TOR NETWORKS

Computer architecture | University Of Hradec Králové

Anonymity, Security, Protocol and Weaknesses

Ivan Lopez Justicia

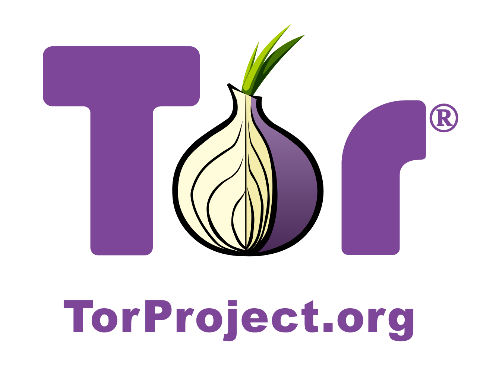
2019

**INDEX**

1. Origin 2
   1. What is?
   2. First versions and developers
   3. Evolution
2. Design 3
   1. TOR Circuit
   2. TOR Circuit Components

* Entry/Guard Relay
* Middle Relay
* Exit Relay
* Bridge Nodes

1. Working behaviour 5
   1. You as user
   2. The user network
   3. The request compiler
   4. Practical example
2. Applications and use 8
3. TOR vs Others anonymous methods 9
   1. VPN vs Tor
   2. Multi-hop VPN services
4. Security 10
   1. Onion Routing
   2. How Onion routing works
   3. Example
5. Debilities 13
6. Webgraphy 14
7. **Origin**
   1. What is?

The Onion Router, from which the acronym TOR comes from, was primarily developed for the use by the US military, predominantly Navy, as to mask their IP addresses which could lead to theft of sensitive data while gathering information on missions. Once the military passed to in-house high anonymity VPN systems, TOR was released as an open source free software, that is completely legal to use and possible to download from the TOR website.

The Tor network disguises your identity by moving your traffic across different Tor servers, and encrypting that traffic so it isn't traced back to you. Anyone who tries would see traffic coming from random nodes on the Tor network, rather than your computer.

* 1. First versions and developers

In the 90’s, the lack of security in Internet and its capacity for monitoring and surveillance became evident. In 1995, David Goldschlag, Mike Reed and Paul Syverson, from US Naval Research Lab (NRL), created and deployed the first research designs and onion routing prototypes.

The main target of onion routing was to achieve a way to surf in Internet with the most possible privacy and the idea was routing the traffic through several servers and encrypt it in each step of the path.

* 1. Evolution

In the beginning of 2000’s, Roger Dingledine started to work in an onion routing project with Paul Syverson. To differentiate this project from the others attempts of onion routing project which were appearing, this project was called TOR (The Onion Routing). Time after Nick Mathewson join to the project.

In the early 90’s, the approach of onion routing was based in a decentralized network. This network should be operated by entities with diverse interest and assumed trust and the software should be free and open to maximize the transparency. That’s why in October 2002, after the initial implementation of the network, it was published open software.

In 2004, Electronic Frontier Foundation (EFF) started to finance the project. In 2006, Tor Project, Inc., was founded, a non-profit organization, to keep the development of TOR.

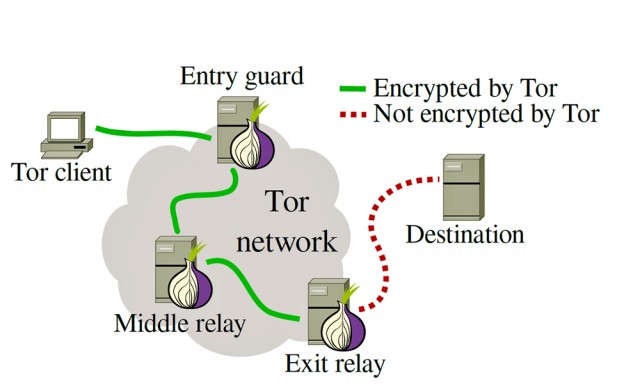
In 2007, the organization began to build bridges with the Tor network to face censorship, such as the need to bypass government firewalls, and allow its users to access the open web.

Thanks to the success that TOR obtained, in 2008 the development of TOR Browser was started. The development of a browser which facilitated access to the network made TOR more famous between users and Internet activists.

1. **Design**
   1. TOR circuit

The anonymity of TOR works by transmitting data though a TOR circuit which is a key component of TOR that appends to onion routing mechanism.

