LINUX VOICE TUTORIAL

TOR: ENCRYPT YOUR INTERNET TRAFFIC

BEN EVERARD

Discover how this anonymity network is helping activists around the globe, and run your own node to contribute back.

WHY DO THIS?

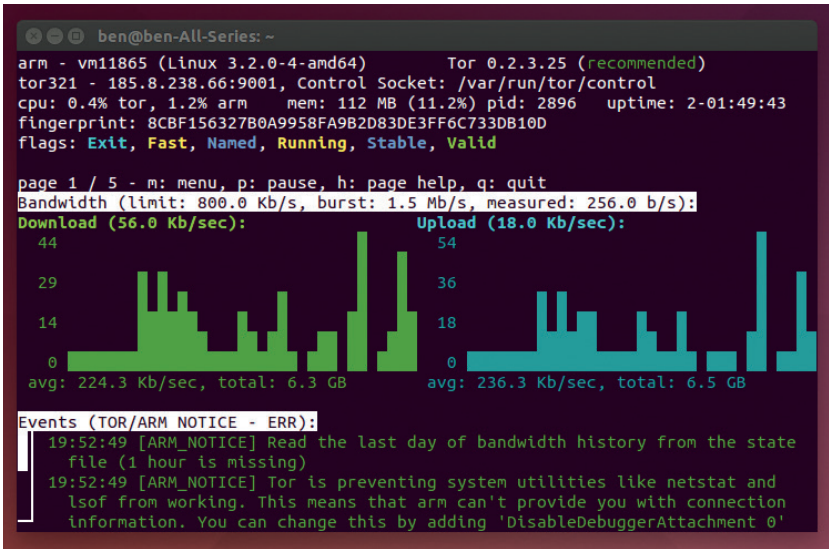
- Keep yourself safe online
- Help whistle-blowers and activists stay beyond the reach of those who would silence them
- Bypass censorship

Imagine you're a blogger complaining about the actions of your repressive government. Or perhaps you've discovered a load of documents that incriminate some powerful people, and you want to get them out to a friendly journalist. In both cases you'd be crazy to use the open internet – it's about as secure as the writing on the back of a postcard, and you'd run the risk of a one-way trip to Guantánamo Bay, or worse. What you'd need is a secure internet anonymising service – like Tor.

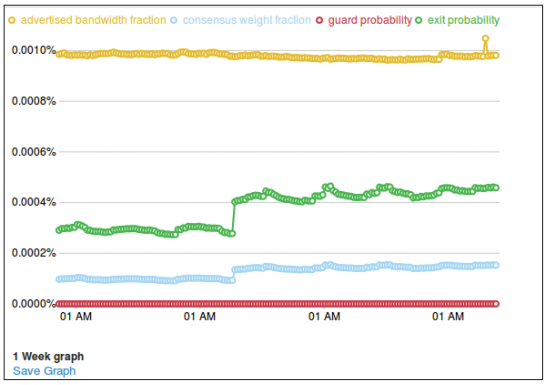
Tor is a global network of computers run by volunteers to provide online anonymity to anyone who needs it. The network is based on the principal of onion routing (the name Tor simply stands for 'The Onion Router'). This means that a connection goes through several encrypted layers, and the router at each layer only knows what is essential to perform the work at that layer.

When you connect to the Tor network the following process occurs: the client downloads a list of all available Tor relays and selects three: one guard, one middle and one exit.

If you then send information through the Tor network onto the internet, it's first encrypted so that only the exit relay can see what the website you're requesting is. Then this already encrypted layer is further encrypted so that only the middle relay knows that it should be sent to the exit relay. This doubly-encrypted layer is encrypted so that only the guard relay can see who the middle relay is. All this encryption is done before it leaves your computer, so:



Arm – the Anonymising Relay Monitor – provides a Curses-based interface that works over SSH to give you all the information you need to keep your Tor node healthy.



The Atlas website can give you lots of graphs on how much data is flowing through individual nodes. Here's a week's traffic through the Linux Voice exit node.

- Anyone monitoring your internet connection can only see you exchanging encrypted information with the guard relay.
- The guard relay only knows your IP address and who the middle relay is.
- The middle relay only knows the guard relay and the exit relay, but not who you are or what website you're requesting.
- The exit node knows what you're requesting off the internet, and who the middle relay is, but not who you are or who the guard relay is.

This process completely separates the content you're requesting from anything that can be used to establish your identity.

