



# DARKWEB + PYTHON: DISCOVER, ANALYZE AND EXTRACT INFORMATION FROM HIDDEN SERVICES

**@jmortegac** May, 2019



# About me



<http://jmortega.github.io/>

**PYTHON & JAVA & DOCKER SECURITY CONFERENCES**

MUSIC BOX PREVIOUS SITE TALKS **CONFERENCES** Java

Python Ireland Testing python security [YouTube] [PYCONIE]

All Day DevOps Common Vulnerabilities & Exposures (CVE) In Docker Containers [YouTube] [Sonatype] [ALLDAYDEVOPS]

Python & OSINT para proyectos de seguridad [YouTube] [BITUP]

**TESTING DOCKER IMAGES SECURITY** **DISCOVERING PYTHON SEARCH ENGINE** **TESTING DOCKER IMAGES SECURITY** **OSINT TOOLS FOR SECURITY AUDITING**

Testing python security

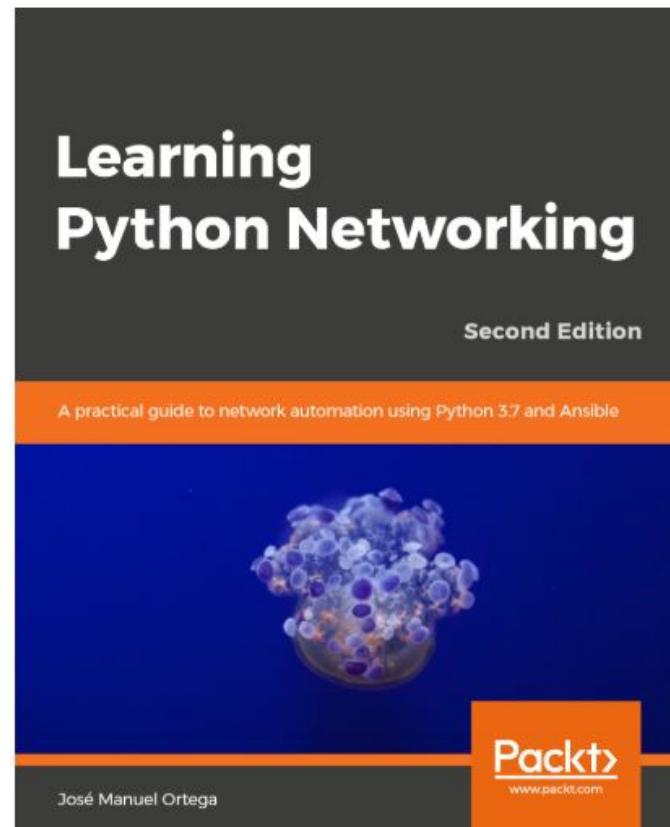
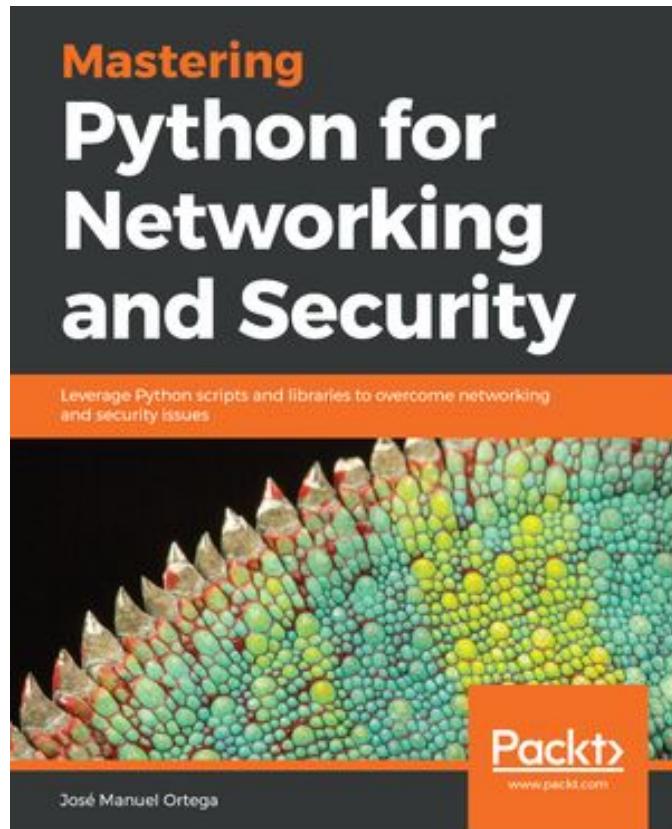
## Testing python security

by Jose Manuel Ortega

Pycones 2018 1 @jmortega

Pycones 2018 MÁLAGA

# About me



# Agenda



- **Introduction to Tor project and hidden services**
- **Discovering hidden services**
- **Modules and packages we can use in python for connecting with Tor network**
- **Tools that allow search hidden services and automate the crawling process in Tor network**

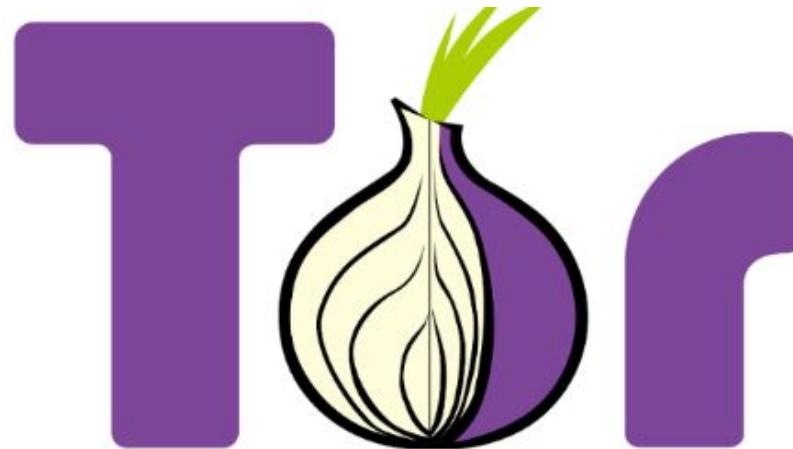
# Surface vs Deep vs Dark Web



# What is Tor?



- Tor is a free tool that allows people to use the internet anonymously.
- Tor anonymizes the origin of your traffic



The Onion Router

# What is Tor?



 WikiLeaks

Leaks News About Partners

Search

## WikiLeaks:Tor

The following method requires some technical ability. If you are used to installing new software and configuring proxy servers you should have the required skills, otherwise you may wish to use one of our [other submission methods](#). Don't let the technology defeat you!

Tor, or The Onion Router, is a cryptographic technique first implemented by US Navy research to permit intelligence agents to use the internet without being traced, by encrypting and routing communications through many different internet servers. Subsequently, Tor has been developed by the US university [MIT](#) and by the California internet rights watchdog the [Electronic Frontier Foundation](#) and subsequently incorporated into WikiLeaks.

Using our anonymous access package ([see below](#)) you can prevent internet spies knowing that your computer has connected to WikiLeaks.

Most Wikileakers do not need this extra security, and there are simpler and possibly safer alternatives for once-off high-risk leaks ([see Submissions](#)). But for those who are at risk and want to access WikiLeaks from the comfort of their homes or offices or need to bypass Internet censorship, Tor (Onion Routing) is an excellent solution.

When you have installed our Tor access package ([see below](#)), you may then connect to WikiLeaks via our anonymous address (the ".onion" is short for "Onion Routing", but you do not need to be concerned with this detail). *NB: the original .onion link for browsing WikiLeaks is currently unavailable; however, if you have installed Tor and are redirecting all of your browsing through the Tor network, you can still browse the normal WikiLeaks site with a high degree of anonymity (but not end-to-end encryption). The secure .onion address provided here and on the [submissions page](#) should still work, in any case.*

To upload a document anonymously using Tor:

<http://suw74isz7wqzpmgu.onion/>  
(this link will only work once you have installed and configured Tor.)

# What is Tor?



## The Advantages of TOR:

1. Tor is considered a strong security tool against surveillance
2. It's the low latency anonymity network
3. Unlike Mix-Networks, the message doesn't have to wait for more messages

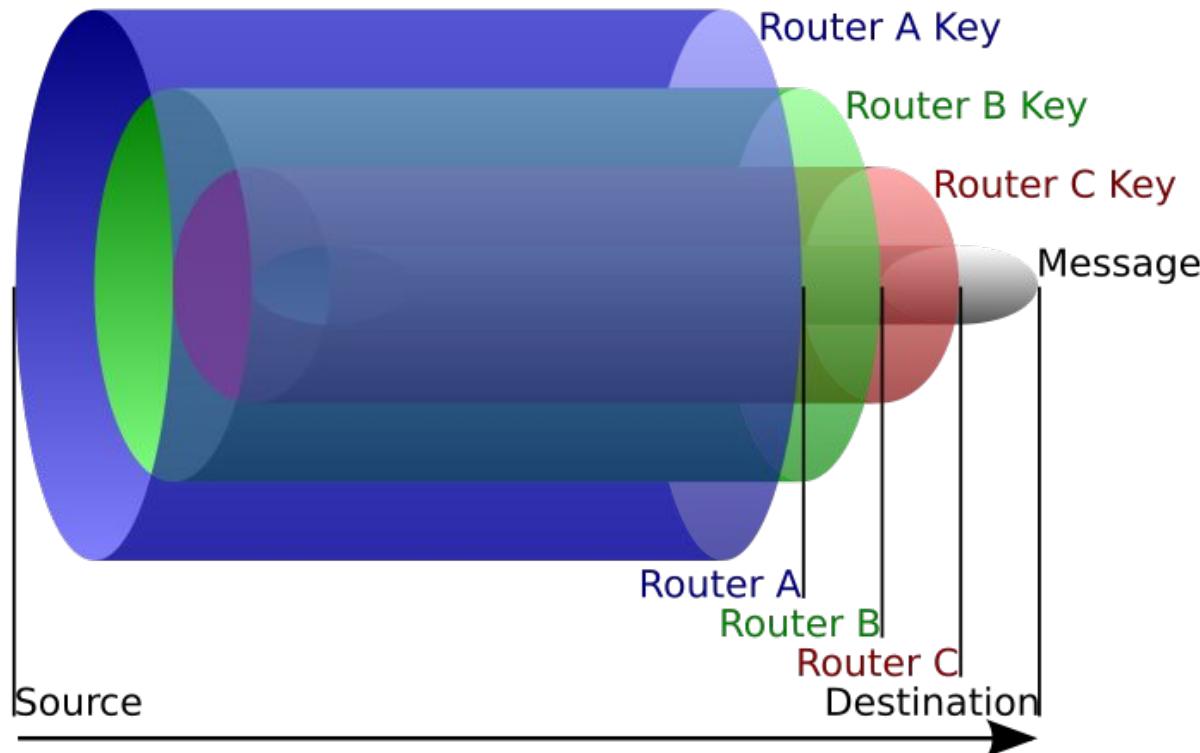
## The Disadvantages of TOR:

1. It isn't safe against end-to-end correlation (attacks at the boundaries of Tor network) and manual traffic analysis.
2. It's slow as the message is routed many times
3. Some ISP tries to search and block Tor relays