The circuit is a combination of entry or guard relay, middle relay and exit relay. Also exists Bridge nodes which are involved for some transmissions (this occurrences are only observed when the known entry and exit nodes are blocked by a certain party like a government of a corporate organization).



* 1. TOR Circuit components
* Entry/Guard relay

It is the entry point to the TOR Network. Each client that wants to connect to the TOR network will first connect to the guard node meaning, they can see the real IP Address of the client who is attempting to connect. The list of guard nodes is available in the public list of TOR nodes and are updated almost every minute.

There are cases where attackers have control or observe certain relays and they can be used to see the victim’s browsing. However, such chances are also nullified because the TOR guard nodes are changed for every hop.

* Middle Relay

Middle relays cover most part of the Tor circuit in any given transmission. They consist of relays through which data is passed in encrypted format and no node knows more than its predecessor and descendant. All the available middle relay nodes show themselves to the guard and exit nodes so that any may connect to them for transmission. Even if any middle relay is known to transmit malicious traffic (such as attacker’s exploit or the attack itself) they’re not held responsible as they’re neither the source nor destination of the traffic. A middle relay will never be allowed to act as an exit node. It is most suitable for users who want to utilize TOR from home or workplace (if it’s allowed).

* Exit Relay

The exit relay is the final relay in the TOR circuit. They are the nodes that send the data to the destination and are often considered the culprit because the Exit node is perceived as the origin of the traffic. Therefore, the exit node’s IP will be directly visible to the destination and often receive multiple complaints, legal notices, take down notices etc. In order to host an End node one must be ready to handle problems such as, Legal issues like take-downs or DMCA notices, Own a dedicated IP and make sure their reverse DNS is easily discovered, setting up an Exit Node Hosting notice (the most important step) etc.

* Bridge Node

Bridge nodes are the nodes which are not listed on the public directory of TOR nodes. Most of the entry and exit nodes are publicly available on the internet and therefore they can be blocked if one wishes to restrict the usage of TOR. Many ISPs, Corporate Organizations and even Governments have filters set to ban the usage of TOR. For example, the Chinese government has blocked all publicly available nodes on their country level firewall. To avoid such a scenario, there are Bridge nodes. You will need to follow a different configuration settings in order to connect to the TOR network via a Bridge node (which shall be discussed in the upcoming articles).

1. Working Behaviour

Behind the Tor project, there are three main parts: **you as user, the user network,** and the **request compiler.** We can focus this as the steps between you and the data you are trying to access being your IP address masked. Part of the idea of TOR is that any user can be any of those steps, making it impossiblo to pinpoint where the information is going and from whom if you are on the outside.

* You as user

The TOR Browser is the main part of the TOR’s interface which consists of a modified Mozilla Firefox ESR browser, TOR Launcher, the TOR Button, NoScript, HTTPS EFE and the TOR proxy. This is the whole program that installs all the parts beneath the browser automatically. Once you send your request, TOR browser will encrypt it and send it through.

After knowing for what TOR is used for, the feature that TOR has is that we can install and use it from an external device, in this way, the browser doesn’t interact with the system files of the computer.

When TOR is closed, all the data, cookies and history is deleted from the device and the network, so there are no traces of the IP address which was accessing to which information.

The performance of the network can make you not only to be a usual user who is searching information, but also as proxy for other people, who are using your IP address as a network node to mask their IP address. As we can see, all the users contribute at the same to the anonymity and security of the network.

This fact not only provide benefits, also it is one the biggest risk factors of the project, but as the information request is encrypted and fragmented makes TOR browser capable of fulfilling its primary role of masking your IP address.

* The User Network

The TOR project use its users instead of dedicated servers for conduce the traffic, which goes through the users, more than 3 million online at the same time. This is a very ingenious system as it removes the main weak points of a proxy connection and relies only on Peer-to-Peer (P2P) connections to mask the IP addresses.

It is also good to point out that TOR doesn’t actually hide your IP address from the internet or negates the connection that you have with your ISP, but rather creates a flooding effect where people who are on any point in the connection can’t be certain about the origin of the IP address that they are getting the request.

When you send a request to a webpage or other service or application online, it will first bounce through several random connections worldwide before reaching the destination. In this way, in case the destination has any proxy scanning software, it will only find one of the computers in the network, which wouldn’t even have the connection logged, making the search for your personal data futile.

