Engineer

Onion-сервіс

адреса, що працює лише через Tor

Що це таке

Звичайний сайт знаходять за DNS-ім’ям, що вказує на IP-адресу, а onion-сервіс досяжний лише всередині мережі Tor, і його ім’я виведене з власного відкритого ключа сервісу. У розмові часто кажуть просто onion, один демон tor може тримати кілька onion-сервісів поруч, а специфікація й досі вживає давнішу назву hidden service.

Саме ім’я — це 56 символів base32, за якими йде `.onion`. Після декодування з них виходить 35 байтів: 32-байтовий ідентифікаційний ключ ed25519, двобайтова контрольна сума, взята з геша SHA3-256 цього ключа, і байт версії зі значенням 3. Цей байт стоїть останнім, тому кожна чинна адреса закінчується літерою d. Мітку ніхто не обирає: це і є ключ, записаний літерами. Імена версії 2, що були до неї, складалися з 16 символів, відрізаних від геша SHA-1 ключа RSA-1024, і tor відмовився від них у випуску 0.4.6.1-alpha 2021 року.

Щоб дві сторони зустрілися й жодна не розкрила, де вона є, протокол використовує два види місць зустрічі. Сервіс обирає кілька ретрансляторів точками знайомства (introduction points), тримає до них відкриті ланцюжки й публікує на ретрансляторах-каталогах підписаний дескриптор з їхнім переліком. Дескриптор зберігається під засліпленим ключем, який змінюється щоперіоду, а його вміст зашифровано, тож ретранслятор, що зберігає його, не знаючи адреси, переліку не прочитає. Клієнт, якому адреса відома, завантажує дескриптор, перевіряє підпис ключем з імені, сам обирає ретранслятор точкою зустрічі (rendezvous point) і через одну з точок знайомства просить сервіс прийти туди. Сервіс будує власний ланцюжок до точки зустрічі й завершує рукостискання, а ретранслятор з’єднує два ланцюжки. Увесь шлях проходить через шість ретрансляторів, по три від кожної сторони, і жоден кінець не дізнається IP-адреси іншого. Необов’язковий режим, що тепер зветься restricted discovery, а раніше

звався client authorization, додатково шифрує дескриптор для ключа кожного дозволеного клієнта, тож лише ці клієнти дізнаються, де точки знайомства.

Застосуйте поняття до рішення

Застосуйте «Що таке onion-сервіс?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Wi-Fi, фаєрволи і віддалений доступ налаштовані так, щоб працювати можна було звідусіль, а заходити — лише своїм.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Керувала 30 інстансами Ubuntu Linux VPS — 30 серверів Linux під управлінням, із планом відновлення

Керувала 30 інстансами Ubuntu Linux VPS Зв'язатися з нами →

Чому це важливо

Для звичайного сайту DNS і центр сертифікації ділять між собою завдання сказати читачеві, з ким він має справу. Onion-сервісу не потрібен жоден з них, бо цю роботу виконує саме ім’я. Перевіривши підпис дескриптора й завершивши рукостискання, клієнт знає, що на тому боці той, хто володіє приватним ключем за адресою. RFC 7686, який резервує `.onion` як доменне ім’я спеціального призначення, описує такі імена як поєднання розташування та автентифікації й робить висновок: onion-ім’я не можна зареєструвати, призначити, передати чи відкликати. Володіти ним означає розпоряджатися парою ключів, і нічого більше.

Шифрування випливає з тієї самої конструкції. Трафік до onion-сервісу ніколи не залишає мережі Tor, тож на дальньому кінці немає вихідного ретранслятора й немає останнього відрізка відкритим текстом. Звичайний `http://` на onion-адресі несе сторінку, зашифровану на всьому шляху, до хоста, чию особу клієнт звірив з ім’ям, яке отримав.

З погляду експлуатації не менш важливий напрямок з’єднань. Сервіс лише сам дзвонить назовні — до своїх точок знайомства й до кожної точки зустрічі, — тож йому не потрібен відкритий вхідний порт, і він працює за NAT. Перелік відкритих портів хоста лишається таким самим, як був.

Адресу читачам усе одно треба якось дізнатися. Сайт може оголосити її заголовком відповіді `Onion-Location`, і тоді Tor Browser показує в адресному рядку фіолетову кнопку, що веде до onion-копії сторінки. Tor Project ставить три умови: значення — це URL `http:` або `https:` з onion-хостом, сторінку,

що його надсилає, отримано через HTTPS, і сама вона не розміщена на onion. Там, де заголовки задати неможливо, те саме оголошення може нести метатег, але без шляху, тож кожна сторінка вказує на одну незмінну адресу.

Як це виглядає на практиці

Onion-дзеркало компанії на читабельній адресі описано крок за кроком. Демон встановлено з власного пакета дистрибутива, а вихідний порт проксі вимкнено. Приватний ключ, що відповідає адресі, згенеровано поза хостом, і він ніколи не потрапляв ні до репозиторію, ні на ноутбук; роль відмовляється перезаписувати ідентифікаційні дані, яких не генерувала сама, тож покладений вручну ключ використовується як є. Адреса записана в інвентарі як оголошене значення, і кожне застосування конфігурації перевіряє, чи файл ідентифікаційних даних на хості з ним збігається, тож підмінений ключ із застарілим оголошенням зупиняє прогін. Обидві сторожі секретів ловлять файл ключа — після того як вимірювання показало, що очевидний шаблон для файлу ключа збігався з ключем розгортання й проминав цей.

