GUÍA DE PRIVACIDAD

Tor a fondo

Guía completa

Esta guía explica cómo funciona Tor, la red que permite usar internet sin revelar quién eres ni desde dónde te conectas: circuitos, cifrado en capas, directorio, servicios onion, puentes, navegador y límites reales de la protección.

De la red de cebolla al análisis de tráfico

Contenido

INTRODUCCIÓN Y BASES

· Cómo usar esta guía 3

· Los conceptos básicos 3

CAPÍTULOS

01 Qué es Tor y qué problema resuelve 5

02 Modelo de amenazas: qué protege Tor y qué no 5

03 Circuitos de tres saltos 6

04 Celdas y cifrado en capas 7

05 El apretón de manos ntor 9

06 Directorio y consenso 11

07 Servicios onion 12

08 Puentes y transportes enchufables 14

09 Tor Browser 15

10 Ataques y límites 16

11 Tor, VPN y combinaciones 18

12 Tails y Whonix 19

13 Buenas prácticas 19

14 Cómo evoluciona el protocolo 20

15 Comunidad, financiación y legalidad 21

16 Fuentes 21

INTRODUCCIÓN

Cómo usar esta guía

Público y prerrequisitos. No hace falta saber de redes ni de criptografía. La sección "Antes de empezar" explica los conceptos básicos con ejemplos cotidianos, y la guía sube de nivel poco a poco hasta llegar al detalle del protocolo.

Estado de la red. Los datos describen Tor a mediados de 2026: la implementación clásica en C ("C Tor"), la reescritura en Rust llamada Arti, el nuevo cifrado Counter Galois Onion y las defensas contra la denegación de servicio en los servicios onion.

Cómo leerla. Si partes de cero, empieza por "Antes de empezar". Los capítulos 1 a 3 dan la visión general y se pueden leer sin más. Los capítulos 4 y 5 contienen las dos fórmulas clave del protocolo, explicadas paso a paso con ejemplos pequeños. Los capítulos 6 a 9 describen las piezas de la red y el navegador. Los capítulos 10 a 13 tratan de límites, combinaciones con otras herramientas y buenas prácticas. El capítulo 14 cuenta cómo evoluciona el protocolo y el 15 quién lo sostiene.

ANTES DE EMPEZAR

Los conceptos básicos

Para entender Tor bastan cuatro ideas: qué es una dirección IP, quién ve qué cuando navegas, qué hace un intermediario (proxy) y qué significa cifrar en capas.

La dirección IP

Cada dispositivo conectado a internet usa una **dirección IP**, un número parecido a una dirección postal. Cuando abres una web, tu paquete de datos lleva dos direcciones: la del destino y la tuya, para que la respuesta sepa adónde volver. Esa dirección suele revelar tu país, tu ciudad aproximada y tu proveedor de internet.

Quién ve qué

Sin ninguna protección, cada parte del camino ve algo distinto:

QUIÉN	QUÉ VE
Tu proveedor de internet	A qué servidores te conectas y cuándo, aunque la web use HTTPS
La web que visitas	Tu dirección IP y lo que haces en su sitio
Una red Wi-Fi pública	Tu tráfico no cifrado y los destinos de todo lo demás
Empresas de publicidad	Tu IP, tu navegador y un rastro de las webs que visitas

El cifrado HTTPS protege el *contenido* de lo que envías, pero no oculta con quién hablas. Esa información sobre quién habla con quién se llama **metadatos**, y es justo lo que Tor busca esconder.

Los intermediarios

Un **proxy** es un intermediario: tú hablas con él y él habla con la web en tu nombre. La web ve la dirección del proxy, no la tuya. El problema es que el proxy sí te ve a ti y a lo que pides, así que solo has cambiado de quién te fías. Una VPN funciona con la misma idea, con el túnel cifrado entre tú y el intermediario.

Tor evita ese punto único de confianza repartiendo el trabajo entre varios intermediarios, de modo que ninguno por sí solo sabe a la vez quién eres y qué haces.

Cifrar en capas

Imagina que quieres enviar una carta sin que nadie sepa quién la manda y a quién va. La metes en un sobre dirigido al destinatario, ese sobre en otro dirigido a un segundo intermediario, y este en un tercero dirigido a un primer intermediario. Cada intermediario abre solo su sobre y encuentra dentro la dirección del siguiente. Ninguno ve la carta completa ni conoce a la vez al remitente y al destinatario.

La red Tor hace exactamente eso con cifrado: tres capas, una por intermediario. Por eso se la llama *onion routing* (enrutado de cebolla).

CAPÍTULO 1

Qué es Tor y qué problema resuelve

Tor es una red de miles de ordenadores voluntarios, llamados **relays** (relés), que reenvían tráfico de internet en capas cifradas. Su nombre viene de *The Onion Router*. La idea del enrutado de cebolla nació en los años noventa en un laboratorio de investigación naval de Estados Unidos, y el diseño actual de Tor se presentó en un artículo académico de 2004. Hoy lo mantiene una organización sin ánimo de lucro, The Tor Project, con una comunidad de voluntarios que operan los relays.