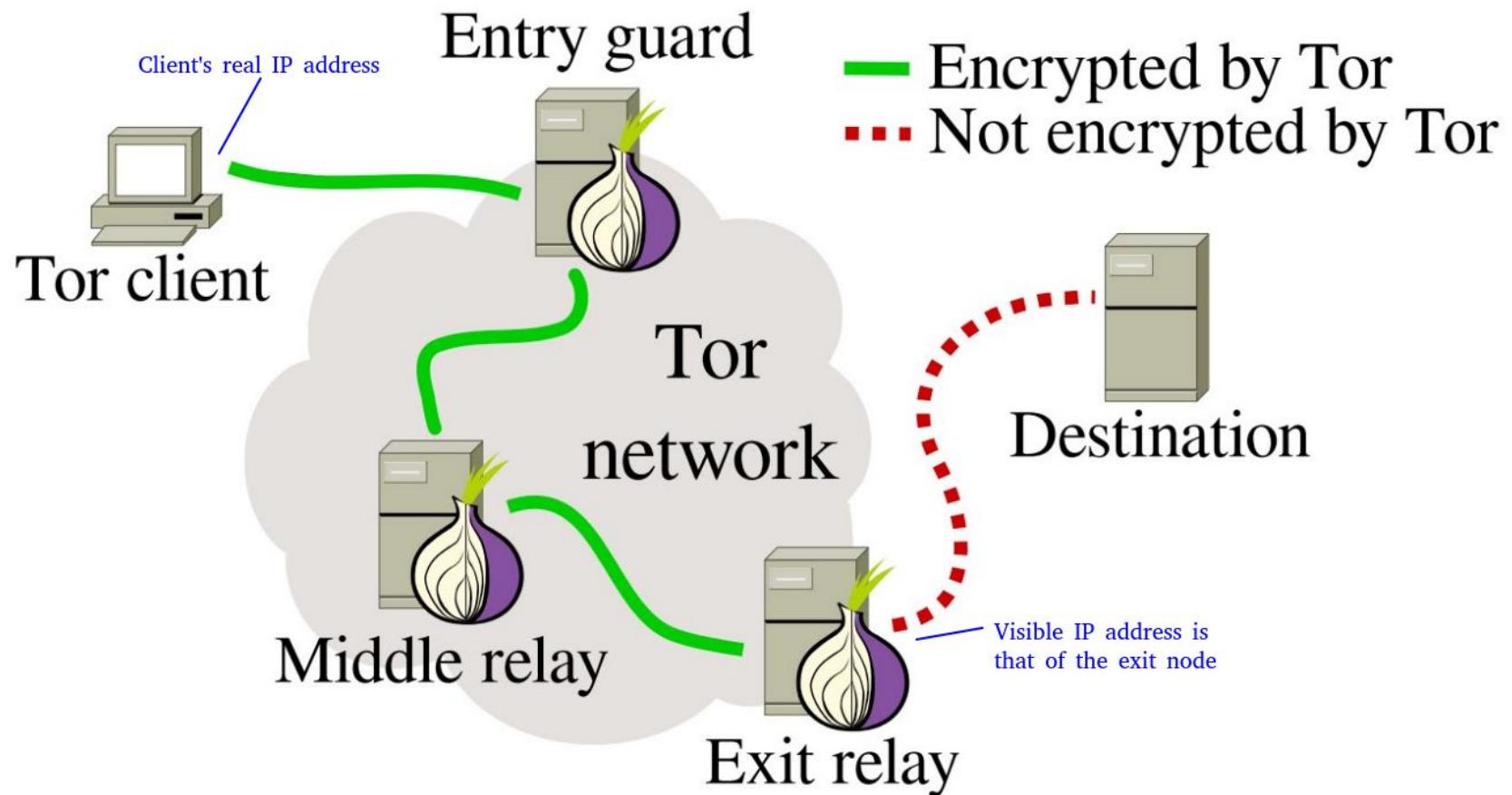
# Onion Routing



**Tor is based on Onion Routing, a technique for anonymous communication over a computer network.**

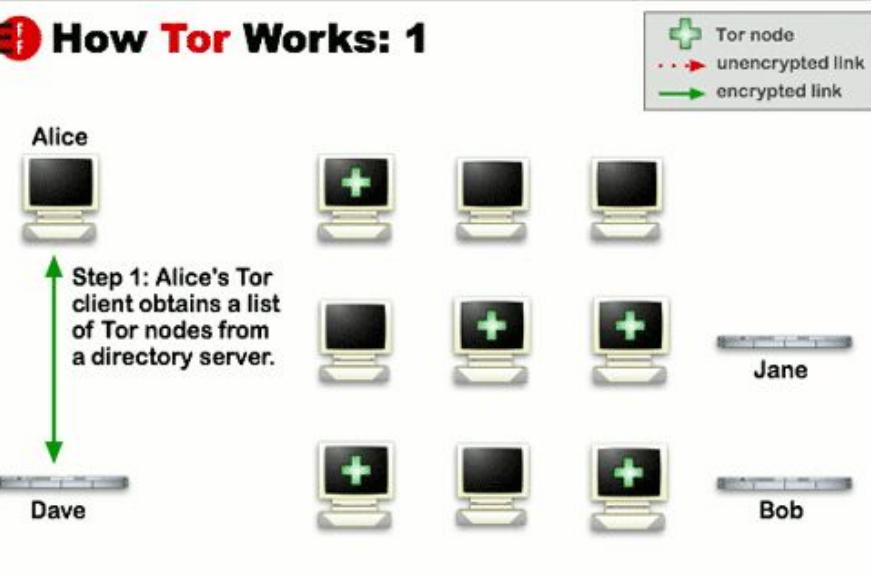


# Onion Routing



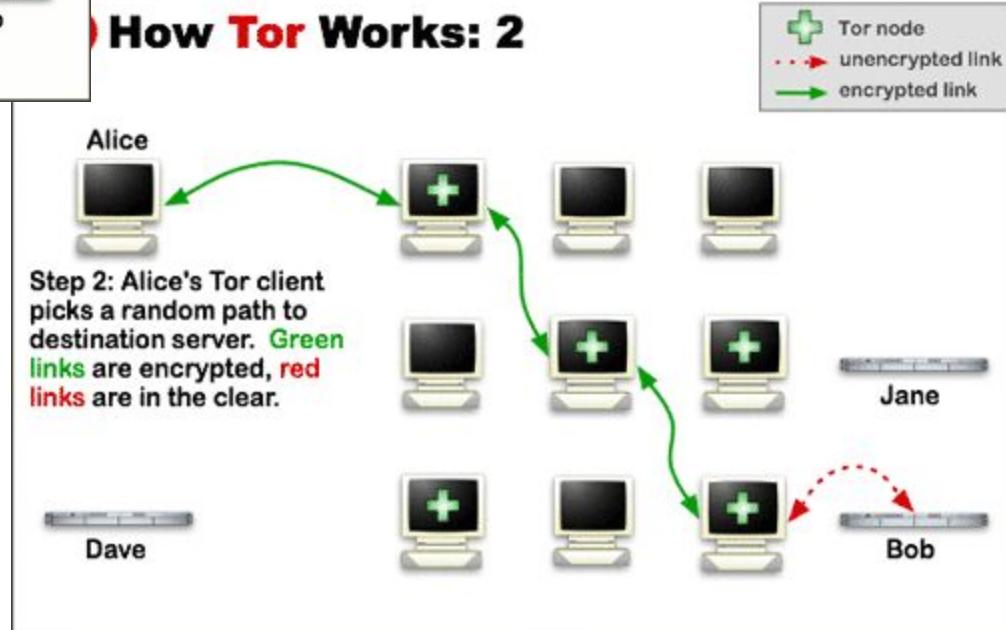
# Establish TOR circuit

## How Tor Works: 1

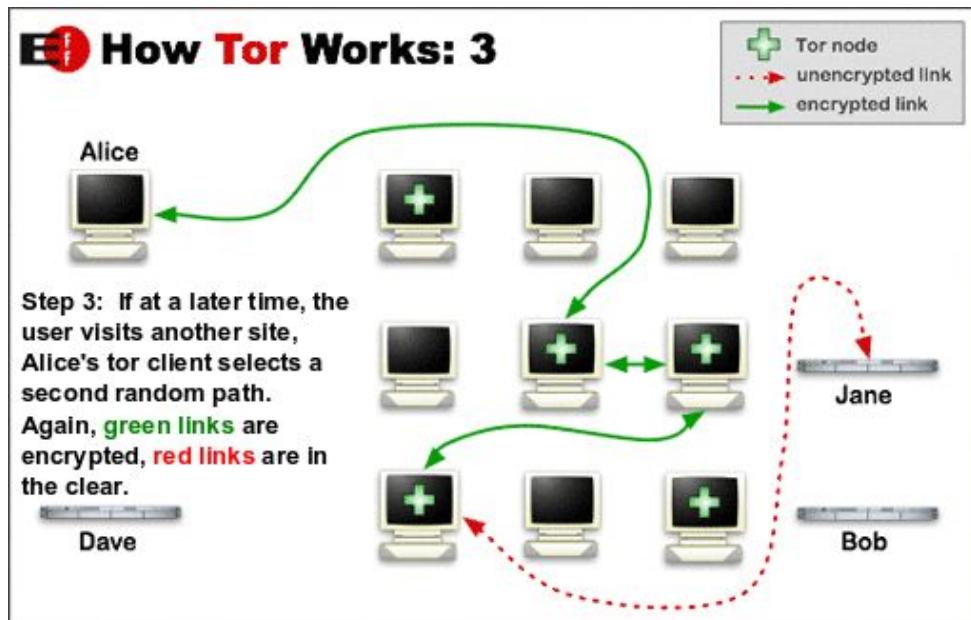


User's software or client incrementally builds a circuit of encrypted connections through relays on the network.