The Tor team has done excellent work to make sure that it's easy to use, because the people who need it most (activists and people persecuted by their governments) may not be tech-savvy. All you need to do is download the Tor Browser Bundle from **www.torproject.org**, unzip it, and run the **start-tor-browser** script in the unzipped directory. This will connect to Tor and open a web browser.

Another option is to run the Tails live CD. This can be burned onto a DVD or USB stick and provides a secure Tor environment for web browsing, instant messaging, and other uses.

It's also possible to stay anonymous on the go using Orbot, an app for Android that will link your phone or tablet to the Tor network.

Running the network

For the Tor network to function, it needs people to run the relays that pass the data around the network and

onto the internet. These aren't run by a centralised organisation (since if one organisation controlled a significant number of the relays, it would be able to look at the information in several of these and spy on users), but by a number of individuals and projects around the world.

Linux Voice, for example, currently runs two, the first one being a fairly modest exit node called Tor321. You can see the current status of the node at **http://tinyurl.com/lvtornode**. We also run a bridge node (for details see the 'A Network Under Attack' boxout on page 85).

Running a Tor node is simply a case of installing the **tor** program and setting the appropriate options in the **torrc** file. However, before you start that, you should understand the implications of the options you select.

The problem revolves around the fact that by adding your computer to the Tor network, you're allowing other people to send data through your machine. This data could be anything from someone shopping on eBay to Edward Snowden communicating with journalists in America to someone downloading illegal content (whatever that means in your country).

Diplomatic immunity

This could attract the attention of your ISP and could cause you to get into trouble. However, this will only be visible to your ISP if you're an exit node. If you're one of the first two hops on the Tor network, all the data flowing into and out of your computer on the Tor network will be encrypted so that your ISP (or you for that matter) can't see what it is. This means there should be no legal consequences for people running non-exit Tor nodes in most countries (should you happen to live in a country with restrictive laws governing internet usage such as China or Iran, you should get legal advice before running a Tor node of any sort).

Some people who use the Hulu video streaming service have reported problems with their IP address being blocked when they started running Tor nodes, though this has been quickly dealt with by the Hulu support team.

Provided you have sufficient bandwidth to spare, it's perfectly possible to run a non-exit relay or bridge on a home internet connection. The easiest way to do this is using the Vidalia graphical client. You can find this in most distro's repositories (if you're using Ubuntu, you should add the Tor project's repository by following the instructions at **https://www.torproject.org/docs/debian.html** to make sure you get the most up-to-date version of Tor).

For example, in Debian, you just need to run

```
sudo apt-get install vidalia
```

Then restart the computer to pick up the new user settings, and run **vidalia**. This will open the graphical client and connect you to the Tor network. Click on Set Up A Relay, then check the box marked Relay Traffic Inside The Tor Network (Non-Exit Relay). In the options, you can name your relay, add contact information, and limit the speed if you wish, but these are optional. Click on OK to start your relay running.

In theory, you can run an exit relay from your home internet connection, and a few brave souls do, but most people shy away from letting unregulated traffic into their home as it can cause problems.

The majority of people who run exit nodes do so on a server running in a data centre. However, not all data centres are happy with people running Tor exit nodes on their machines. If you're interested in running an exit node, the first step is usually to find a place that's willing to host it. The Tor wiki provides a list of hosting providers that people have had good and bad service at (**https://trac.torproject.org/projects/tor/wiki/doc/GoodBadISPs**), however, since diversity in all aspects is good for the Tor network, you may want to consider emailing a few hosts and asking if they'll consider using a Tor exit node.

You can rent a VPS (a Virtual Private Server – a virtualised environment on a shared server) to host

Strength through diversity

Diversity is one of the key things that helps keep the Tor network anonymous. That means many things. It means that a diverse spread of relays is important, because by spreading them out across many different networks in many different countries, it becomes much harder to run timing attacks. Similarly, diversity among exit nodes is also important because this means that anyone trying to listen in on all Tor traffic has to listen in more places. A diversity of bridge nodes is absolutely critical to keeping the Tor network open to people inside restrictive countries.

These are all quite obvious areas where diversity helps the network, but less obviously, it's also important to have a diversity of users. If, for example, only whistle-blowers used the Tor network, then there would still be some anonymity, but any website operators would know that any connection coming from a Tor exit node was from a whistle-blower. Only by getting a wide range of users on the network can it offer true anonymity to its users. Because of this, you shouldn't shy away from using the Tor network for fear of using up resources that other people may need more. The sheer act of using it actually makes it more secure for everyone (although you shouldn't run high-bandwidth traffic through the Tor network unless necessary).

