Tor – The Onion Router

A way too short introduction...

Florian Preinstorfer

12.10.2012

Inhalt

1. Motivation
2. The Onion Router
3. Getting started

The problem

Public networks like the Internet are vulnerable to traffic analysis:

- Packet headers identify sender and recipient
- Packet routes can be tracked
- An adversary can link sender and recipient
- Encryption does **not** hide the link between sender and recipient

Who needs anonymity? I

- Activists
- Journalists
- Law enforcement
- The military
- Corporations
- Censorship resistant publishers
- Socially sensitive communicants (chat rooms for people with illnesses, abuse survivors)

Who needs anonymity? II

- You
  - With whom do you communicate
  - Which websites are you visiting
  - Which books do you read
  - What kind of people do you know
  - Where do you work
  - Where do you live

Who needs anonymity? III

- Criminals
  - They already have it
  - Tor aims to help the others

Anonymity from whom?

- Channel and data anonymity:
  - Recipient of your messages
  - Sender of your message
- Channel anonymity:
  - Passive adversary from outside
  - Network infrastructure
- Tor offers channel anonymity (it does not care about the content)

What is Tor?

From `torproject.org`:

Tor is free software and an open network that helps you defend against a form of network surveillance that threatens personal freedom add privacy, confidential business activities and relationships, and state security known as traffic analysis.

What is Tor?

- An overlay network designed to anonymize TCP-based applications (web, im, ssh, ...)
- A circuit based anonymity network
- A low latency anonymity network

Quick overview

- License: BSD
- Written in C
- Works on Linux, BSD, OSX, Solaris, Win32, Android, ...
- Runs in user space (no kernel modules, no root)

Design goals

- Deployability
- Usability
- Flexibility
- Simple protocol design

What Tor is not...

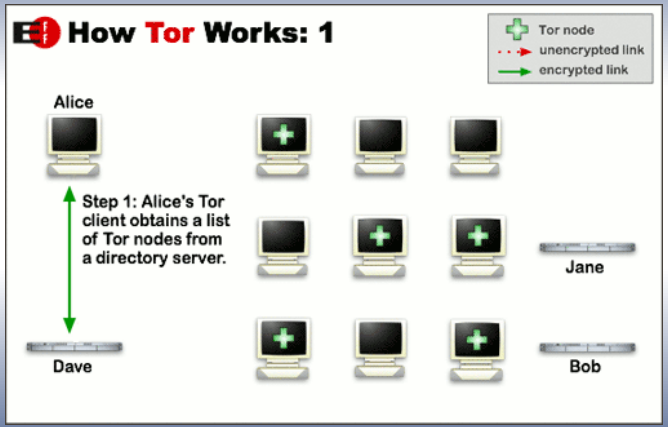
- A P2P anonymity network
- A protocol normalizer
- A steganographic tool
- Secure against end-to-end attacks

Threat model

Most anonymity systems try to cope with a global passive adversary. Tor does **not**. The Tor threat model tries to withstand the following attacks against the network:

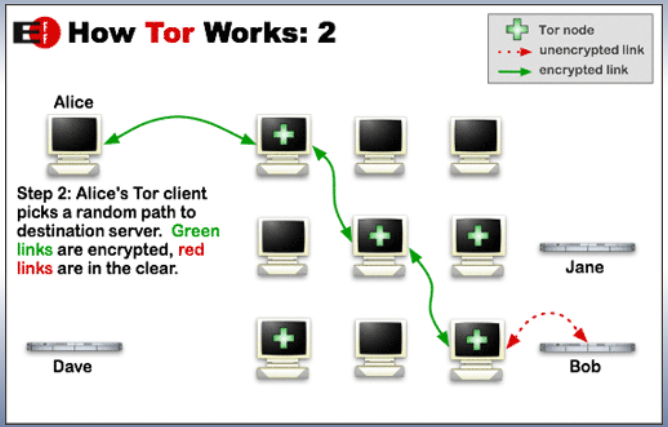
- An adversary can ...
  - observe some fractions of the network
  - generate, modify, delete, delay traffic
  - can operate bogus onion routers
  - can compromise some existing onion routers

How does it work (simple)?