## How Tor Works: 2

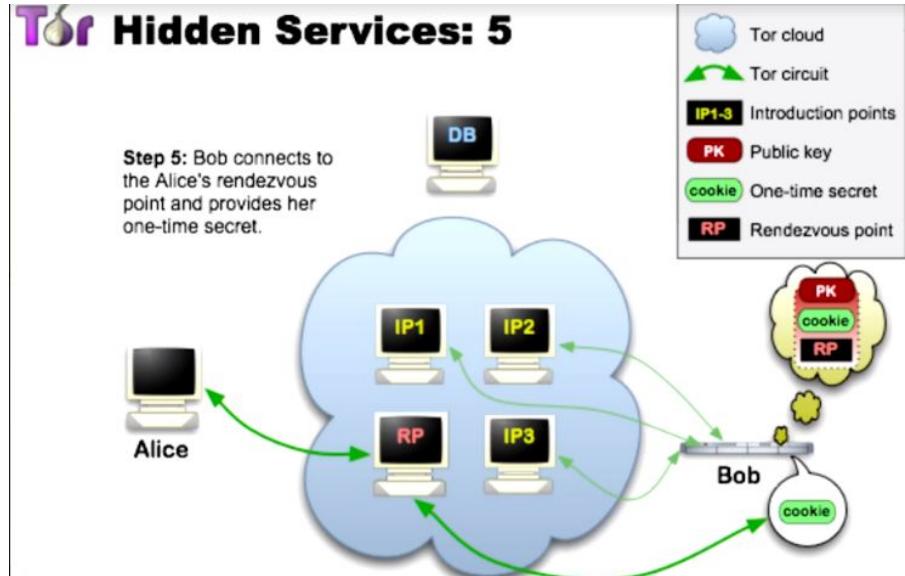


# Establish TOR circuit

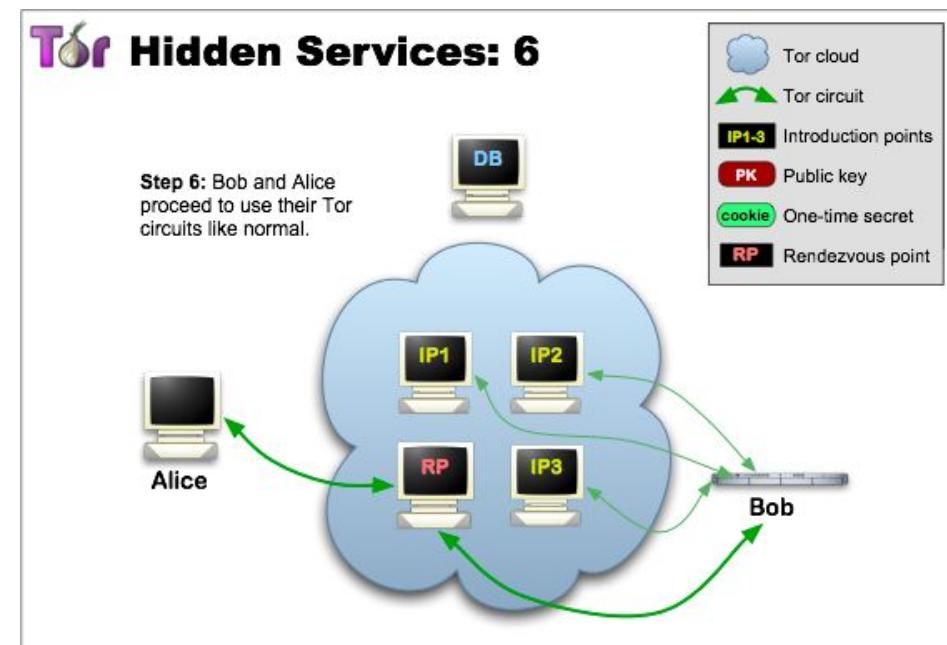


When we connect to the TOR network, we do it through a circuit formed by 3 repeaters, where the encrypted packet sent from the client is passing. Each time the packet goes through a repeater, an encryption layer is added.

# Hidden services



User's software or client incrementally builds a circuit of encrypted connections through relays on the network.

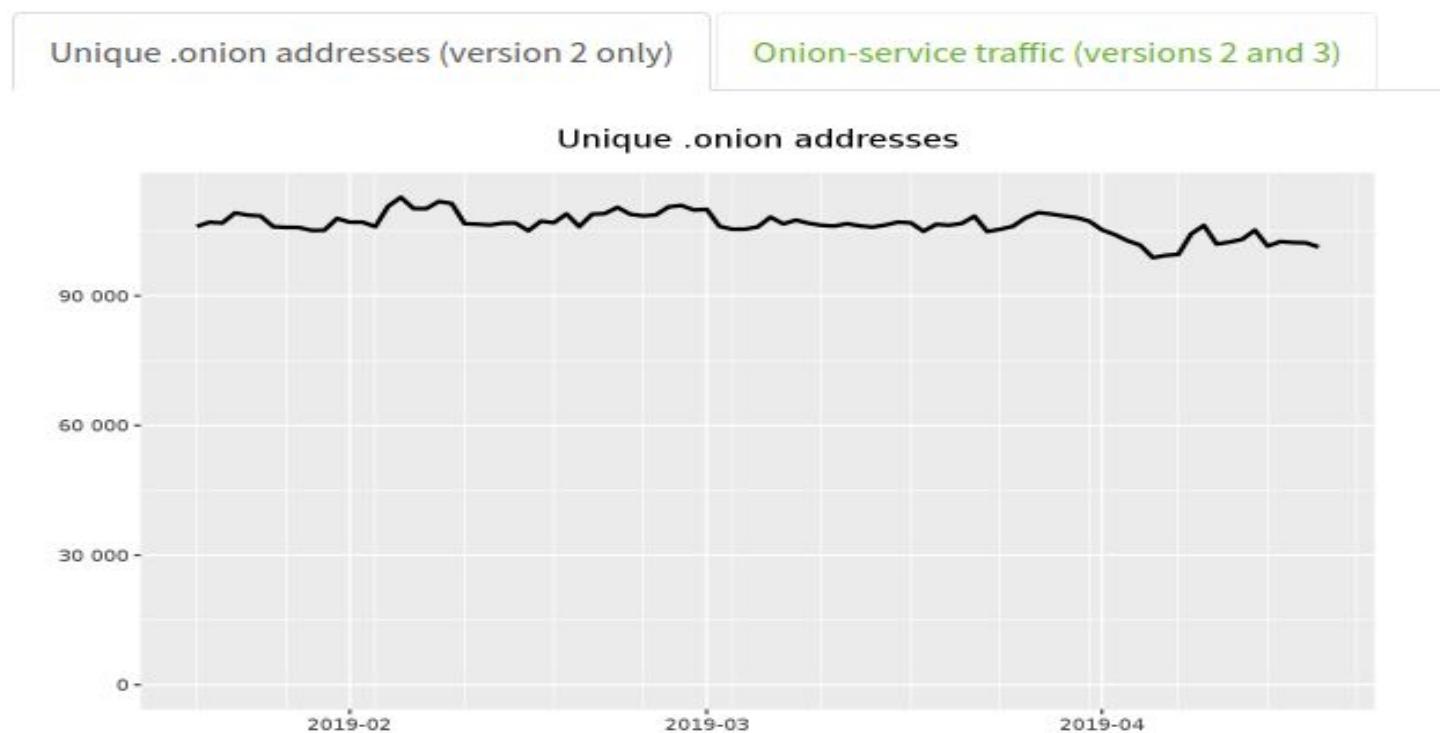


# Hidden services

<https://metrics.torproject.org/hidserv-dir-onions-seen.html>

## Onion Services

Onion services are services that are only accessible via the Tor network.



# Tor NODE List



Aggregate Network Statistic Summary   Graphs / Details		
<b>Total Bandwidth of displayed Routers [MByte/s]</b>	20734	
<b>Total Number of Routers</b>	6641	100%
<b>Routers in Current Query Result Set</b>	6640	99.98%
<b>Total Number of 'Authority' Routers</b>	10	0.2%
<b>Total Number of 'Bad Directory' Routers</b>	0	0%
<b>Total Number of 'Bad Exit' Routers</b>	1	0%
<b>Total Number of 'Exit' Routers</b>	921	13.9%
<b>Total Number of 'Fast' Routers</b>	5783	87.1%
<b>Total Number of 'Guard' Routers</b>	2922	44%
<b>Total Number of 'Hibernating' Routers</b>	1	0%
<b>Total Number of 'Named' Routers</b>	0	0%
<b>Total Number of 'Stable' Routers</b>	5592	84.2%
<b>Total Number of 'Running' Routers</b>	6641	100%
<b>Total Number of 'Valid' Routers</b>	6641	100%
<b>Total Number of 'V2Dir' Routers</b>	6012	90.5%
<b>Total Number of 'Directory Mirror' Routers</b>	4457	67.1%

# Tor NODE List



<https://www.dan.me.uk/tornodes>

<http://torstatus.blutmagie.de>

Router Name	Bandwidth (KB/s)	Uptime	Hostname	ORPort	DirPort	Bad Exit	FirstSeen	ASName
IPredator	95512	9 d	exit1.ipredator.se [197.231.221.211]	443	9030	✗	2014-04-19	CYBERDYNE, L
BizNasty	73940	80 d	46.166.187.77 [46.166.187.77]	443	9001	9030	✗	2019-02-03
Hansa	72654	15 d	ns3130724.ip-51-75-145.eu [51.75.145.219]	443	80	✗	2019-04-03	OVH, FR
exit4	59340	19 d	exit4.tor-network.net [31.220.40.54]	443	80	✗	2019-03-13	AMARUTU-TEC
flo	58785	62 d	delta.flo-du.com [144.76.196.92]	443	9001	9090	✗	2018-10-19
djjohn	51026	125 d	ns3129544.ip-51-75-144.eu [51.75.144.67]	443	8080	✗	2018-12-21	OVH, FR
AIKO	50623	112 d	78.129.218.56 [78.129.218.56]	443	80	✗	2019-01-03	IOMART-AS, GE
PIAzrhexit	46313	42 d	zrh-exit.privateinternetaccess.com [195.206.105.217]	443	80	✗	2018-12-21	M247, GB
exit3	45161	42 d	exit3.tor-network.net [31.220.0.225]	443	80	✗	2018-12-15	AMARUTU-TEC
PrivacyRepublic0001	44514	157 d	tor-exit-node.1.privacyrepublic.org [178.32.181.96]	443	80	✗	2014-11-21	OVH, FR
ExitNinja	44412	59 d	46.165.245.154 [46.165.245.154]	443	80	✗	2014-11-06	LEASEWEB-DE
haedus	44146	13 d	static.232.44.9.176.clients.your-server.de [176.9.44.232]	443	9001	9030	✗	2018-08-03
torrelay01	43837	128 d	78.129.150.54 [78.129.150.54]	443	9001	9030	✗	2018-12-18
HORUS1	43349	4 d	ra.horus-it.com [94.130.34.199]	443	9001	9030	✗	2016-08-04
hyacinthinus	41936	21 d	94.23.150.81 [94.23.150.81]	443	80	✗	2017-05-01	OVH, FR
theWOPR	41336	125 d	customer.worldstream.nl [212.8.243.229]	443	9001	9030	✗	2018-12-21
martasy	41094	50 d	ip81-173-112-100.pbiaas.com [81.173.112.100]	443	9030	9040	✗	2018-12-11
sellerie	40211	55 d	tor-exit-anonymizer-09.appliedprivacy.net [109.70.100.10]	443	80	✗	2018-05-01	NEXTLAYER-AS
idideditheconfig	40084	58 d	185-171-130-240.skpnet.nl [185.171.130.240]	443	9001	9030	✗	2018-02-11
pouity	39976	44 d	static.166.105.4.46.clients.your-server.de [46.4.105.166]	443	80	✗	2016-01-26	HETZNER-AS, I
gurke	39557	60 d	tor-exit-anonymizer-07.appliedprivacy.net [109.70.100.8]	443	80	✗	2018-04-16	NEXTLAYER-AS
rucola	38926	60 d	tor-exit-anonymizer-08.appliedprivacy.net [109.70.100.9]	443	80	✗	2018-04-18	NEXTLAYER-AS
Unnamed	38806	128 d	78.129.150.83 [78.129.150.83]	443	9001	9030	✗	2018-12-18
erbsc	38362	60 d	tor-exit-anonymizer-02.appliedprivacy.net [109.70.100.3]	443	80	✗	2018-04-05	NEXTLAYER-AS
PIAnyexit	38119	42 d	nyc-exit.privateinternetaccess.com [209.95.51.11]	443	80	✗	2019-01-22	HOSTINGSERV
radieschen	37486	60 d	tor-exit-anonymizer-01.appliedprivacy.net [109.70.100.2]	443	80	✗	2018-08-16	NEXTLAYER-AS
sofia	37327	12 h	chomsky.torservers.net [77.247.181.162]	443	80	✗	2017-10-03	NFORCE, NL

# Tor NODE List



<https://onionite.now.sh>



Search...