Tor and the law

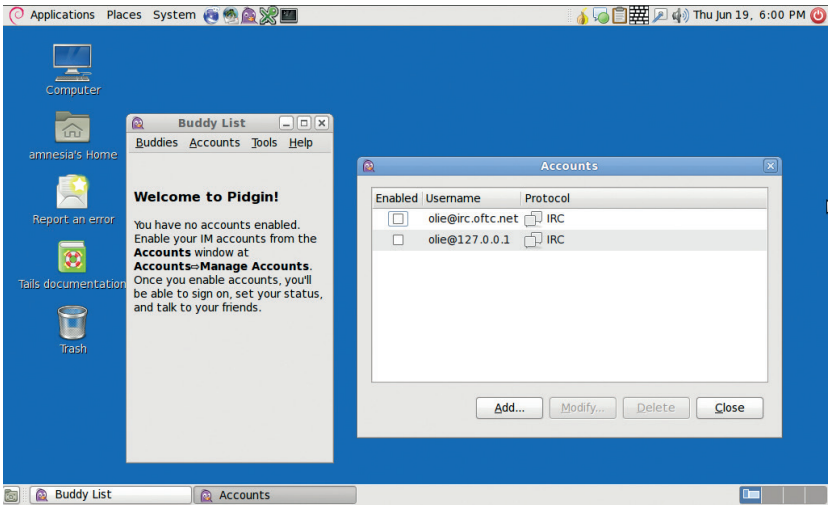
Although there have been several legal controversies surrounding Tor, to our knowledge no one has been convicted for running a Tor exit node. As we're going to press, William Webber has just been convicted in Austria for abetting access to pornographic images of minors after someone downloaded such images through their exit node.

However, the prosecution showed transcripts of conversations where Webber was encouraging the use of Tor for such things, and offering to assist. In other words, he wasn't convicted for running a Tor exit node, he was prosecuted for running a Tor exit node and using it to help people access horrific images.

The Tor project is also being sued in America for allegedly assisting a website accused of purveying "revenge porn". However, this case seems to be built entirely on a lawyer's misunderstanding of what Tor actually is.

This case is being brought against the Tor Project, so it shouldn't have any impact on Tor node operators.

For more information on miss use of the Tor network, see the box out on abuse.



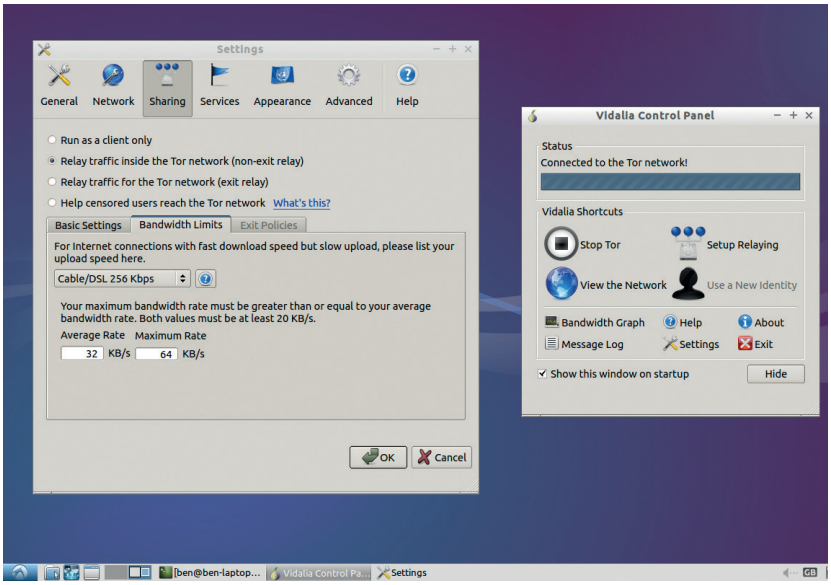
The Tails distro provides more than just web browsing: the Pidgin client is also set up with Tor, to provide anonymous instant messaging.

your Tor node from just a couple of pounds per month, but in general, you get what you pay for, and dedicated servers usually come with much better internet connections, though this does vary from provider to provider. Ultra-low cost ones are likely to be low bandwidth (even if they are unlimited traffic), and may not be stable. It's hard to give a definitive best option, but in general you don't need much hard drive space, and only modest memory and CPU (unless you're going to run a really fast relay). In most cases, the bottleneck will be network speed. If you're unsure about a particular option, the best bet is to try it out. Most hosts provide hosting by the month or sometimes less, so if you find your particular setup needs a bit more oomph, or is costing too much, you can usually switch to a new option.