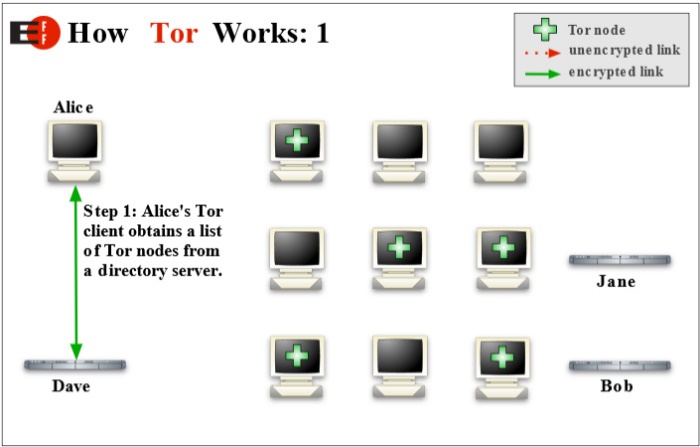
* The Request Compiler

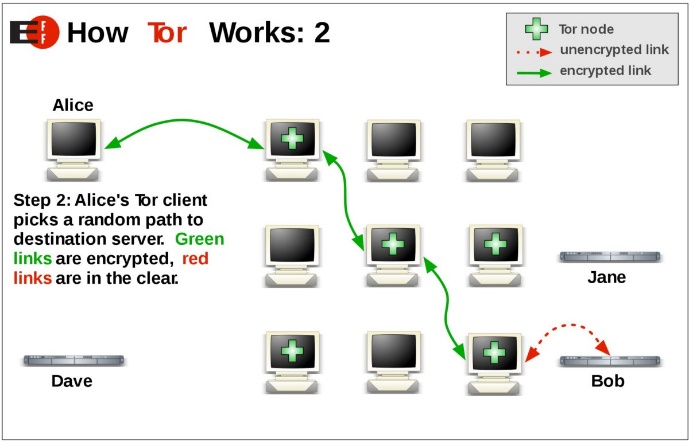
This is the server or computer that has the information that you are requesting and after it receives such request originated from your computer sends it back through the last user that was online, again randomly jumping through TOR users, finally reaching you at your browser.

This way make impossible to ping back all the IP addressed back to you, preventing some more tech-savvy companies from finding your address. TOR acts more like a VPN service that entirely shields the user from scanning software.

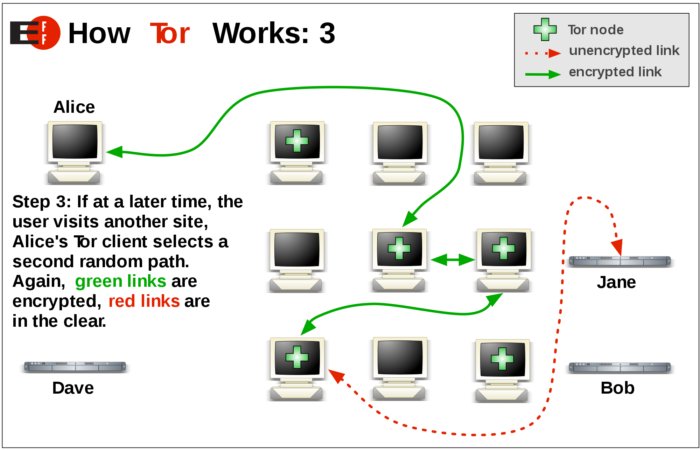
* Practical example

Step 1:

As a result, don't be surprised if Google or another service greets you in a foreign tongue. These services look at your IP address and guesstimate your country and language, but when using Tor, you will often appear to be in a physical location halfway around the world.

 Step 2:

If you live in a regime that blocks Tor or need to access a web service that blocks Tor, you can also configure Tor Browser to use bridges. Unlike Tor's entry and exit nodes, bridge IP addresses are not publicly listed, making it difficult for web services, or governments, to blacklist those IP addresses.

 Step 3:

The Tor network routes TCP traffic of all kinds but is optimized for web browsing. Tor does not support UDP, so don't try to torrent free software ISOs, as it won't work.

1. Applications and use

The applications of this networks are several. Despite in essence the main purpose is to keep privacy for protect the users data, it is TOR’s bread and butter.

