, audit trail
Адмінський SSH	FIDO2 hardware key + bastion	YubiKey з SSH-ca
API token	Short-lived JWT + DPoP	Резистентний до кражі токена

Заключення: парадигма зсунулася, звички — ні

Останні чотири роки принципово переписали правила автентифікації. Довжина побила складність. Passphrase витіснила криптографічне місиво. Примусова ротація визнана шкідливою. SMS зі золотого стандарту став RESTRICTED. Passkeys перейшли з новинки в дефолт для Microsoft, Google, Apple, Amazon. NIST SP 800-63B-4 офіційно благословив цю революцію 31 липня 2025 року.

Але розрив між сучасним стандартом і реальною практикою залишається прірвою. Глобальний топ-пароль досі 123456. 85% користувачів повторюють паролі. Credential abuse — перший вектор 22% усіх breach-ів. LastPass-vault'и продовжують дешифруватися

через слабкі майстер-паролі три роки після інциденту. Більшість IT-відділів все ще проводять "обов'язкову заміну пароля кожні 90 днів", не знаючи, що NIST уже сім років як від цього офіційно відмовився.

Парадоксально, але **найпотужнішим захистом 2026 року є не складний пароль, а відсутність пароля взагалі** — криптографічна пара ключів, розблокована біометрією в Secure Enclave, прив'язана до конкретного домену, яка вміє відмовитися працювати на фішинговому сайті швидше, ніж людина встигне помітити підміну.

Поки passkeys ще не покривають усе — лишайтеся з менеджером паролів, 16+ символьними генерованими секретами, FIDO2-ключем на критичних акаунтах і регулярним моніторингом HIBP. Не складно. І не дорого. І набагато надійніше за будь-який композиційно-валідний пароль, який ви наразі повторно використовуєте на трьох сервісах.

> **Інструменти ITEZ**: вбудований генератор паролів використовує `crypto.getRandomValues` (CSPRNG), оцінює стійкість через zxcvbn з відображенням ентропії в бітах і часу до брутфорсу, а також інтегрує перевірку HIBP через k-anonymity API — пароль ніколи не залишає ваш браузер. Це канонічна реалізація NIST SP 800-63B-4 best-practice для public-facing інструментів.

Джерела й подальше читання

- NIST Special Publication 800-63B-4: Digital Identity Guidelines (31 липня 2025)
- NIST SP 800-207: Zero Trust Architecture (серпень 2020)
- OWASP Password Storage Cheat Sheet (2025)
- RFC 9106: Argon2 Memory-Hard Function for Password Hashing
- FIDO Alliance Passkey Index (жовтень 2025)
- Verizon Data Breach Investigations Report 2025
- IBM Cost of a Data Breach Report 2024
- Hive Systems Password Table 2025
- Have I Been Pwned (haveibeenpwned.com)
- Daniel Wheeler. *zxcvbn: Low-Budget Password Strength Estimation*, USENIX Security 2016
- WebAuthn Level 3, W3C Candidate Recommendation, 13 січня 2026