Tor resuelve un problema concreto: que la web que visitas no conozca tu dirección IP y que quienes ven tu conexión no sepan qué web visitas. No cifra el contenido por sí mismo (para eso están HTTPS y el cifrado de extremo a extremo), no te vuelve invisible en una web donde inicias sesión con tu nombre y no convierte un ordenador comprometido en uno seguro.

PIEZA	QUÉ ES
Tor (la red)	Los relays voluntarios y las reglas con las que se coordinan
Tor Browser	Un navegador basado en Firefox que ya viene configurado para usar la red
Servicios onion	Sitios y servicios a los que solo se llega por Tor y cuya dirección termina en .onion
Puentes	Relays no listados públicamente para quien tiene Tor bloqueado
Arti	La reimplementación de Tor en el lenguaje Rust, en desarrollo activo

CAPÍTULO 2

Modelo de amenazas: qué protege Tor y qué no

Un **modelo de amenazas** responde a tres preguntas: qué quieres proteger, de quién y qué puede hacer esa persona o entidad. Tor se diseñó con un adversario concreto en mente.

Lo que Tor oculta. Tu dirección IP frente a los sitios web y servicios que visitas. Los sitios que visitas frente a tu proveedor de internet y frente a quien vigile tu red local. La relación entre ambos extremos frente a cualquier intermediario individual.

Lo que Tor no oculta. Que usas Tor, porque tu proveedor ve que te conectas a relays conocidos (los puentes reducen esto). Lo que haces dentro de una web si inicias sesión o rellenas formularios con datos que te identifican. El contenido de tu tráfico si la web usa HTTP sin cifrar: el relay de salida lo ve. Lo que hace tu propio dispositivo si está infectado.

El adversario para el que Tor no fue diseñado. Alguien capaz de observar a la vez la entrada y la salida de tu circuito puede comparar los patrones de tráfico y relacionar ambos extremos, aunque no pueda leer nada. Los diseñadores de Tor lo reconocen desde el principio: es un límite de la idea, no un fallo que se corrija con un parche. El capítulo 10 lo trata a fondo.

Por eso Tor sirve muy bien contra el rastreo comercial, la vigilancia local y la censura, y sirve de forma limitada frente a adversarios con visión de buena parte de internet, cuyo ataque exige además pericia y recursos.

CAPÍTULO 3

Circuitos de tres saltos

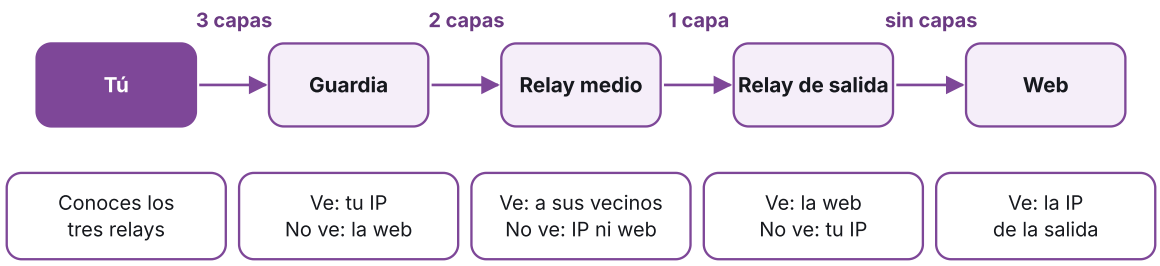
Cuando abres Tor Browser, tu cliente elige tres relays y construye con ellos un **circuito**: un camino de tres saltos por el que pasa todo tu tráfico. Cada relay solo conoce al anterior y al siguiente.

Guardias, relays medios y de salida

ROL	QUÉ SABE	QUÉ NO SABE
Guardia (entrada)	Tu dirección IP y que usas Tor	Qué web visitas
Relay medio	Solo al guardia y al relay de salida	Quién eres y qué web visitas
Relay de salida	La web destino y, si no hay HTTPS, el contenido	Quién eres

El **guardia** no se cambia en cada circuito: tu cliente elige uno y lo conserva durante semanas o meses. Si eligiera uno nuevo cada vez, tarde o temprano caerías en manos de un relay malicioso con probabilidad creciente. Con un guardia fijo, o ese guardia es honesto y tu entrada está

protegida, o no lo es y el riesgo se concentra en un solo lugar. El **relay de salida** es el que ve la web de destino; por eso los operadores de salidas son los que más quejas legales reciben.



Cada relay solo conoce al anterior y al siguiente

Cada relay solo ve a su vecino; las capas se van quitando a lo largo del circuito.

Cómo se construye un circuito

El cliente construye el circuito como un telescopio, salto a salto:

1. Se conecta al guardia por una conexión TLS y negocia con él una clave secreta (el apretón de manos del capítulo 5).
2. A través del guardia, ya cifrado, pide al relay medio que se una al circuito y negocia otra clave con él. El guardia solo reenvía bytes que no puede leer.
3. A través de los dos anteriores, repite el proceso con el relay de salida.

Al terminar, el cliente comparte una clave distinta con cada relay y ningún relay conoce las claves de los otros. Tor reutiliza un mismo circuito durante unos minutos y luego abre otro; Tor Browser además usa circuitos separados para sitios distintos, de modo que dos webs no puedan saber que eres la misma persona.