The people who use it is really varied:

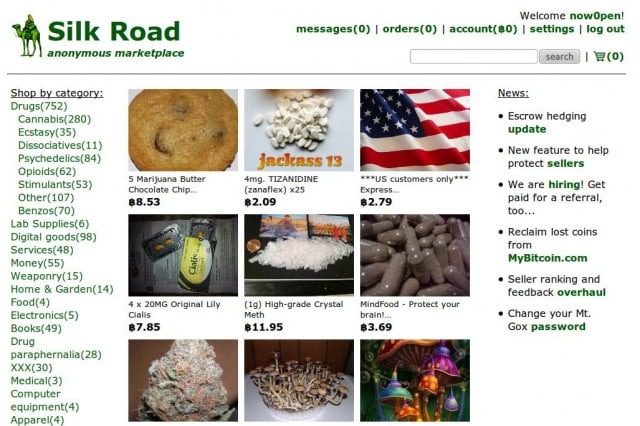
* Normal people: To protect their privacy from unscrupulous marketers and identity thieves, their communications from irresponsible corporations, their children online, to research sensitive topics…
* Journalists and their audience: [Reporters without borders](https://rsf.org/) advise them to use TOR, citizen journalists in China, citizens and journalists in Internet black holes…
* Law enforcement officers: Online surveillance, sting operations, truly anonymous tip lines...
* Activists and whistle-blowers: Human rights activists use Tor to anonymously report abuses from danger zones.
* Others.

The fact of provide anonymous browsing and routing, it also provides a way to host a website or service online anonymously. Tor hidden services are websites that have the .onion top-level-domain (TLD).

For example: htttp://s34s4txr3afds4cs62hpip.onion/[[1]](#footnote-1)

Most TOR hidden services are not human readable because they are generated using a cryptographic algorithm. It is possible using a lot of CPU power to create somewhat custom named hidden services.

Even Facebook has their own Tor hidden service which you can access using a Tor browser at: https://facebookcorewwwi.onion/

 Many websites of the Internet (around 80 percent) comprises pages unknown to most people, locked behind passwords and protocols. Tor allows web pages, like clients, to protect their anonymity, by configuring a server to connect with clients at a Tor relay in between. The server does not need to provide the IP address, and the user does not need it, instead using an “onion address,” a 16 character code that clients enter in place of a traditional URL. The hidden pages on the Tor network comprise one of the most famous darknets, networks only accessible through specific protocols. A phrase like darknet conjures up images of shady dealings, and not without cause; some of the most notable hidden sites are used for trafficking illegal goods, such as the Silk Road, a popular site for selling drugs which was shut down by the FBI in 2013.

1. TOR vs Others anonymous methods
   1. VPN vs Tor

For most people seeking online privacy and security, a VPN is the go-to solution. While there are many shady VPNs that should be avoided, particularly with free VPN services, there are also a few providers that stand out from the crowd.

A VPN will encrypt all traffic between your device (computer, tablet, router, smartphone etc.) and a VPN server. This makes traffic (your online activities) unreadable to third parties, such as your internet provider, hackers, and any other snoopers. Most VPNs have servers all over the world, particularly larger providers. And unlike with Tor, when you use a VPN, you can manually select the server you want and the geographic location you’d like to appear in.

In general, a good VPN service offers some advantages:

* Much better performance than Tor.
* System-wide encryption.
* More versatility.
* Supported on many devices and operating systems.
* More trust.
* Many users.

In contrast to TOR, the most of this services route traffic over a single VPN server using very strong encryption. For the majority of users, it provides enough security and anonymity.

* 1. Multi-hop VPN services

Similar to the Tor network, there are also a few VPN services that will route traffic across multiple VPN servers, or “hops” in the network. When a VPN routes traffic over two or more servers, this is often called a “cascade” or multi-hop VPN setup. We’ll take a look at a few of these below.

**Double-hop VPNs:** Your traffic is being encrypted over two separated servers before exiting to the destination.

**Self-configurable multi-hop VPNs**: With self-configurable VPN services, you can build your own unique cascade by choosing the servers you want.

1. Security
   1. Onion routing

Onion routing is a technique for anonymous communication over a computer network. In an onion network, messages are encapsulated in layers of encryption, analogous to layers of an onion.

One way to understand onion routing is to start with the concept of proxy servers. A proxy server is a server that relays your connection through that server, which basically adds a step in the path of your data packets. If someone traced your IP address, they’d see it as the proxy server’s IP address instead of your home address.