## Top nodes by consensus weight

#	Nickname	Bandwidth	Uptime	Country	Flags	Type
1	<a href="#">IPredator</a>	143 MB/s	9d 21h	Liberia	✓	Relay
2	<a href="#">flo</a>	92 MB/s	62d 4h	Germany	✓	Relay
3	<a href="#">PrivacyRepublic0001</a>	71.3 MB/s	157d 2h	France	✓	Relay
4	<a href="#">torrelay01</a>	69.8 MB/s	128d 6h	United Kingdom	✓	Relay
5	<a href="#">Hansa</a>	113 MB/s	15d 15h	Germany	✓	Relay
6	<a href="#">AIKO</a>	81.3 MB/s	112d 17h	United Kingdom	✓	Relay
7	<a href="#">exit4</a>	81.8 MB/s	19d 6h	Germany	✓	Relay
8	<a href="#">Unnamed</a>	65.3 MB/s	128d 6h	United Kingdom	✓	Relay
9	<a href="#">drjohn</a>	81.7 MB/s	125d 1h	Germany	✓	Relay
10	<a href="#">exit3</a>	62.3 MB/s	42d 4h	Netherlands	✓	Relay

# Exonera TOR



<https://metrics.torproject.org/exonerator.html>

Home » Services » ExoneraTor

## ExoneraTor

Enter an IP address and date to find out whether that address was used as a Tor relay:

IP address

86.59.21.38

Date

dd/mm/yyyy

Search

### About Tor

Tor is an international software project to anonymize Internet traffic by [encrypting packets and sending them through a series of hops before they reach their destination](#). Therefore, if you see traffic from a Tor relay, this traffic usually originates from someone using Tor, rather than from the relay operator. The Tor Project and Tor relay operators have no records of the traffic that passes over the network and therefore cannot provide any information about its origin. Be sure to [learn more about Tor](#), and don't hesitate to [contact The Tor Project, Inc.](#) for more information.

### About ExoneraTor

The ExoneraTor service maintains a database of IP addresses that have been part of the Tor network. It answers the question whether there was a Tor relay running on a given IP address on a given date. ExoneraTor may store more than one IP address per relay if relays use a different IP address for exiting to the Internet than for registering in the Tor network, and it stores whether a relay permitted transit of Tor traffic to the open Internet at that time.

# Relay search



<https://metrics.torproject.org/rs.html#simple>

## Relay Search

Simple Search

Aggregated Search

Advanced Search

The relay search tool displays data about single relays and bridges in the Tor network. It provides useful information on how relays are configured along with graphs about their past.

1.161.127.163

Search Top Relays

You can search for Tor relays and bridges by using keywords. In particular, this tool enables you to search for (partial) nicknames (e.g., “moria”), IP addresses (e.g., “128.31.”), and fingerprints (e.g., “9695DFC3”). It is also possible to combine searches, e.g., “moria 128.31.”. Finally, you can use qualifiers to search for relays in specific countries (e.g., “moria country:us”), with specific contact information (e.g., “contact:arma”), or with specific flags (e.g., “flag:Authority”).

If you are searching for a bridge, you will need to search by the hashed fingerprint. This prevents leaking the fingerprint of the bridge when searching. You can find this in the `hashed-fingerprint` file in the Tor data directory. On Debian systems, this is in `/var/lib/tor` but may be in another location on your system. The location is specified as `DataDirectory` in your `torrc`.

# Relay search



<https://metrics.torproject.org/rs.html#simple>

## Relay Search

country:it

Show 10 entries

Advertised										Type
Nickname <sup>†</sup>	Bandwidth	Uptime	Country	IPv4	IPv6	Flags	Add. Flags	ORPort	DirPort	Type
● pingu (4)	20.4 MiB/s	23d 22h	🇮🇹	176.126.83.211	-	⚡ ⚡ 🚫 HS ⇕ ○ V2 ✓		9001	9030	Relay
● Dazzle (1)	18.86 MiB/s	97d 6h	🇮🇹	54.37.207.82	-	⚡ 🚫 HS ⇕ ○ V2 ✓	⚠	443	0	Relay
● bauruine55 (8)	18.1 MiB/s	51d 7h	🇮🇹	158.58.170.183	2a02:29e0:2:5::7	⚡ 🚫 HS ⇕ ○ V2 ✓	v6	443	80	Relay
● mailaddressislegit (1)	17.6 MiB/s	28d 9h	🇮🇹	83.136.106.130	2a02:29e0:2:6:1:1:123d:4d1c	⚡ 🚫 HS ⇕ ○ V2 ✓	v6	54200	54400	Relay
● CanopolT (6)	16.88 MiB/s	47d 8h	🇮🇹	37.9.231.195	2001:4b78:2006:ffc3::1	⚡ 🚫 HS ⇕ ○ V2 ✓	v6 v6	443	80	Relay
● bauruine66 (8)	15.15 MiB/s	51d 7h	🇮🇹	94.198.98.21	2a02:29e0:2:5::1c	⚡ 🚫 HS ⇕ ○ V2 ✓	v6	3443	8888	Relay
● bauruine56 (8)	12.95 MiB/s	51d 7h	🇮🇹	94.198.98.21	2a02:29e0:2:5::1c	⚡ 🚫 HS ⇕ ○ V2 ✓	v6	443	80	Relay
● bauruine65 (8)	12.71 MiB/s	51d 7h	🇮🇹	158.58.170.183	2a02:29e0:2:5::7	⚡ 🚫 HS ⇕ ○ V2 ✓	v6	4443	5580	Relay
● Unnamed (1)	11.49 MiB/s	203d 17m	🇮🇹	91.134.147.134	-	⚡ 🚫 HS ⇕ ○ V2 ✓	⚠	9001	9030	Relay
● blackmamba (1)	11.49 MiB/s	45d 8h	🇮🇹	195.135.194.134	2001:678:7dc:134::dead:beef	⚡ 🚫 HS ⇕ ○ V2 ✓	⚠ 🔍 v6	443	80	Relay
Total	378.56									
	MiB/s									

# Relay search



<https://metrics.torproject.org/rs.html#simple>

## Relay Search

### Details for: Unnamed •

#### Configuration

##### Nickname

Unnamed

##### OR Addresses

1.161.127.163:80

#### Contact

kcuw A csie org

#### Dir Address

none

#### Exit Addresses

1.161.127.163

#### Advertised Bandwidth

1.58 MiB/s

#### IPv4 Exit Policy Summary

accept

53

80

443

1194

8080

#### Properties

##### Fingerprint

0A50AA36FD41F4BED31810DED7CDB9B3D32E1686

##### Uptime

1 day 1 hour 23 minute and 45 seconds

##### Flags

⚡ Exit ⚡ Fast ⇄ Running 🏠 V2Dir ✅ Valid

##### Additional Flags

none

##### Host Name

1-161-127-163.dynamic-ip.hinet.net

##### Country

🇹🇼 Taiwan (⚡)

##### AS Number

AS3462

##### AS Name

Data Communication Business Group

# Discover hidden services



**HiddenWiki:** <http://wikitjerrta4qgz4.onion/>

**Dark Links:** <http://wiki5kauuihowqi5.onion>

**Tor Links:** <http://torlinkbqs6aabns.onion>

**Dark Web Links:**

<http://jdpskjmgy6kk4urv.onion/links.html>

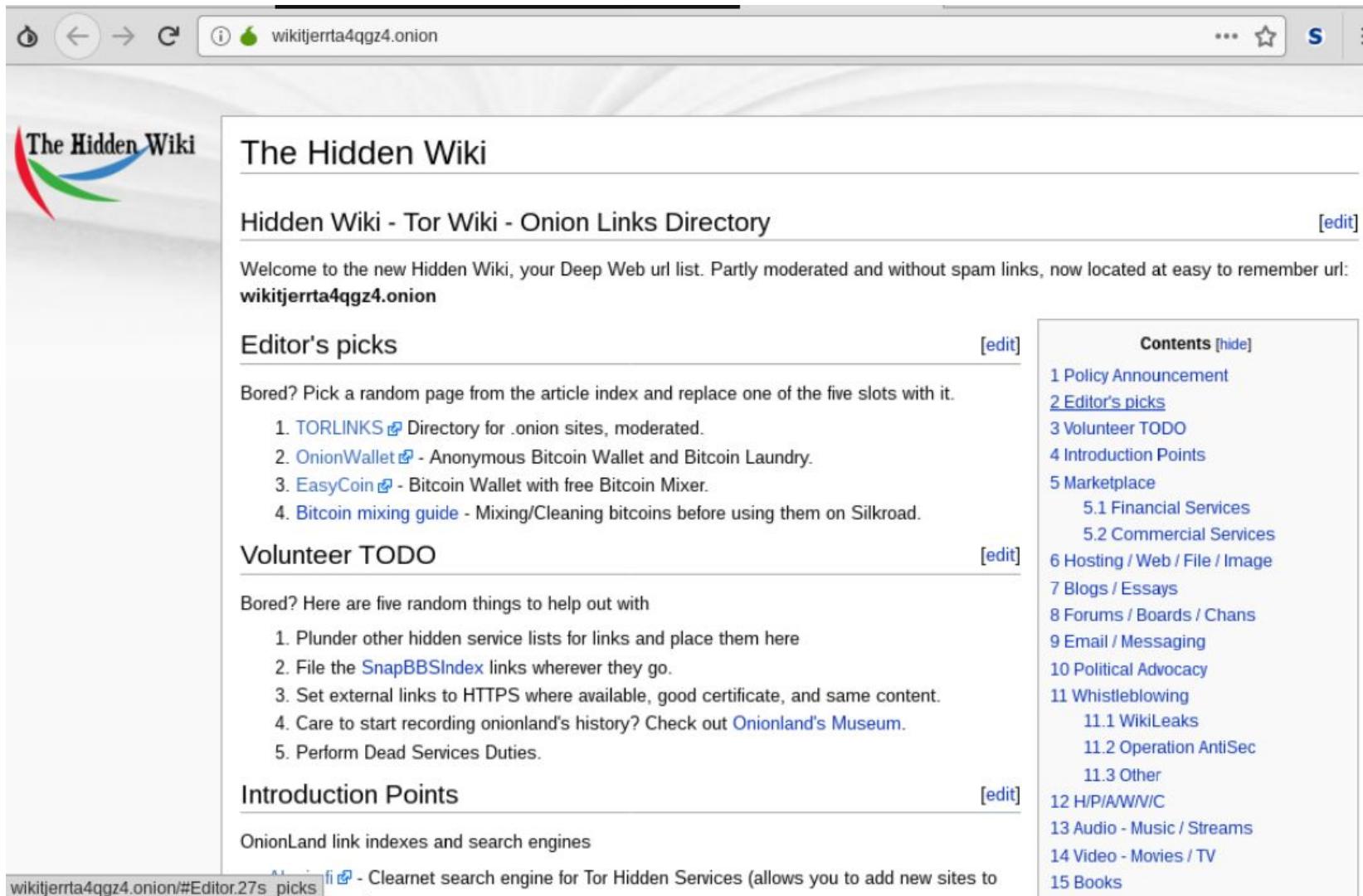
**HDWiki:** <http://hdwikicorldcisiy.onion>

**OnionDir:** <http://dirnxxdraygbifgc.onion>

**DeepLink:** <http://deeplinkdeatbml7.onion>

**Ahmia:** <http://msydqstlz2kzerdg.onion>

# Tor onion services



The screenshot shows a web browser displaying 'The Hidden Wiki' at the URL [wiki.jerrta4qgz4.onion](http://wiki.jerrta4qgz4.onion). The page content includes sections for 'Editor's picks', 'Volunteer TODO', and 'Introduction Points', each with a list of links. A sidebar on the right contains a 'Contents' table of contents with numbered links from 1 to 15, covering various topics like Policy Announcement, Editor's picks, Volunteer TODO, Introduction Points, Marketplace, and Books. At the bottom of the page, there is a note about a search engine and a footer link to the Onionland Museum.