CAPÍTULO 4

Celdas y cifrado en capas

Todo lo que viaja por un circuito se empaqueta en **celdas** de tamaño fijo, de unos 512 bytes, con unos 498 bytes útiles de datos. Que todas las celdas midan lo mismo impide que un observador distinga una petición corta de una larga por el tamaño de los paquetes.

Las capas, una por relay

El cliente cifra cada celda tres veces, una por relay, como las capas de una cebolla. Cada relay quita su capa y reenvía el resultado. Solo el último relay ve el contenido original, y solo el primero ve tu dirección IP.



Cada capa solo la puede quitar el relay que comparte su clave con el cliente

El mensaje viaja envuelto en tres capas y cada relay quita la suya.

El cifrado clásico de Tor (llamado *tor1*) usa AES en modo contador. Este modo genera una secuencia de bytes pseudoaleatoria, el *keystream*, y la mezcla con los datos mediante la operación XOR. Esa elección es la que hace posibles las capas.

Fórmula clave. Cada capa es un XOR con el keystream de un relay:

$$C = M \oplus S_{\text{guardia}} \oplus S_{\text{medio}} \oplus S_{\text{salida}}$$

> **En palabras:** el mensaje que sale del cliente es el mensaje original mezclado, uno tras otro, con tres secuencias secretas; cada relay deshace solo la suya y lo que queda sigue siendo ilegible hasta el último.

- `M` es el mensaje original, los datos de la celda.
- `S` es el keystream de cada relay, derivado de la clave que el cliente compartió con él. Solo el cliente y ese relay lo conocen.
- `⊕` es el XOR bit a bit: da 1 cuando los bits son distintos y 0 cuando son iguales. Aplicarlo dos veces con la misma secuencia devuelve el valor original.
- `C` es la celda cifrada que el cliente envía al guardia.

Ejemplo con bits. Sea `M = 1011`, `S` del guardia `= 0110`, del medio `= 1100` y de la salida `= 0011`.

1. El cliente calcula `1011 ⊕ 0110 ⊕ 1100 ⊕ 0011`, que da `C = 0010`.
2. El guardia aplica su secuencia: `0010 ⊕ 0110 = 0100`. No sabe qué es.
3. El relay medio aplica la suya: `0100 ⊕ 1100 = 1000`. Tampoco.

4. El relay de salida aplica la suya: `1000 ⊕ 0011 = 1011`, el mensaje original.

Por qué funciona. XOR es conmutativo y cada secuencia se anula consigo misma, así que da igual el orden en que se quiten las capas. Y como cada relay solo conoce su propio keystream, nadie puede reconstruir el mensaje antes de la salida.

Este esquema tiene un punto débil conocido. Como XOR no autentica nada, un relay malicioso puede alterar bits de una celda sin que se note en ese salto; si otro relay del atacante, más adelante en el circuito, ve el mismo patrón de alteración, ambos pueden concluir que forman parte del mismo circuito. Es el *tagging attack*. Además, tor1 solo añade 4 bytes de control de integridad por celda y usa la misma clave durante toda la vida del circuito. El capítulo 14 explica la sustitución que corrige ambos problemas.

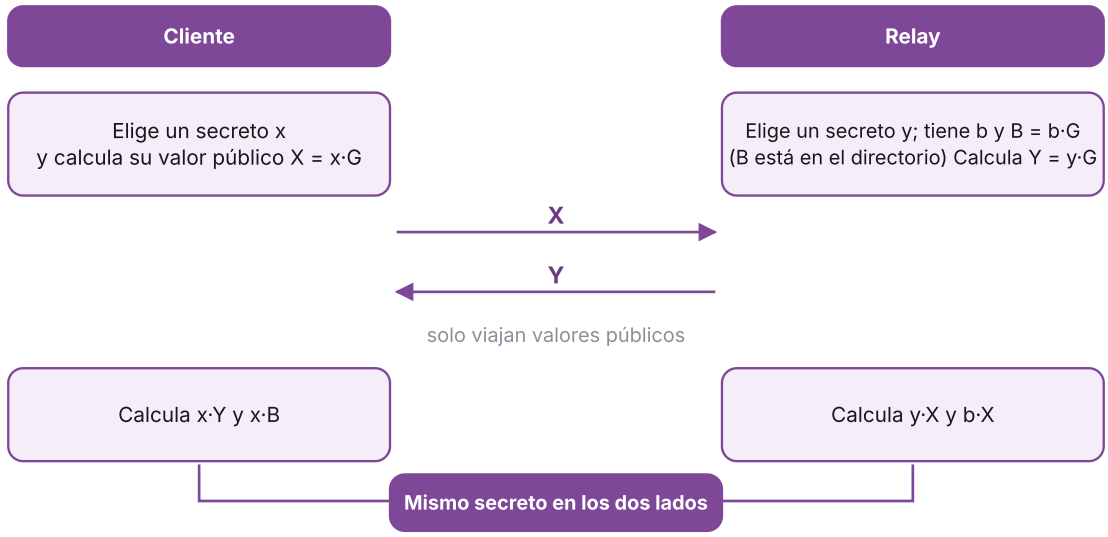
CAPÍTULO 5

El apretón de manos ntor

Queda una pregunta: cómo acuerdan el cliente y un relay una clave secreta si hablan por una red que puede estar vigilada. Tor usa un protocolo llamado **ntor**, basado en el intercambio de claves de Diffie-Hellman sobre la curva Curve25519.