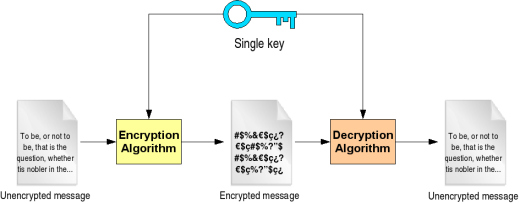
But proxy servers aren’t exactly anonymous. They keep logs of all the traffic that passes through, which means that they can actually point back to you if necessary. For most activities, the proxy server is fine even though it’ll add a bit of latency to your connection. Your anonymity would not be entirely protected, however, if your proxy service was hit with a subpoena for your IP information.

Onion routing is like an advanced form of proxy routing. Instead of routing through a single unprotected server, it uses a network of nodes that constantly encrypt your data packets at every step. Only at the end of this “chain” of onion nodes does your data become decrypted and sent to the final destination. In fact, only this “exit node” has the power to decrypt your message, so no other node can even see what you’re sending.

* 1. How onion routing works

In a usual browser, you request webpages by making simple GET requests to servers without any intermediary. It is just a single connection between a client and a server and someone sniffing on your network can know which server your computer is contacting.

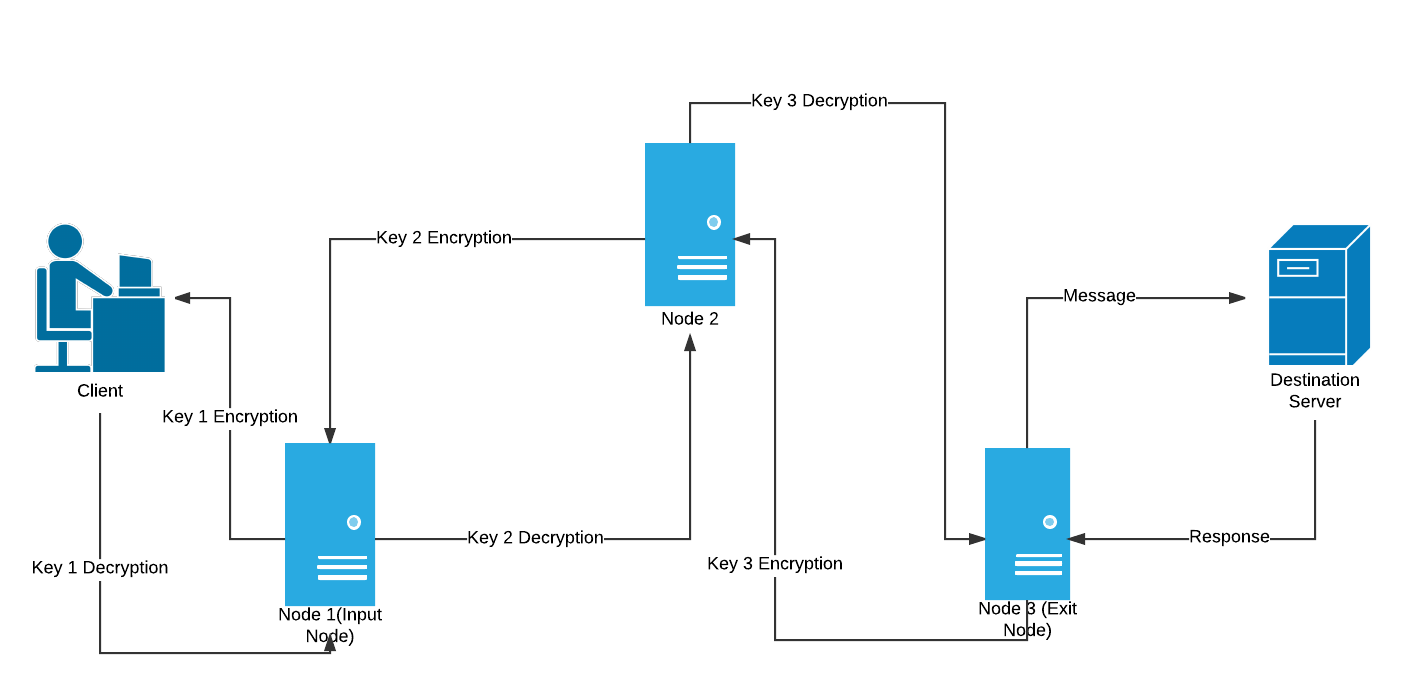
Onion routing does this differently: the connection is maintained between different nodes i.e. the connection hops from one server to another and when it reaches the last server on this circuit it is the server that we wanted to contact and it will process our request and serves us the desired webpage which is sent back to us.