Читабельний префікс можна знайти, але не обрати. Кожен наступний символ множить очікуваний пошук на 32, тож вісім літер `engineer` потребують у середньому близько 1,1 трильйона пар ключів. Випадкову адресу, з якою дзеркало запрацювало спершу, вивели з ужитку поетапним переходом, і опублікована адреса відповідала весь час. Один крок здавався останнім, але ним не був: ключ ліг на місце, застосування конфігурації повідомило про відсутність змін, а нову адресу так і не опублікували, бо ніщо не називало її теки. Тепер роль завершується помилкою, якщо на диску є ідентифікаційні дані, яких не використовує жодна служба.

Сторінка подяк наводить повну адресу, `engineer2ezpeji52ipphljcflzyhetvy5r6tnqubtq45nzlxxv6rsqd.onion`, поруч з адресами Gemini та Gopher. Усі четверо вхідних дверей ведуть до однієї теки з файлами, тож onion-адресу оновлює те саме розгортання, що й три інші, а вебверсія надсилає `Onion-Location`, і Tor Browser сам пропонує onion-адресу. Сторінка також пояснює, що бачить власна мережа читача крізь кожні двері: через Gopher — яку сторінку читають, через Gemini — яку адресу запитали, через onion-сервіс — ні того, ні іншого. Префікс вона називає зручністю, а не доказом, і просить звіряти всю адресу.

Будову адреси можна перевірити без мережі. Ці рядки декодують її й заново обчислюють контрольну суму, і обидва порівняння друкують `True`:

```
import base64
import hashlib

label = "engineer2ezpeji52ipphljcflzyhetvy5r6tnqubtq45nzlxxv6rsqd"
raw = base64.b32decode(label.upper())
key, checksum, version = raw[:32], raw[32:34], raw[34:]
digest = hashlib.sha3_256(b".onion checksum" + key + version).digest()
print(version == b"\x03", digest[:2] == checksum)
```

Правильна контрольна сума показує, що адресу складено коректно, але не те, кому вона належить. Це підтверджує її поява на власних HTTPS-сторінках компанії.

Де виникають проблеми

Ключ — єдиний документ на володіння. За ім'ям не стоїть жоден реєстратор, тож утрачений приватний ключ означає втрачену адресу, а скопійований дає змогу комусь іншому відповідати на ній. Посібник Tor Project з налаштування радить тримати файли ключів у таємниці й робити резервну копію ключа, якщо сервіс має жити довго. Ставтеся до нього як до ключа підпису, а не як до конфігурації.

Читають лише перші літери. RFC 7686 застерігає, що зловмисник може знайти ключі, чиї імена схожі на потрібне, і що ризик більший, бо людині ці імена нічого не кажуть. Читабельні імена, здобуті ціною значних обчислень, RFC вважає одним із засобів захисту, але восьмисимвольний префікс лишає решту 48 символів такими ж непрозорими, як і раніше. Звіряйте всю адресу або приходьте через закладку чи посилання на власних HTTPS-сторінках сайту.

Сервер видає себе сам. Tor приховує, де працює сервіс, але програмне забезпечення за ним однаково може це сказати. Серед способів, якими ламається захист, RFC 7686 називає сервер, що розкриває свої ідентифікаційні дані в повідомленні рівня застосунку, і сторінка помилки зі справжнім іменем хоста машини — саме такий випадок. Посібник з налаштування також радить передавати трафік вебсерверу через Unix-сокет, а не TCP, щоб сервіс не просочувався в локальну мережу:

```
HiddenServiceDir /var/lib/tor/my-website/
HiddenServicePort 80 unix:/var/run/tor/my-website.sock
```

Найбільше це важить там, де саме розташування і є таємницею.

Ім'я потрапляє до DNS. Програма, що не вміє працювати з Tor, питає про onion-ім'я звичайний резолвер, і локальна мережа та постачальник DNS дізнаються, про який onion ішлося. RFC 7686 вимагає, щоб така програма повертала помилку без запиту, а DNS-сервери відповідали, що такого імені не існує. Передавайте ім'я клієнтові Tor.

Кнопка `Onion-Location`, яка так і не з'являється. Заголовок, заданий лише на переадресації зі звичайного HTTP, порушує умову про HTTPS, а заголовок з помилкою в адресі веде читачів у нікуди. Перевіряйте заголовок на остаточній HTTPS-відповіді.

Що почитати далі

- Tor Rendezvous Specification, version 3 — увесь протокол: ідентифікаційні й засліплені ключі, дескриптори, знайомство та зустріч.
- Encoding onion addresses — формула, що перетворює відкритий ключ на 56-символьне ім'я, з прикладами адрес.
- How do Onion Services work? — точки знайомства, дескриптори й точка зустрічі, розібрані крок за кроком.
- Set up Your Onion Service — рядки torrc, файли, які створює tor, і поради щодо ключів та витоків розташування.
- Onion-Location — налаштування заголовка й метатега для Apache, Nginx і Caddy та умови, яким має відповідати заголовок.
- RFC 7686 — чому `.onion` зарезервовано і що з ним мають робити резолвери, оператори DNS та реєстратори.

Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке onion-сервіс?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/onion/