Diffie-Hellman con pinturas

La analogía clásica son las pinturas. Tú y el relay acordáis un color público, por ejemplo amarillo. Cada uno mezcla en secreto su propio color privado con el amarillo y envía al otro la mezcla. Después, cada uno añade su color privado a la mezcla recibida. Los dos acaban con el mismo color final, y quien solo vio las mezclas no puede obtenerlo porque separar colores mezclados es prácticamente imposible.



Cliente y relay intercambian solo valores públicos y obtienen el mismo secreto.

Fórmula clave. Los dos extremos llegan al mismo secreto por caminos distintos:

$$x \cdot Y = x \cdot (y \cdot G) = y \cdot (x \cdot G) = y \cdot X$$

En palabras: si tú combinas tu número secreto con el valor público del relay, y el relay combina su número secreto con tu valor público, ambos obtienen el mismo resultado sin haberlo enviado nunca.

- `G` es un punto fijo y público de la curva, conocido por todos.
- `x` es un número secreto que elige el cliente para esta conexión y descarta después.
- `X = x · G` es el valor público del cliente. Se puede enviar: deducir `x` a partir de `X` es inviable.
- `y` e `Y = y · G` son lo mismo para el relay.
- La operación `·` es la multiplicación de un punto de la curva por un número. Es fácil en un sentido y prácticamente imposible de invertir.

Ejemplo con números pequeños. La curva real trabaja con números enormes, pero la idea se ve con aritmética de reloj de 23 horas y base 5.

1. El cliente elige `x = 6` y envía `5^6 mod 23 = 8`.
2. El relay elige `y = 15` y envía `5^15 mod 23 = 19`.
3. El cliente calcula `19^6 mod 23 = 2`.
4. El relay calcula `8^15 mod 23 = 2`.

Ambos obtienen 2, y quien solo vio el 8 y el 19 tendría que adivinar `x` o `y`.

Por qué funciona. La multiplicación es conmutativa: `x · (y · G)` y `y · (x · G)` son el mismo punto. Y la dificultad de recuperar `x` o `y` a partir de `X` o `Y` (el problema del logaritmo discreto) es lo que protege el secreto.

ntor añade dos cosas. Primero, el relay tiene además una clave larga `b` con valor público `B = b · G`, publicada en el directorio; el cliente también calcula `x · B`, que el relay recupera como `b · X`. Eso demuestra que quien responde es realmente ese relay y no un impostor, porque solo él conoce `b`. Segundo, el secreto resultante, junto con las identidades y los valores públicos, pasa por una función de derivación de claves (HKDF) que produce las claves de cifrado y de verificación del circuito. Existe una variante, ntor v3, que permite al cliente enviar datos adicionales dentro del propio apretón de manos.

CAPÍTULO 6

Directorio y consenso

Para elegir relays, tu cliente necesita una lista fiable de cuáles existen, dónde están y qué claves públicas tienen. Esa lista no puede venir de un único lugar, porque quien la controlara podría llevarte a relays elegidos por él.

La solución es un puñado de **autoridades de directorio**, servidores operados por voluntarios independientes de distintos países (son del orden de una decena). Cada hora, cada autoridad publica su propia opinión sobre el estado de la red y todas las demás la leen. Después votan y producen un documento único, el **consenso**, que solo es válido si lo firma la mayoría de las autoridades. Los clientes llevan incorporadas las claves públicas de las autoridades, así que comprueban la firma y no tienen que fiarse de quien les entrega el documento.

DATO DEL CONSENSO	PARA QUÉ SIRVE
Lista de relays y sus claves	Que el cliente pueda construir circuitos
Banderas (Guard, Exit, Stable, Fast, HSDir)	Decidir qué relays pueden ser guardia, salida o directorio de servicios onion
Ancho de banda medido	Repartir la carga según la capacidad real, medida por las propias autoridades
Parámetros de la red	Ajustar el comportamiento de toda la red a la vez

Un atacante que quisiera falsear la red tendría que controlar a la mayoría de las autoridades, no a una. Es una razón por la que su número reducido y su diversidad son una decisión de diseño, y no una carencia.