The Hidden Wiki

## Hidden Wiki - Tor Wiki - Onion Links Directory [edit]

Welcome to the new Hidden Wiki, your Deep Web url list. Partly moderated and without spam links, now located at easy to remember url: [wiki.jerrta4qgz4.onion](http://wiki.jerrta4qgz4.onion)

### Editor's picks [edit]

Bored? Pick a random page from the article index and replace one of the five slots with it.

1. [TORLINKS](#) - Directory for .onion sites, moderated.
2. [OnionWallet](#) - Anonymous Bitcoin Wallet and Bitcoin Laundry.
3. [EasyCoin](#) - Bitcoin Wallet with free Bitcoin Mixer.
4. [Bitcoin mixing guide](#) - Mixing/Cleaning bitcoins before using them on Silkroad.

### Volunteer TODO [edit]

Bored? Here are five random things to help out with

1. Plunder other hidden service lists for links and place them here
2. File the [SnapBBSIndex](#) links wherever they go.
3. Set external links to HTTPS where available, good certificate, and same content.
4. Care to start recording onionland's history? Check out [Onionland's Museum](#).
5. Perform Dead Services Duties.

### Introduction Points [edit]

OnionLand link indexes and search engines

[fi](#) - Clearnet search engine for Tor Hidden Services (allows you to add new sites to [wiki.jerrta4qgz4.onion/#Editor.27s\\_picks](http://wiki.jerrta4qgz4.onion/#Editor.27s_picks))

### Contents [hide]

- 1 Policy Announcement
- 2 [Editor's picks](#)
- 3 Volunteer TODO
- 4 Introduction Points
- 5 Marketplace
  - 5.1 Financial Services
  - 5.2 Commercial Services
- 6 Hosting / Web / File / Image
- 7 Blogs / Essays
- 8 Forums / Boards / Chans
- 9 Email / Messaging
- 10 Political Advocacy
- 11 Whistleblowing
  - 11.1 WikiLeaks
  - 11.2 Operation AntiSec
  - 11.3 Other
- 12 H/P/A/W/V/C
- 13 Audio - Music / Streams
- 14 Video - Movies / TV
- 15 Books

# Tor onion services



[https://en.wikipedia.org/wiki/List\\_of\\_Tor\\_onion\\_services](https://en.wikipedia.org/wiki/List_of_Tor_onion_services)

[https://en.wikipedia.org/wiki/The\\_Hidden\\_Wiki](https://en.wikipedia.org/wiki/The_Hidden_Wiki)

V · T · E	Tor onion services	[hide]
	<a href="#">List</a> · <a href="#">Category</a>	
<a href="#">Web directories</a>	<a href="#">The Hidden Wiki</a>	
<a href="#">Search engines</a>	Ahmia · DuckDuckGo · Grams · MetaGer · Searx	
<a href="#">File storage and peer-to-peer file sharing</a>	BTDig - Freedom Hosting - Free Haven Project - KickassTorrents - The Pirate Bay	
<a href="#">Email and instant messaging</a>	Bitmessage.ch · Riseup · Tor Mail · TorChat · SIGAIN · ProtonMail	
<a href="#">Social media and forums</a>	8chan · Dark0de · Facebook · HackBB · Russian Anonymous Marketplace · The Hub · Tor Carding Forum	
<a href="#">Cryptocurrency tumblers</a>	Bitcoin Fog · Blockchain.info · Helix	
<a href="#">Commerce</a>	Agora · AlphaBay · Assassination market · Atlantis · Black Market Reloaded · Dream Market · Evolution · The Farmer's Market · Hansa · Sheep Marketplace · Silk Road · TheRealDeal · Utopia	
<a href="#">News, whistleblowing, and document archives</a>	Archive.is · BuggedPlanet · DeepDotWeb · Doxbin · Filtrala · GlobaLeaks · Independent Media Center · Ljost · NawaatLeaks · ProPublica · SecureDrop · Sci-Hub · The Daily Stormer · The Intercept · WildLeaks · WikiLeaks	
<a href="#">Nonprofit organizations</a>	Courage Foundation · Freedom of the Press Foundation · La Quadrature du Net · Telecomix	
<a href="#">Pornography</a>	Lolita City · Playpen · Childs Play	
Tor · .onion domain · Tor2web		

<https://www.onion.to>



## onion.to Tor Hidden Services Gateway

This gateway to Tor hidden services provides convenient access to Tor hidden services. It is a pure proxy that forwards requests to the respective hidden service. We do not store any data and are not liable for the content.

### No anonymity!

Onion.to as a gateway **cannot offer any anonymity for the visitor**. For example, both onion.to and the hidden service itself can see the visitor's IP address, and use [browser fingerprinting](#) to track users across different sessions. [In all cases, it is better to download the Tor Browser Bundle](#) and access the hidden service using Tor (don't forget to remove the .to from the URL!).

Enter onion address here:

[open via onion.to proxy](#)

# TOR browser



<https://www.torproject.org/download/>

A screenshot of the official Tor Project download page. The header features the 'Tor' logo, navigation links for About, Documentation, Support, Blog, and Donate, a language dropdown set to English (En), and a prominent 'Download Tor Browser' button. The main message is 'Defend yourself.' followed by the subtext 'Protect yourself against tracking, surveillance, and censorship.' Below this, four operating system icons (Windows, macOS, Linux, and Android) are shown in circular frames, each accompanied by a small 'Sig' button at the bottom.

# Onion Routing

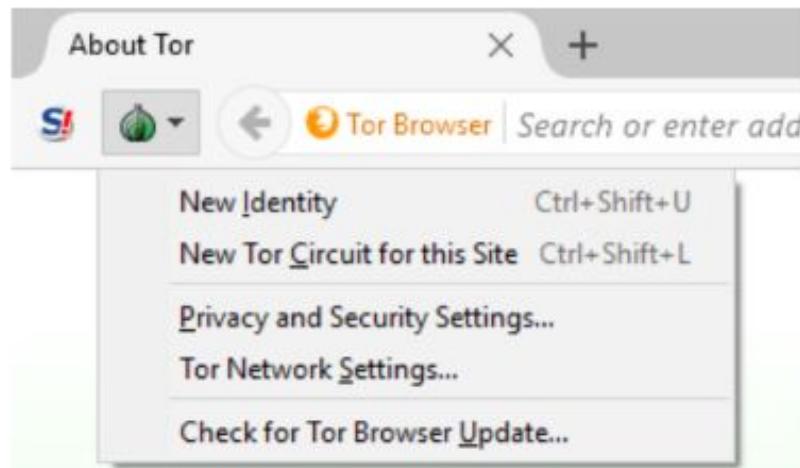


## How do circuits work?

Circuits are made up of randomly assigned relays, which are computers around the world configured to forward Tor traffic. Circuits allow you to browse privately and to connect to onion services.

1 of 3

Next



Duck Duck Go, Inc.  
Secure Connection

Tor Circuit

- This browser
- Germany 217.182.196.70 Guard
- Germany 144.76.196.92
- Germany 80.240.31.211
- Relay
- Relay
- Relay
- 3g2upl4pq6kufc4m.onion

New Circuit for this Site

Your Guard node may not change. [Learn more](#)

Circuit Display

This diagram shows the relays that make up the circuit for this website. To prevent linking of activity across different sites, each website gets a different circuit.

2 of 3

Next

# Installing TOR



```
sudo apt-get update  
sudo apt-get install tor  
sudo /etc/init.d/tor restart
```



```
## Configuration file for a typical Tor user
## Last updated 22 September 2015 for Tor 0.2.7.3-alpha.
## (may or may not work for much older or much newer versions of Tor.)
##
## Lines that begin with "## " try to explain what's going on. Lines
## that begin with just "#" are disabled commands: you can enable them
## by removing the "#" symbol.
##
## See 'man tor', or https://www.torproject.org/docs/tor-manual.html,
## for more options you can use in this file.
##
## Tor will look for this file in various places based on your platform:
## https://www.torproject.org/docs/faq#torrc
##
## Tor opens a SOCKS proxy on port 9050 by default -- even if you don't
## configure one below. Set "SOCKSPort 0" if you plan to run Tor only
## as a relay, and not make any local application connections yourself.
#SOCKSPort 9050 # Default: Bind to localhost:9050 for local connections.
#SOCKSPort 192.168.0.1:9100 # Bind to this address:port too.
```

# Running TOR



```
$ tor --SocksPort 9050 --ControlPort 9051
```

```
Tor 0.2.9.14 running on Linux with Libevent 2.0.21-stable, (c)
Tor can't help you if you use it wrong! Learn how to be safe.
Read configuration file "/etc/tor/torrc".
ControlPort is open, but no authentication method has been configured.
Upgrade your Tor controller as soon as possible.

Opening Socks listener on 127.0.0.1:9050
Opening Control listener on 127.0.0.1:9051
Bootstrapped 0%: Starting
Bootstrapped 80%: Connecting to the Tor network
Bootstrapped 85%: Finishing handshake with first hop
Bootstrapped 90%: Establishing a Tor circuit
Tor has successfully opened a circuit. Looks like client functionality is available.
Bootstrapped 100%: Done
```