The name of onion routing come from the fact that the message which is sent and the responses are encrypted with different keys, with a unique key for encryption for every different hop or server visit. The client has access to all the keys but the servers only have access to the keys specific for encryption/decryption to that server.

Since this process wraps your message under layers of encryption which have to be peeled off at each different hop just like an onion that’s why it is called an onion router.

* 1. Onion routing example



1. The client with access to all the encryption keys i.e. key 1, key 2 & key 3 encrypts the message(get request) thrice wrapping it under 3 layers like an onion which have to be peeled one at a time.
2. This triple encrypted message is then sent to the first server i.e. Node 1(Input Node).
3. Node 1 only has the address of Node 2 and Key 1. So it decrypts the message using Key 1 and realises that it doesn’t make any sense since it still has 2 layers of encryption so it passes it on to Node 2
4. Node 2 has Key 2 and the addresses of the input & exit nodes. So it decrypts the message using Key 2 realises that it is still encrypted and passes it onto the exit node.
5. Node 3 (exit node) peels of the last layer of encryption and finds a GET request for youtube.com and passes it onto the destination server.
6. The server processes the request and serves up the desired webpage as a response.
7. The response passes through the same nodes in the reverse direction where each node puts on a layer of encryption using their specific key.
8. It finally reaches the client in the form of a triple encrypted response which can be decrypted since the client has access to all the key.
9. Debilities

If someone is listening in on a server at the same time and the matches the request at the destination to a request made by a client on the other side of a network by analysing the length and the frequency of the characters found in the intercepted request or response at the destination server and using that to match with a same request made by a client a fraction of a second (time-stamps on requests and responses can also be helpful in deducing that) and then tracking them down and knowing their online activity and shattering the idea of anonymity.

It is possible to monitor traffic from a Tor exit node (the last hop in the Tor network) and the destination server because that traffic is not encrypted. If you were, for example, to send personally identifiable information through a Tor exit node and someone was monitoring that exit node, your identity would be revealed.

Other way that try to expose the identity of Tor users have targeted the Tor browser and vulnerabilities in the browser. If an attack is able to get the browser or user’s workstation to execute code, they can have that browser connect directly to a server on the Internet, bypassing the Tor network, which would expose that user’s identity.

TOR browsing involves usage of two types of ports i.e. ORPort and DirPort. ORPorts are usually used to make connections and transmissions where as DirPorts are used to fetch updates from the directory servers. The ORPorts usually include ports 80 & 443 but can also be changed with advanced proxy settings while DirPorts include port 9001 and port 9003. Firewall and IDS filters can be configured to monitor any traffic going towards or coming from the ports 9001 and 9003.

Webgraphy

* <https://2019.www.torproject.org/about/overview.html.en>
* <https://www.le-vpn.com/what-is-tor-browser/>
* <https://www.torproject.org/es/about/history/>
* <https://www.researchgate.net/publication/313951935_Detecting_and_blocking_onion_router_traffic_using_deep_packet_inspection>
* <https://restoreprivacy.com/tor/>
* <https://www.youtube.com/user/Computerphile>
* <https://www.makeuseof.com/tag/what-is-onion-routing-exactly-makeuseof-explains/>
* <https://www.researchgate.net/publication/313951935_Detecting_and_blocking_onion_router_traffic_using_deep_packet_inspection>
* <https://www.csoonline.com/article/3287653/what-is-the-tor-browser-how-it-works-and-how-it-can-help-you-protect-your-identity-online.html>
* <https://gizmodo.com/tor-the-anonymous-internet-and-if-its-right-for-you-1222400823>
* <https://www.netresec.com/?page=Blog&month=2013-04&post=Detecting-TOR-Communication-in-Network-Traffic>
* <https://www.rsreese.com/detecting-tor-traffic-with-bro-network-traffic-analyzer/>
* <https://www.wordfence.com/learn/the-tor-network-faq/>
* <https://medium.com/coinmonks/tor-nodes-explained-580808c29e2d>
* <https://lifehacker.com/what-is-tor-and-should-i-use-it-1527891029>
* <https://es.wikipedia.org/wiki/Tor_(red_de_anonimato)>
* <https://www.digitaltrends.com/computing/a-beginners-guide-to-tor-how-to-navigate-through-the-underground-internet/>
* <https://rsf.org/>

1. This url has been checked but in case it coincide with any malicious website, I’m not responsible, it was generated randomly. [↑](#footnote-ref-1)