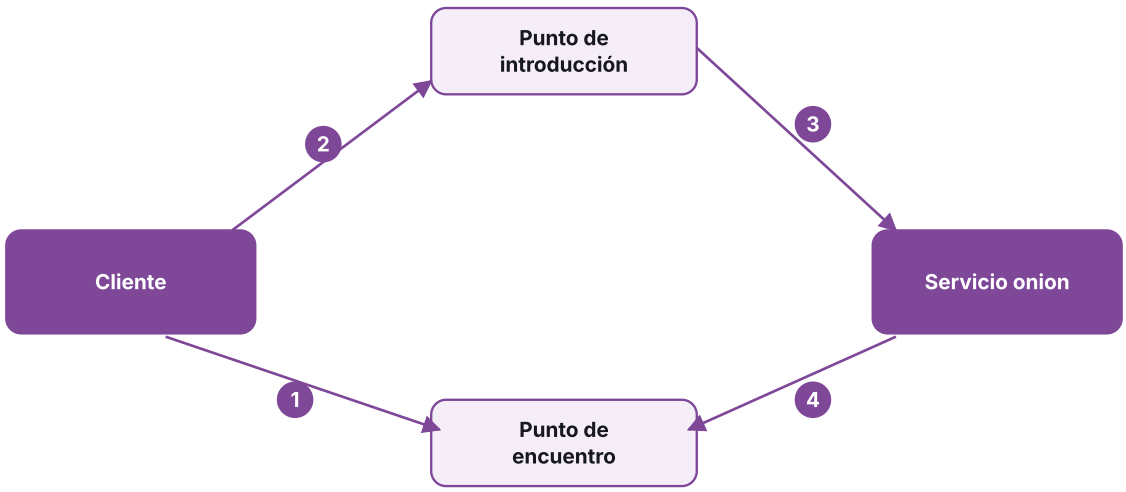
CAPÍTULO 7

Servicios onion

Tor no solo oculta al visitante. Un **servicio onion** es un sitio o servicio cuya ubicación también está oculta: nadie, ni siquiera quien lo visita, sabe en qué servidor está ni dónde. Su dirección acaba en `.onion`, tiene 56 caracteres y es, a la vez, su clave pública. Eso significa que la dirección *es* la identidad del servicio y no hace falta una autoridad externa que certifique que es auténtico.

Quién hace qué

La conexión a un servicio onion se arma con cuatro piezas:



1. El cliente construye un circuito hasta un punto de encuentro.
2. Por otro circuito, pide a un punto de introducción una cita en ese punto.
3. El punto de introducción avisa al servicio, sin saber quién es.
4. El servicio construye su circuito hasta el punto de encuentro; los dos circuitos se unen.

Cada flecha es un circuito de tres saltos; el punto de encuentro une los dos lados.

1. **Puntos de introducción.** El servicio elige unos cuantos relays y construye circuitos hasta ellos. Son sus "buzones": reciben peticiones de conexión sin saber quién es el servicio.
2. **Descriptor.** El servicio publica en unos relays especiales (los HSDir) un documento firmado que dice cuáles son sus puntos de introducción. El descriptor va cifrado con claves derivadas de la propia dirección, así que los HSDir no pueden leerlo.
3. **Punto de encuentro.** El cliente que quiere conectarse elige un relay y construye un circuito hasta él. Es el lugar de cita.
4. **Introducción y encuentro.** El cliente envía a un punto de introducción el mensaje "quédame en este punto de encuentro". El servicio lo recibe, construye su propio circuito hasta ese punto de encuentro y los dos circuitos se unen.

El resultado son seis saltos entre cliente y servicio, tres de cada lado, con el punto de encuentro en medio. Ningún relay conoce a la vez al cliente y al servicio, y el tráfico va cifrado de extremo a extremo con una clave que solo comparten ellos.

La dirección .onion

La dirección v3 no es un hash arbitrario: contiene la clave pública del servicio, un control de errores y un número de versión.

Fórmula secundaria. La dirección se arma así:

$$\text{dirección} = \text{base32}\big(K \,\|\, \text{control} \,\|\, \text{versión}\big)$$

> **En palabras:** la dirección es la clave pública del servicio con un pequeño código de comprobación, para que un error al copiarla se detecte.

Las claves que se publican en el directorio se renuevan cada día a partir de la clave principal mediante un proceso llamado *blinding*, de modo que un relay no pueda seguir a un servicio de un día a otro. Desde 2021 solo funcionan las direcciones v3, mucho más seguras que las antiguas de 16 caracteres.

Frente a los ataques de saturación, los servicios onion disponen desde agosto de 2023 de una defensa de **prueba de trabajo**: en condiciones normales no se nota, pero si el servicio recibe una avalancha de peticiones, los clientes deben resolver un pequeño acertijo de cálculo antes de ser atendidos. Un cliente legítimo tarda de milisegundos a un minuto según el ataque; un atacante que envía miles de peticiones paga ese coste miles de veces.