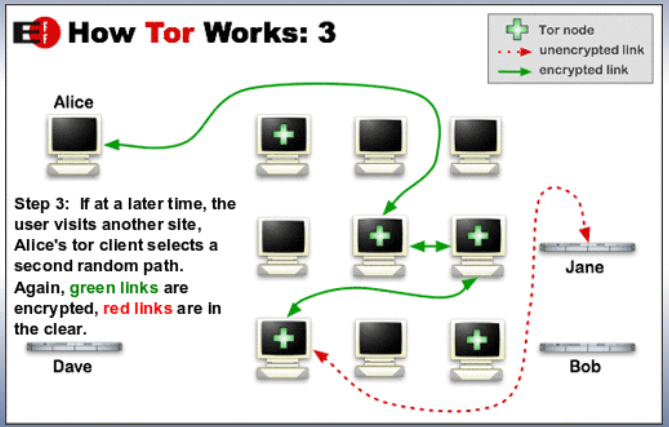
Source: `http://torproject.org`

How does it work (simple)?



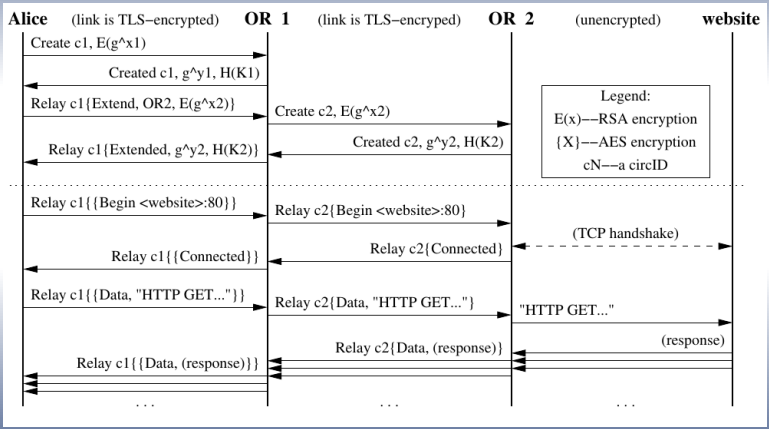
Source: `http://torproject.org`

How does it work (simple)?



Source: `http://torproject.org`

How does it work (extended)?



Source: `http://torproject.org`

A few features I

- Directory servers
  - Maintain a list of onion routers (locations, exit policies, keys)
  - Their addresses are hardcoded in the source code
  - Important for bootstrapping
  - Caching is possible (reduce load, bottleneck)

A few features II

- Circuit multiplexing
  - Tor builds circuits in the background (expensive)
  - Multiple TCP streams share a circuit (cheap)
  - Circuit rotation is important

A few features III

- Bandwidth rate limiting
  - Token bucket: limits average bandwidth, but permits bursts)
  - Throttling: controls congestion on streams

A few features IV

- Configurable exit policies
  - If you want to run a tor node, you can decide which services you allow and which not
  - Helps to reduce the abuse of Tor
  - From the design paper: *„Exit policies reduce the possibilities for abuse, but ultimately the network requires volunteers who can tolerate some political heat.“*

A few features V

- Hidden services
  - Hide the location of a service
  - Uses „rendezvous points“ to establish the connection between user and hidden service
  - Usage: censorship resistant publishing, anonymous publishing, ...

Tor browser bundles

- Provides you Tor and a preconfigured browser
- Click to play (no installation required)
- Available for all major platforms
- `https://www.torproject.org/download`

Smartphones

- Tor for Android: Orbot
- `https://play.google.com/store/apps/details?id=org.torproject.android`
- Tor for iOS: Onion Browser
- `http://v3.mike.tig.as/onionbrowser`

Linux distributions

- Almost all of them provide Tor packages
- Use $packagemanager
- Do not forget to start the Tor daemon

SOCKS Proxy

- Tor provides a SOCKS5 proxy at **localhost:9050**
- Configure your applications to use above proxy address

Application layer proxies

- Tor can also be used with application layer proxies
- HTTP: privoxy, polipo
- Setup: $browser $\leftrightarrow$ $proxy $\leftrightarrow$ Tor

Am i using Tor?

- Check at: `https://check.torproject.org`

References

- The Tor design paper
- Tor: An anonymizing overlay network for TCP
- `http://torproject.org`

Tor – The Onion Router

A way too short introduction...

Florian Preinstorfer

12.10.2012