# Running TOR



```
Bootstrapped 0%: Starting
Bootstrapped 5%: Connecting to directory server
Bootstrapped 10%: Finishing handshake with directory server
Bootstrapped 15%: Establishing an encrypted directory connection
Bootstrapped 20%: Asking for networkstatus consensus
Bootstrapped 25%: Loading networkstatus consensus
I learned some more directory information, but not enough to build a circuit: I
have 0% of guards bw, 0% of midpoint bw, and 0% of exit bw = 0% of path bw.)
Bootstrapped 50%: Loading relay descriptors
Bootstrapped 55%: Loading relay descriptors
Bootstrapped 61%: Loading relay descriptors
Bootstrapped 66%: Loading relay descriptors
Bootstrapped 72%: Loading relay descriptors
Bootstrapped 80%: Connecting to the Tor network
Bootstrapped 90%: Establishing a Tor circuit
Tor has successfully opened a circuit. Looks like client functionality is work
Bootstrapped 100%: Done
```

service tor start/restart  
service tor status

```
● tor.service - Anonymizing overlay network for TCP (multi-instance-master)
  Loaded: loaded (/lib/systemd/system/tor.service; enabled; vendor preset: enabled)
  Active: active (exited) since lun 2019-04-22 14:51:32 CEST; 6s ago
    Process: 6371 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 6371 (code=exited, status=0/SUCCESS)
     Tasks: 0
    Memory: 0B
      CPU: 0
     CGroup: /system.slice/tor.service
```

# Connecting with TOR



Stem

<https://stem.torproject.org/>

TorRequest

<https://github.com/erdiaker/torrequest>

Requests + socks5

# pip install stem

```
.. data:: Signal (enum)           # Nombre
          Signals that the tor process will accept.
Description
.. versionchanged:: 1.3.0
    Added the HEARTBEAT signal.
Competitors()
Re===== =====
Signal             Description
===== =====
**RELOAD** or **HUP**      reloads our torrc
**SHUTDOWN** or **INT**    shut down, waiting ShutdownWaitLength first if we're a relay
**DUMP** or **USR1**       dumps information about open connections and circuits to our log
**DEBUG** or **USR2**      switch our logging to the DEBUG runlevel
**HALT** or **TERM**       exit tor immediately
**NEWNYM**                switch to new circuits, so new application requests don't share any circuits
**CLEARDNSCACHE**         clears cached DNS results
**HEARTBEAT**             trigger a heartbeat log message
===== =====
```

# TOR descriptors



**Server descriptor:** Complete information about a repeater

**ExtraInfo descriptor:** Extra information about the repeater

**Micro descriptor:** Contains only the information necessary for TOR clients to communicate with the repeater

**Consensus (Network status):** File issued by the authoritative entities of the network and made up of multiple entries of information on repeaters (router status entry)

**Router status entry:** Information about a repeater in the network, each of these elements is included in the consensus file generated by the authoritative entities.

# TOR spec



**Tor** index : torspec

Tor's protocol specifications

summary refs log tree commit diff log msg ▾

Branch	Commit message	Author
master	Merge remote-tracking branch 'tor-github/pr/73'	Nick Mathewson

Age Commit message Author

2019-04-04 Merge remote-tracking branch 'tor-github/pr/73' HEAD Nick Mathewson

2019-03-30 Merge remote-tracking branch 'tor-github/pr/74'

bandwidth-file: Add time to report half network header KeyValue

bandwidth-file: Add relay line diagnostic KeyValues

bandwidth-file: Add summaries for format version 1.3.0 and 1.4

bandwidth-file: Add new RelayLine monitoring KeyValues

dir-spec: Update the Tor version for bandwidth-file-digest

bandwidth-file: Remove SP from the end of KeyValue

bandwidth-file: Fix missing cardinalities, format versions, and h

bandwidth-file: Add version 1.4.0 examples

Recognized keys and their values include:

"version" -- The version of the server's software, which MAY include the name of the software, such as "Tor 0.0.9.4". The name of the software, if absent, is assumed to be "Tor".

"config-file" -- The location of Tor's configuration file ("torrc").

"config-defaults-file" -- The location of Tor's configuration defaults file ("torrc.defaults"). This file gets parsed before torrc, and is typically used to replace Tor's default configuration values. [First implemented in 0.2.3.9-alpha.]

"config-text" -- The contents that Tor would write if you send it a SAVECONF command, so the controller can write the file to disk itself. [First implemented in 0.2.2.7-alpha.]

"exit-policy/default" -- The default exit policy lines that Tor will \*append\* to the ExitPolicy config option.

"exit-policy/reject-private/default" -- The default exit policy lines that Tor will \*prepend\* to the ExitPolicy config option when ExitPolicyRejectPrivate is 1.

```
from stem import Signal
from stem.control import Controller

with Controller.from_port(port = 9051) as controller:
    controller.authenticate(password='your
password set for tor controller port in torrc')
    print("Success!")
    controller.signal(Signal.NEWNYM)
    print("New Tor connection processed")
```

# Periodic Tor IP Rotation



```
import time
from stem import Signal
from stem.control import Controller
def main():
    while True:
        time.sleep(20)
        print ("Rotating IP")
        with Controller.from_port(port = 9051) as controller:
            controller.authenticate()
            controller.signal(Signal.NEWNYM) #gets new identity

if __name__ == '__main__':
    main()
```

# Stem.Circuit status



```
from stem.control import Controller  
  
controller = Controller.from_port(port=9051)  
controller.authenticate()  
  
print(controller.get_info('circuit-status'))
```

# Stem.Network status



```
from stem.control import Controller  
  
controller = Controller.from_port(port=9051)  
controller.authenticate(password)  
  
entries = controller.get_network_statuses()  
for routerEntry in entries:  
    print(routerEntry)
```

# Stem.circuits



```
for circ in sorted(controller.get_circuits()):  
    if circ.status != CircStatus.BUILT:  
        continue  
  
    print("")  
    print("Circuit %s (%s)" % (circ.id, circ.purpose))  
  
    for i, entry in enumerate(circ.path):  
        div = '+' if (i == len(circ.path) - 1) else '|'  
        fingerprint, nickname = entry  
  
        desc = controller.get_network_status(fingerprint, None)  
        address = desc.address if desc else 'unknown'  
  
        print(" %s- %s (%s, %s)" % (div, fingerprint, nickname, address))
```

# Stem.circuits



## Circuit 10 (GENERAL)

```
| - CE3FE883C6C9EF475EA097DC3E33A6F32B852DA1 (AIKO, 78.129.218.56)
| - 12CF6DB4DAE106206D6C6B09988E865C0509843B (ATZv5, 159.69.114.110)
+- E19D4503D2FD584C8099A954270A9BC819596E74 (Unnamed, 51.68.206.35)
```

## Circuit 11 (GENERAL)

```
| - CE3FE883C6C9EF475EA097DC3E33A6F32B852DA1 (AIKO, 78.129.218.56)
| - 44DF1007B545B4D8057F279025EBB33CF99BE227 (Kroell, 80.241.214.102)
+- 9612664500871798CFB52E8A71A956F316AA0503 (Polaris, 130.230.113.235)
```

## Circuit 12 (GENERAL)

```
| - CE3FE883C6C9EF475EA097DC3E33A6F32B852DA1 (AIKO, 78.129.218.56)
| - 9E1E4F5B5F94812D02C4D18CB4086CE71CA5C614 (torpidsDEhetzner1, 78.46.217.214)
+- 615ABEA2DE76EB3760BC51E7306BAA59F15CD8F2 (Cloud, 5.135.158.101)
```

## Circuit 13 (GENERAL)

```
| - CE3FE883C6C9EF475EA097DC3E33A6F32B852DA1 (AIKO, 78.129.218.56)
| - 91B14EB2893544F0EC8F16086261A10B8E46B5C5 (okthx, 163.172.210.167)
+- 03EE7DDD931D92BB57B81B3038AE7C40A08AB237 (Shockrealm, 123.30.128.138)
```

## Circuit 14 (GENERAL)

```
| - CE3FE883C6C9EF475EA097DC3E33A6F32B852DA1 (AIKO, 78.129.218.56)
```

# Server descriptors



```
import stem.descriptor.remote

server_descriptors = stem.descriptor.remote.get_server_descriptors().run()

with open('/tmp/descriptor_dump', 'wb') as descriptor_file:
    descriptor_file.write(''.join(map(str, server_descriptors)))
```

```
class stem.descriptor.server_descriptor.RelayDescriptor(raw_contents, validate=False, annotations=None, skip_crypto_validation=False)
```

Bases: `stem.descriptor.server_descriptor.ServerDescriptor`

Server descriptor (**descriptor specification**)

**Variables:**

- `certificate` (`stem.certificate.Ed25519Certificate`) -- ed25519 certificate
- `ed25519_certificate` (`str`) -- base64 encoded ed25519 certificate
- `ed25519_master_key` (`str`) -- base64 encoded master key for our ed25519 certificate
- `ed25519_signature` (`str`) -- signature of this document using ed25519
- `onion_key` (`str`) -- \* key used to encrypt EXTEND cells
- `onion_key_crosscert` (`str`) -- signature generated using the onion\_key
- `ntor_onion_key_crosscert` (`str`) -- signature generated using the ntor-onion-key
- `ntor_onion_key_crosscert_sign` (`str`) -- sign of the corresponding ed25519 public key
- `signing_key` (`str`) -- \* relay's long-term identity key
- `signature` (`str`) -- \* signature for this descriptor

\* attribute is required when we're parsed with validation

*Changed in version 1.5.0:* Added the `ed25519_certificate`, `ed25519_master_key`, `ed25519_signature`, `onion_key_crosscert`, and `ntor_onion_key_crosscert` attributes.

*Changed in version 1.6.0:* Moved from the deprecated `pycrypto` module to `cryptography` for validating signatures.

*Changed in version 1.6.0:* Added the `certificate` attribute.

*Deprecated since version 1.6.0:* Our `ed25519_certificate` is deprecated in favor of our new `certificate` attribute. The base64 certificate is available via the `certificate`'s `encoded` attribute.

# Introduction points



```
from stem.control import Controller

with Controller.from_port(port = 9051) as controller:
    controller.authenticate()
    desc = controller.get_hidden_service_descriptor('3g2upl4pq6kufc4m')

    print("DuckDuckGo's introduction points are...\n")

    for introduction_point in desc.introduction_points():
        print(' %s:%s => %s' % (introduction_point.address, introduction_point.port, introduction_point.identifier))
~
```

DuckDuckGo's introduction points are...

```
93.72.79.64:9001 => c2otnj2rbdm5o62cfse67j4dz66gmk7y
82.64.78.170:9001 => 2fszgcjszlkiiw66lzfsnedgu6rylbd3
95.216.136.46:9001 => atr3les6wx3gy3wiccr5o4bobsr7tvph
65.49.20.10:9001 => 2nlifk3ofhmrribq4a5jx22coagznipp
194.187.249.116:443 => zzmm6ztpecmfoubpz2xrznjmpofp7pld
144.76.236.14:443 => lgnhbb2wasezhiictmtxl3xyzhsaxnsa
144.217.94.84:443 => ujqniwgkiagv6t6265uqeqbqlmvkkdeu
212.89.225.242:443 => eeyq6sagnjgkvxvsx7jsj4ftr6hbisgw
88.198.70.137:9090 => fdcszw5xrwsgcjg2mlwb2z4wncphdeqq
198.16.70.10:9001 => r3ri5lyiosxzwa7lxv43l22qlhrfi42v
```

<https://nyx.torproject.org/>



## Tor Nyx

---

### Download

Nyx is available Mac OSX, Linux, and BSD but **not Windows**. Find your platform below to get started. For what's changed see our **change log**.