CAPÍTULO 8

Puentes y transportes enchufables

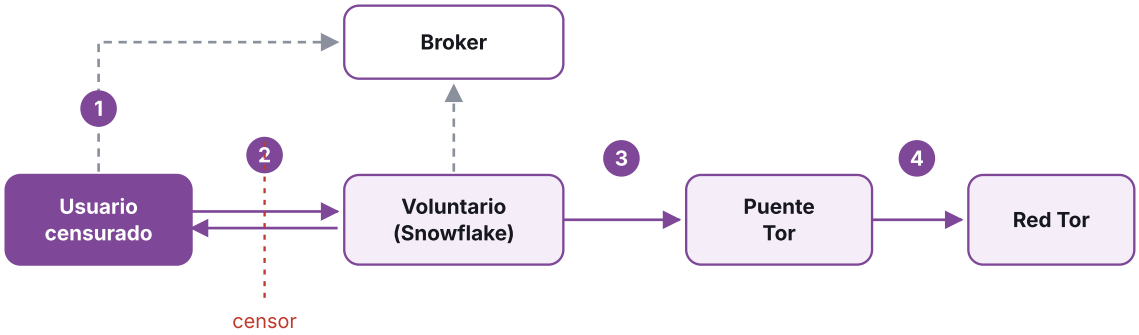
Los relays son públicos, así que un gobierno o una red pueden bloquear Tor simplemente impidiendo conexiones a esas direcciones. Los **puentes** son relays que no figuran en la lista pública. Se piden uno a uno a través de canales como el sitio de puentes del proyecto, y así un censor no puede obtenerlos todos de golpe.

Pero ocultar la dirección no basta: el tráfico de Tor tiene una forma reconocible. Por eso se usan **transportes enchufables**, que disfrazan la conexión:

TRANSPORTE	CÓMO SE DISFRAZA
obfs4	Hace que el tráfico parezca ruido aleatorio sin estructura reconocible
Snowflake	Usa conexiones WebRTC a través de voluntarios que prestan su navegador como puente temporal
WebTunnel	Imita una conexión web HTTPS normal hacia un sitio común

Snowflake: un puente prestado

Con **Snowflake**, cualquier persona puede dejar abierta una pestaña o instalar una extensión y convertirse en un puente efímero para quien esté censurado. Un servicio intermediario, el *broker*, pone en contacto a cada usuario con un voluntario, y la conexión va cifrada entre ambos. El tráfico se parece al de una videollamada, algo que los censores dudan en bloquear porque tiene muchos usos legítimos. Tor Browser incluye un asistente de conexión que, si Tor no se conecta, sugiere puentes adecuados según el país.



1. Usuario y voluntario se encuentran a través del broker.
2. Se conectan por WebRTC, cifrado de extremo a extremo: parece una videollamada.
3. El voluntario reenvía el tráfico a un puente.
4. El puente lo introduce en la red Tor.

Un voluntario presta su navegador como puente efímero; el censor solo ve una videollamada.

Ninguna de estas medidas garantiza que Tor funcione en todos los países: es una carrera continua entre quien bloquea y quien inventa disfraces nuevos.

CAPÍTULO 9

Tor Browser

Usar Tor con un navegador cualquiera no sirve de mucho: tu navegador habitual tiene cookies, extensiones, tipografías y configuración que te identifican aunque la IP sea distinta. **Tor Browser** es una versión de Firefox endurecida y configurada para que todos sus usuarios parezcan

iguales.

MEDIDA	QUÉ EVITA
Todo el tráfico por Tor	Que alguna petición salga con tu IP real
Circuito distinto por sitio	Que dos webs descubran que eres la misma persona
Aislamiento de cookies y datos por sitio	El rastreo entre webs con cookies de terceros
Huella uniforme	Que tu navegador se distinga por pantalla, fuentes o idioma
Ajuste del tamaño de ventana	Que tu tamaño de pantalla funcione como huella
NoScript y niveles de seguridad	Limitar JavaScript y funciones peligrosas
Sin guardar historial ni cookies	Que quede un rastro local al cerrar

Tor Browser ofrece tres niveles de seguridad. En el estándar casi todo funciona, en el más seguro se desactivan funciones que han servido para ataques, a costa de que algunas webs no se vean bien, y el nivel intermedio queda entre ambos. Hay versión para Windows, macOS, Linux y Android. En iOS no existe un Tor Browser oficial; el proyecto recomienda Onion Browser.

La idea central es la **uniformidad**: cuantos más usuarios sean idénticos, más difícil es distinguirte. Por eso no conviene añadir extensiones, cambiar el tamaño de la ventana a mano ni instalar tipografías nuevas: cada cambio te hace más único.

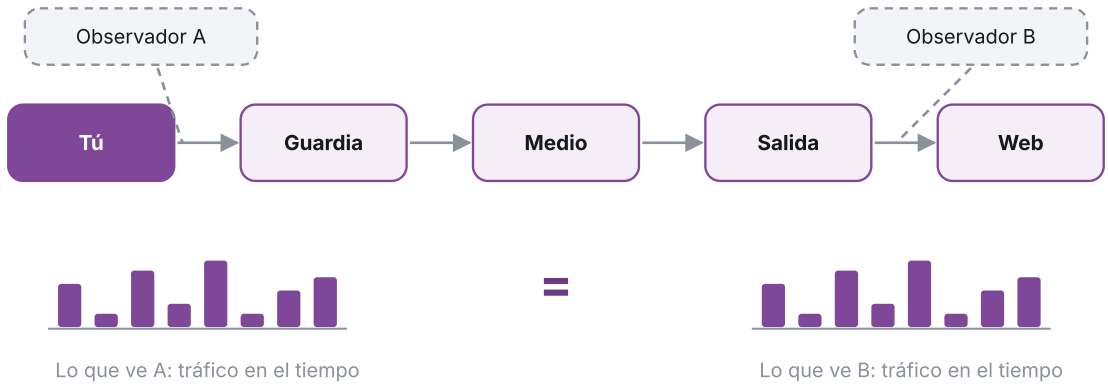
CAPÍTULO 10

Ataques y límites

Tor no es una garantía de anonimato, sino una dificultad considerable para quien quiera rastrearte. Estos son los ataques conocidos.