---

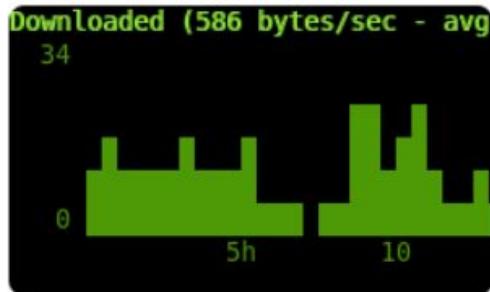


### Python Package Index

Signed releases and instructions for both Python 2.x and 3.x. You can easily install from its **tarball (sig)**, or with **pip**...

```
% sudo easy_install pip  
% sudo pip install nyx
```

## What does Nyx provide?



### << Bandwidth Graph

Bandwidth used by Tor. You can press 'i' to pick the graphing **interval**, or 's' to show other usage **statistics**.

```
18:49:57 [INFO] router_pick_pub
18:49:57 [INFO] resolve_my_addr
  public IP addresses.
18:49:57 [INFO] resolve_my_addr
18:49:57 [INFO] resolve_my_addr
18:49:57 [BW] READ: 586, WRITTE
18:49:56 [BW] READ: 0, WRITTEN:
18:49:55 [BW] READ: 0, WRITTEN:
18:49:54 [BW] READ: 1172, WRITT
```

### << Event Log

Tor logs a wealth of information about itself. We present it, colorized and deduplicated. Press 'e' to select what **events** are logged and 'f' to **filter** to just what you want.



A list of connection endpoints. The entry 'tor.noreply.org:41249' is highlighted with a green background. Other entries include: tor.globenet.org:58010, tor.maimed.org:48250, tor2.digineo.de:52112, torserver.uvt.nl:45561, unpatented.com:57261, vecna.uncg.edu:33392, wiredwings.org:41099, zitaraku.ath.cx:62753.

### << Connections

Connection data similar to netstat or lsof, but correlated with Tor relay information to make it much richer. Press 'enter' for more **details**, 's' to **sort**, and 'd' to see raw **descriptor** data.

# Tor nyx



```
3 User tor
4 DataDirectory /etc/tor
5 SocksPort 0
6 SocksListenAddress 127.0.0.1
7 PidFile /var/run/tor/tor.pid
8 Log notice file /var/log/tor
9 RunAsDaemon 1
10 ControlPort 9051
11 HashedControlPassword 16:22B
```

## « Torrc

Provides your torrc with line numbers and syntax highlighting. Comments can be **stripped** by pressing 's'.

```
os: Linux x86_64
version: 0.2.1.30
flags: Fast, Guard, Named, Runn
exit policy: reject 1-65535
contact: www.atagar.com/contact

>>> GETINFO version
0.2.2.23-alpha (git-b85eb949b52)
```

## « Interpreter

Integration with Stem's **interpreter**, providing **raw controller access**, irc-style commands like **/help**, tab completion, history scrollback, and a python prompt.

Actions	View	Graph	Log
Close Menu	7.173.186:9001, D		
New Identity	1.0% arm mem:		
Stop Tor	6A29789D9BC8A8879		
Reset Tor			
Setup Wizard			
Pause	to close the men		
Exit	it: 40 Mb/s, burs		
Download (0.0 b/sec):			

## « ... and more!

That's just the tip of the iceberg. All pages within Nyx provide **help** information when you press 'h' and **menu** in response to 'm'. So go explore!

# Tor nyx



```
page 3 / 5 - m: menu, p: pause, h: page help, q: quit
Tor Configuration (press 'a' to show all options):
CookieAuthentication (General Option)
Value: True (default, Boolean, usage: 0|1)
Description: If this option is set to 1, allow connections on the control port when the connecting process knows the contents of a file named "control_auth_cookie", which Tor will create in its data directory. This authentication method should only be used on systems with good filesystem security. (Default: 0)

BandwidthRate          1 GB
BandwidthBurst         1 GB
RelayBandwidthRate     0 B
RelayBandwidthBurst    0 B
ControlPort             <none>
HashedControlPassword <none>
CookieAuthentication   True
DataDirectory           /home/jmoc/Escritorio/tor-browser_en-US/Browser/TorBrowse...
Log                     notice stdout
RunAsDaemon            False
User                   <none>
Bridge                 <none>
ExcludeNodes          <none>
MaxCircuitDirtiness   10 minutes
SocksPort              <none>
UseBridges             False
BridgeRelay            False
ContactInfo            <none>
ExitPolicy              <none>
MyFamily               <none>
Nickname               <none>
ORPort                 <none>
AccountingMax          0 B
AccountingStart        <none>
DirPortFrontPage      <none>
DirPort                <none>
HiddenServiceDir       <none>
HiddenServicePort      <none>

Average bandwidth usage limit
Maximum bandwidth usage limit
Average bandwidth usage limit for relaying
Maximum bandwidth usage limit for relaying
Port providing access to tor controllers (nyx, vidalia, etc)
Hash of the password for authenticating to the control port
If set, authenticates controllers via a cookie
Location for storing runtime data (state, keys, etc)
Runlevels and location for tor logging
Toggles if tor runs as a daemon process
UID for the process when started
Available bridges
Relays or locales never to be used in circuits
Duration for reusing constructed circuits
Port for using tor as a Socks proxy
Make use of configured bridges
Act as a bridge
Contact information for this relay
Traffic destinations that can exit from this relay
Other relays this operator administers
Identifier for this relay
Port used to accept relay traffic
Amount of traffic before hibernating
Duration of an accounting period
Publish this html file on the DirPort
Port for directory connections
Directory contents for the hidden service
Port the hidden service is provided on
```

# TorRequest



```
from torrequest import TorRequest  
  
with TorRequest() as tr:  
    response = tr.get('http://ipecho.net/plain')  
    print(response.text) # not your IP address  
  
    tr.reset_identity()  
  
    response = tr.get('http://ipecho.net/plain')  
    print(response.text) # another IP address
```

# Request



```
import requests
```

```
def get_tor_session():
    session = requests.Session()
    # Tor uses the 9050 port as the default socks port
    session.proxies = {'http': 'socks5h://127.0.0.1:9050',
                       'https': 'socks5h://127.0.0.1:9050'}
    return session
```

```
# Following prints your normal public IP
print(requests.get("http://httpbin.org/ip").text)
```

```
# Make a request through the Tor connection
# Should print an IP different than your public IP
session = get_tor_session()
print(session.get("http://httpbin.org/ip").text)
r = session.get('https://www.facebookcorewwwi.onion/')
print(r.headers)
```

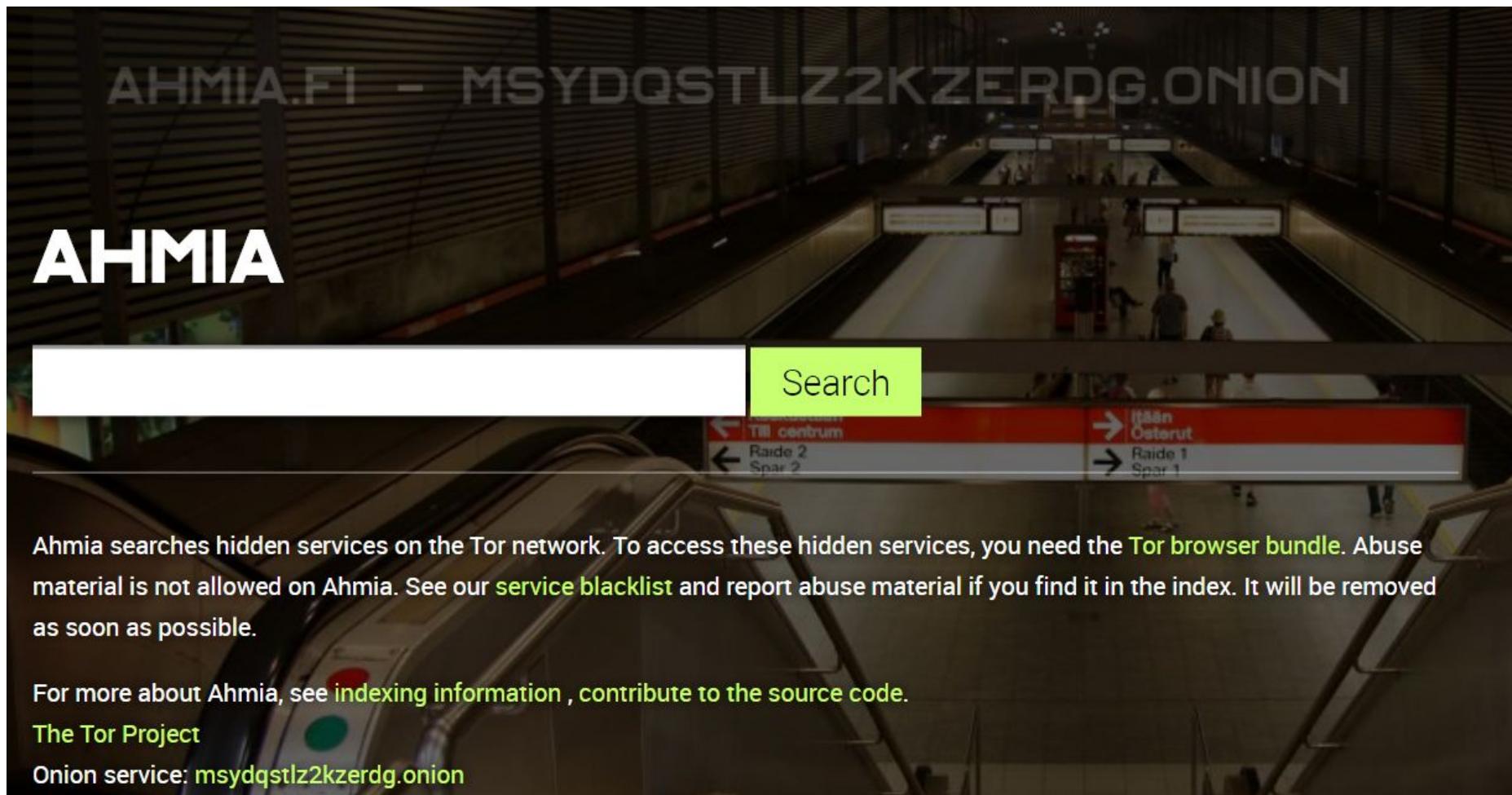
# Analyze hidden services



- 1) Queries to the data sources.
- 2) Filter addresses that are active.
- 3) Testing against each active address and analysis of the response.
- 4) Store URLs from websites.
- 5) Perform a crawling process against each service
- 6) Apply patterns and regular expressions to detect specific content(for example mail addresses)

# Ahmia search engine

<https://ahmia.fi/>



# Torch search engine



<http://xmh57jrzrnw6insl.onion>

A screenshot of a web browser displaying the Torch search engine. The address bar shows the URL "www.xmh57jrzrnw6insl.onion/4a1f6b371c/search.cgi?s=DRP&amp;q=bitcoin". The main page has a large "TORCH" logo at the top. Below it is a search bar with "Search for: bitcoin", a "Search!" button, and an "Extended" link. On the left, there's a sidebar with "Sort by: relevancy | last modified date | title". The main content area displays search results for "bitcoin". At the top of the results, it says "Search for bitcoin. Search results: bitcoin : 15938." and "Results 1-10 of 4748. Search took 4.630 seconds". In the center, a large message reads "Hmm. We're having trouble finding that site." with a scroll bar on the right.