El límite más conocido: la correlación

Correlación de extremo a extremo. Si un adversario ve el tráfico que entra en la red por tu lado y el que sale por el otro, puede comparar cuándo y cuánto se envía y deducir que son la misma conversación. No necesita romper ningún cifrado. Es el límite teórico más conocido de Tor.



Mismo patrón de tiempos y volumen: misma conversación, sin romper ningún cifrado

Quien ve las dos puntas puede emparejar los patrones de tráfico sin descifrar nada.

Relays maliciosos. Cualquiera puede operar relays, y se han descubierto operadores que controlaban una porción grande de la red con fines poco claros. El proyecto los detecta y los expulsa, y el uso de guardias reduce el riesgo de caer una y otra vez en manos de ellos. Los relays de salida pueden ser maliciosos y alterar o espiar tráfico no cifrado, por eso es importante usar HTTPS.

Ataques a servicios onion. En 2024 se documentó el caso de un usuario de un servicio onion identificado por las autoridades (según la prensa, alemanas). Según el análisis del proyecto, la víctima ejecutaba una versión antigua de un programa de chat sin las protecciones más recientes; el atacante habría localizado su guardia y después cruzado datos de tiempo. El mismo proyecto concluyó que Tor Browser sigue siendo seguro si se actualiza, y que los usuarios con conexiones muy largas son más vulnerables.

Huella digital de sitio web. Un adversario que ve solo tu entrada puede intentar adivinar qué web visitas por la forma de tu tráfico: cuántas celdas, en qué orden. Funciona mejor con webs poco numerosas; las defensas que se estudian son añadir relleno y variar los patrones.

Errores del usuario. La mayor parte de las identificaciones reales no rompieron Tor, sino que aprovecharon descuidos: iniciar sesión con una cuenta personal, usar el mismo alias en varios sitios, abrir un documento descargado fuera del navegador o instalar un complemento.

Fallos del navegador. Tor Browser se basa en Firefox y hereda sus fallos. En el pasado se han explotado vulnerabilidades del navegador para revelar la IP real; por eso es esencial mantenerlo actualizado y, si el riesgo es alto, usar el nivel de seguridad más estricto.

CAPÍTULO 11

Tor, VPN y combinaciones

Tor y una VPN se parecen en que ocultan tu IP, pero resuelven problemas distintos.

	TOR	VPN
En quién confías	En nadie en concreto: el tráfico se reparte entre tres relays	En la empresa que opera la VPN
Quién sabe tu IP y tu destino	Ningún relay a la vez	El proveedor, siempre
Velocidad	Más lenta, por los tres saltos	Casi la de tu conexión
Disimula que usas Tor	Solo con puentes	No aplica
Coste	Gratis	Normalmente de pago

Una VPN protege bien frente a tu proveedor de internet y a redes locales, y mueve la confianza hacia el proveedor de la VPN. Tor distribuye esa confianza, pero es más lento y algunos sitios lo bloquean. No son sustitutos: son herramientas para amenazas diferentes.

Tor sobre VPN. Primero te conectas a la VPN y desde ahí abres Tor. Tu proveedor de internet solo ve la VPN, y el guardia de Tor ve la IP de la VPN en lugar de la tuya. Puede ser útil para ocultar que usas Tor, a costa de que el proveedor de la VPN sepa que lo haces. Un ejemplo es una VPN como Mullvad, que no pide datos personales.

VPN sobre Tor. Conectas a Tor y desde ahí a una VPN. Es poco habitual y compleja de configurar. Sirve en casos concretos, como acceder a un servicio que bloquea las salidas de Tor, pero la VPN te identifica a ti ante su proveedor.

Para la mayoría de las personas, usar solo Tor Browser es lo más sencillo y lo más razonable. Combinar herramientas añade complejidad y cada capa nueva puede introducir errores que revelen más de lo que ocultan.

CAPÍTULO 12

Tails y Whonix

Tor Browser solo protege el navegador. Si abres otro programa, el resto de tu ordenador sigue conectándose directamente. Dos sistemas completos resuelven este problema.

Tails es un sistema operativo que se arranca desde una memoria USB. Todo su tráfico sale por Tor, no deja huella en el disco del ordenador y olvida todo al apagarse, salvo lo que guardes en un almacenamiento cifrado opcional. Sirve para trabajar en equipos prestados o compartidos y para periodistas o activistas con un nivel de riesgo alto.

Whonix funciona con dos máquinas virtuales: una pasarela que dirige todo el tráfico por Tor y una estación de trabajo aislada que solo puede hablar con la pasarela. Si un programa en la estación de trabajo está comprometido, no puede averiguar tu IP real, porque nunca la conoce. Suele usarse dentro de Qubes OS.

	TOR BROWSER	TAILS	WHONIX
Protege	Solo el navegador	Todo el sistema	Todas las aplicaciones de la máquina de trabajo
Persistencia	Normal	Ninguna por defecto	Normal dentro de la máquina virtual
Dificultad	Baja	Media	Alta

CAPÍTULO 13

Buenas prácticas

La herramienta importa menos que el hábito. Estas reglas reducen mucho el riesgo.

1. **Descarga Tor Browser solo del sitio oficial** y mantenlo actualizado. Las actualizaciones corrigen fallos que se explotan de verdad.
2. **No inicies sesión con tu identidad real** en una sesión que quieres mantener anónima, ni reutilices alias entre sitios.
3. **Usa HTTPS siempre** o servicios onion, para que el relay de salida no vea tu contenido.

4. **No cambies la configuración** del navegador ni instales extensiones: cada cambio te hace más reconocible.
5. **No abras archivos descargados** fuera de Tor Browser mientras estés conectado, porque pueden conectarse a internet por su cuenta.
6. **Usa puentes** si tu entorno bloquea Tor o si ocultar que lo usas es importante.
7. **Ajusta el nivel de seguridad** al riesgo: más estricto cuanto más te juegues.
8. **Evita sesiones muy largas** con la misma identidad si el riesgo es alto, y usa "Nueva identidad" para separar actividades.
9. **No mezcles identidades:** lo que haces de forma anónima y lo que haces con tu nombre deben mantenerse separados, también en horarios y hábitos.

Y una regla de sentido común: antes de decidir cuánta protección necesitas, responde a las tres preguntas del modelo de amenazas. Con un adversario poco capaz, Tor Browser basta; con uno muy capaz, hacen falta más medidas y mucho cuidado.

CAPÍTULO 14

Cómo evoluciona el protocolo

Tor sigue cambiando. Estas son las mejoras más importantes de los últimos años.

Counter Galois Onion (CGO). Es el nuevo esquema de cifrado de capas que sustituye a tor1. Corrige los tres puntos débiles del capítulo 4: ahora cada celda lleva un control de integridad de 16 bytes en lugar de 4, las claves evolucionan con cada celda (si alguien consigue una clave, no puede leer lo anterior) y las celdas se encadenan, de modo que alterar una invalida todas las siguientes. Con ello se cierra el ataque de etiquetado, que el propio proyecto consideraba el principal que querían resolver. CGO se estrena en Arti, donde ya se considera listo para producción desde julio de 2026, y llegará al resto de la red cuando relays y clientes lo adopten.

Arti. Es una reescritura de Tor en el lenguaje Rust. Este lenguaje evita de forma estructural clases enteras de errores de memoria, que son una fuente habitual de vulnerabilidades en programas escritos en C. Arti ya incluye control de congestión activado por defecto y avanza hacia poder funcionar como relay y como autoridad de directorio.

Control de congestión y conflux. Tor mejoró su rendimiento con un sistema de control de congestión más fino y con *conflux*, que permite repartir una misma conexión entre dos circuitos para usar mejor el ancho de banda disponible.

Defensas de los servicios onion. La prueba de trabajo contra saturación (capítulo 7) y las protecciones llamadas *vanguards*, que dificultan localizar el guardia de un servicio, son parte de este esfuerzo.

CAPÍTULO 15

Comunidad, financiación y legalidad

Tor lo sostiene The Tor Project, una organización sin ánimo de lucro. Durante años dependió mucho de financiación de agencias gubernamentales, y ha trabajado por diversificarla con donaciones individuales, fundaciones y campañas públicas de recaudación. El código es abierto, la red es operada por voluntarios y cualquiera puede revisar cómo funciona.

Cualquier persona puede aportar sin ser experta: operando un relay o un puente, instalando el complemento de Snowflake, traduciendo documentación, reportando fallos o donando. Los relays de salida son los que implican más riesgo legal para quien los opera, porque el tráfico parece originarse en su dirección IP; los relays medios y de entrada son mucho más sencillos.

Usar Tor es legal en la mayoría de los países. En unos pocos está restringido o bloqueado, y conviene informarse de la normativa local antes de usarlo en un entorno de riesgo. Tor es una herramienta: puede servir tanto para proteger a un periodista como para un uso indebido, igual que un cuchillo, y su diseño no permite distinguir intenciones.

CAPÍTULO 16

Fuentes

- Tor Project, “Tor is still safe”: https://blog.torproject.org/tor-is-still-safe/
- Help Net Security, sobre Counter Galois Onion: https://www.helpnetsecurity.com/2025/11/25/tor-project-cgo-encryption-update/
- CyberInsider, sobre la versión 2.5.0 de Arti: https://cyberinsider.com/tor-releases-arti-2-5-0-with-stable-cgo-encryption-and-security-fixes/

- Tor Project, defensa de prueba de trabajo para servicios onion: https://blog.torproject.org/introducing-proof-of-work-defense-for-onion-services
- Especificaciones de Tor, servicios onion: https://spec.torproject.org/rend-spec/overview.html
- Tor Metrics: https://metrics.torproject.org/
- Tor Project, campaña de recaudación: https://blog.torproject.org/2025-fundraiser-donations-matched/