## 1. [Intel chips from last 7 years can be hacked remotely | Deep Dot Web](#) [ 4.993% ]

... Markets List Markets Chart Vpn's Chart **Bitcoin** Casinos BTC Mixer Q&A ~ Ask Here! ... guns as well, its just f... darknet **bitcoin**  
dark web drug sentenced man arrested ...

- <http://deepdot35wmeyd5.onion/2017/05/21...> - 94512 bytes [text/html] - Fri, 02 Nov 2018, 19:57:38 GMT  
[\[Cached copy\]](#)

## 2. [Dav3's Android Botnet -VENDOR- - Wall Street](#) [ 5.058% ]

... on the top we will add it for you. **Bitcoin** Price: \$150 Features: Listen to calls ...

- <http://z2hjm7uhwisw5jm5.onion/viewtopic....> - 13564 bytes [text/html] - Fri, 02 Nov 2018, 19:13:57 GMT  
[\[Cached copy\]](#)

## 3. [Login](#) [ 5.074% ]

... это! » All that you need to know about **Bitcoin**. Всё что тебе нужно знать о Биткоин. » ...

- <http://freexd7d5vpatoe3.onion/forum/inde...> - 12422 bytes [text/html] - Wed, 31 Oct 2018, 23:26:24 GMT  
[\[Cached copy\]](#)

# UnderDir Search engine



Screenshot of the UnderDir search engine results page for "bitcoin".

Search results for bitcoin

LocalBitcoins.com. Bitcoin Exchange #1 in the world

LocalBitcoins.com is a person-to-person bitcoin trading site.  
LocalBitcoins.com is de-centralized, embracing the nature of bitcoin.  
At LocalBitcoins.com, people from different countries can exchange their local currency to bitcoins.

Last check 2019-03-14 11:40 UTC Online (0 sec)

You need to login to report a site [Details](#)

Bitcoin Fog

Bitcoin anonymization taken seriously.

Last check 2019-03-16 10:10 UTC Online (1 sec)

You need to login to report a site [Details](#)

Bitcoin Wash

Last check 2019-03-21 01:20 UTC Online (1 sec)

You need to login to report a site [Details](#)

Navigation: < 1 2 3 4 5 6 7 ... 114 >

# Hidden services



```
import requests

def get_tor_session():
    session = requests.session()
    # Tor uses the 9050 port as the default socks port
    session.proxies = {'http': 'socks5h://127.0.0.1:9050', 'https': 'socks5h://127.0.0.1:9050'}
    return session

def searchUnderDir(address,session):
    for page in range(1,19):
        for searchItem in ['bitcoin']:
            addressWithCriteria = address.replace("CRITERIA_WILDCARD",searchItem)
            #http://underdj5ziov3ic7.onion/search/bitcoin/pg/1
            addressToSearch = addressWithCriteria + "/" + str(page)
            print(addressToSearch)
            response = session.get(address)
            print(response.text)

# Following prints your normal public IP
print(requests.get("http://httpbin.org/ip").text)

# Make a request through the Tor connection
# IP visible through Tor
# Should print an IP different than your public IP
session = get_tor_session()
print(session.get("http://httpbin.org/ip").text)

searchUnderDir('http://underdj5ziov3ic7.onion/search/CRITERIA_WILDCARD/pg',session)|
```

# Search Hidden services



```
while len(deeplinks) <= number_results or length_of_web_links_to_crawl <= iterations:
    try:
        with timeout(10):
            crawl = session.get(deeplinks[iterations])
    except:
        error=1
    if not error:
        crawl = crawl.text
        try:
            soup = BeautifulSoup(crawl, "lxml")
        except:
            print("Error creating 'soup' object")
            os.system("sudo service tor stop")
            exit()
        for a in soup.find_all('a', href=True):
            if len(deeplinks) >= number_results:
                print(" \u033[0;32m LINKS COLLECTED!\u033[0m")
```

```
#process first 5 pages
for page in range(1,5):
    #http://underdj5ziov3ic7.onion/search/bitcoin/pg
    #http://www.xmh57jrznw6insl.onion/4a1f6b371c/search.cgi?s=DRP&q=bitcoin
    search_query = "http://underdj5ziov3ic7.onion/search/"+search_string+"/pg/"+str(page)
    #search_query = "http://www.xmh57jrznw6insl.onion/4a1f6b371c/search.cgi?s=DRP&q="+search_string+"&np="+str(page)

    print("Search query",search_query)
    try:
        content = session.get(search_query)
        content = content.text|
```

## Other tools



POOPAK - TOR Hidden Service Crawler

<https://github.com/teal33t/poopak>

Tor spider

[https://github.com/absingh31/Tor\\_Spider](https://github.com/absingh31/Tor_Spider)

Tor router

<https://gitlab.com/edu4rdshl/tor-router>

# Onnion scan



<https://github.com/s-rah/onionscan>

Summary    Saved Searches

masks3astuf5emnf.onion    Search!

Options

Save Search

Linked Tags

operation-spiky-tomato X

Tag Search Term

Enter Tag... Tag!

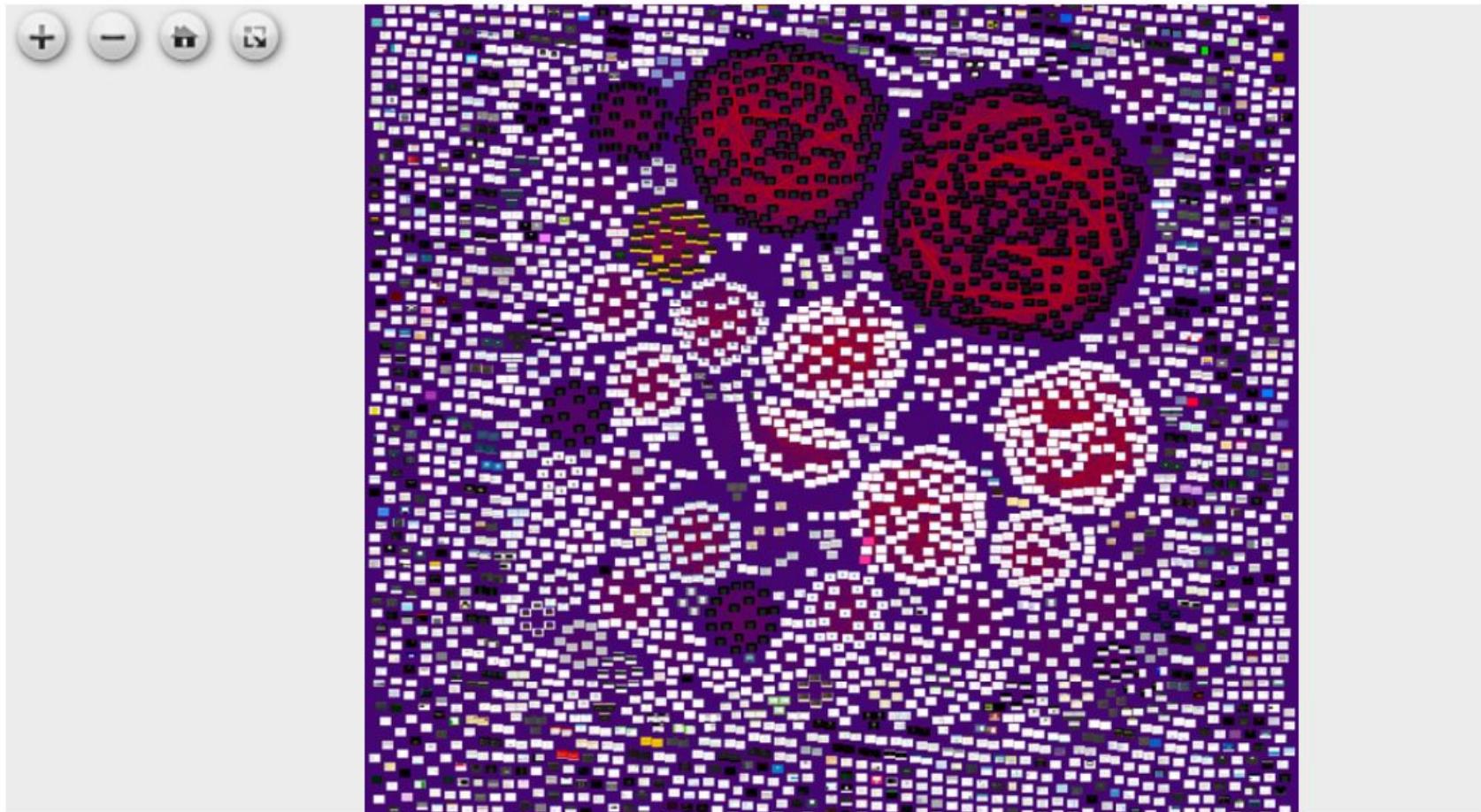
Summary for masks3astuf5emnf.onion (We Sell Masks!) mod\_status pgp

PGP Identities	
Tag Relationships	
Webpage Information	
IP Addresses	13
Co-Hosted Clearnet Sites	12
HTTP Headers	4
Server Information	

GOLANG The Golang logo is a blue cartoon gopher character with a white face and a small tuft of hair on its head.

# Dark Web map

<https://www.hyperiongray.com/dark-web-map/>



# GitHub repositories



<https://github.com/serfer2/python-deepweb>

[https://github.com/jmortega/python\\_dark\\_web](https://github.com/jmortega/python_dark_web)

Get information about Tor network with python stem

```
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
jmoc@jmoc-HP-Compaq-60... x root@jmoc-HP-Compaq-600... x jmoc@jmoc-HP-Compaq-600... x jmoc@jmoc-HP-Compaq-600... x
import sys
import stem.descriptor.remote
from stem.util import str_tools
# provides a mapping of observed bandwidth to the relay nicknames
def get_bw_to_relay():
    bw_to_relay = {}
    try:
        for desc in stem.descriptor.remote.get_server_descriptors().run():
            if desc.exit_policy.is_exiting_allowed():
                bw_to_relay.setdefault(desc.observed_bandwidth, []).append(desc.nickname)
    except Exception as exc:
        print("Unable to retrieve the server descriptors: %s" % exc)
    return bw_to_relay
# prints the top fifteen relays
bw_to_relay = get_bw_to_relay()
count = 1
for nickname, bandwidth in sorted(bw_to_relay.items(), key=lambda item: item[1], reverse=True):
    if count > 15:
        break
    print("%d: %s (%s)" % (count, nickname, str_tools.format_bandwidth(bandwidth)))
    count += 1
```