Command line setup

If you've used Tor previously, you may remember Vidalia as part of the Tor browser bundle, but it now needs to be installed separately.

The biggest difference between setting up a Tor node on a server compared with a desktop is that you don't usually want to use the graphical setup tools. There's nothing to stop you doing this via VNC or an equivalent, but there are command line tools that do the job better in this case.



From a technical perspective, the only difference between running an exit node and a relay is the exit policy listed in the **/etc/tor/torrc** file, so we'll start by looking at this. By default, the exit policy will allow most internet traffic through, but block file-sharing ports and a few ports used by spammers. This will both reduce the number of complaints you receive, and help make sure that your bandwidth is helping web traffic. Our Linux Voice exit node uses this policy.

You can create a custom policy to allow or disallow any ports you like. A more liberal exit policy (stolen from the Destiny exit node) is:

```
reject 0.0.0.0/8:*
reject 169.254.0.0/16:*
reject 127.0.0.0/8:*
reject 192.168.0.0/16:*
reject 10.0.0.0/8:*
reject 172.16.0.0/12:*
reject 94.242.246.23:*
reject *:25
reject *:587
reject *:465
accept *:*
```

This blocks access to any of the local network IP addresses (otherwise a malicious attacker could use your exit node to attack machines on the same local area network), and ports 25, 587 and 465. These are the ports used by SMTP mail servers. Blocking these won't stop a mail client communicating with a server, because that uses a different protocol; but it will stop a computer acting as a mail server and tunnelling through your exit node – so basically, it'll stop email spammers from using your node. Exit policies are public, so you can find out what other people are using by looking up nodes on **https://atlas.torproject.org** or **http://torstatus.blutmagie.de**.

The final line is there to tell it to accept anything not rejected by the previous lines (non-exit nodes have a similar line that rejects everything).

Other than that, it's useful to give your node a name and add a contact email address. Neither of these are essential, but they help with the smooth running of the network, and make it easier for you to check what's going on. An email address will enable the Tor project to contact you if there's a problem.

Donating

If you don't have the time or technical ability to run a Tor node, but still want to contribute financially, you can donate directly to the Tor project itself and help support development via **www.torproject.org/donate/donate.html.en**. Alternatively, you can donate to an organisation that runs Tor nodes, such as **www.torservers.net**.

At Linux Voice, we're currently running a couple of Tor nodes, and would like to upgrade these to handle more traffic. We've pledged to put 50% of our profits towards good causes, and think that the Tor network is just such a good cause. Later in the year, we'll be asking subscribers to vote on where this money should go, and increasing our support of the Tor network will be one option.

A network under attack

Not everyone is happy with the Tor network providing people with anonymous and uncensored access to the internet. Some governments (such as those in China and Iran) have attempted to block access to Tor from within their countries.

The simplest way of blocking access is to get a list of all Tor relays, and stop any packets heading for these IP addresses. When governments realised that they could block access to Tor in this way, the Tor project introduced bridge relays. These are entry points to the Tor network that aren't listed in the main Tor relay directory. They're split up into groups: some of these are available on the internet, but only a few at a time; some of these are available via email; others are distributed via social networks and through trusted contacts.

Governments with large amounts of computer power at their disposal have been able to discover a large number of these bridges. Because of this, it's important for there to be a considerable 'churn'. That means that if you're thinking of setting up a non-exit Tor relay, a bridge is a great place to start. It's also possible to run a bridge for just a few dollars a month by taking advantage of Amazon's free-usage tier in EC2 (see **https://cloud.torproject.org** for details on setting one up).

Another approach that governments have taken to censoring Tor is through Deep Packet Inspection (DPI). This means that instead of finding Tor packets by IP address, they look for data within the TCP/IP stream that signals that it's Tor traffic. Tor attempts to disguise itself by looking as much like Firefox communicating with an Apache TLS session as possible. This disguise isn't perfect, and there is a bit of a cat-and-mouse game going between the Tor project and the western companies that sell DPI equipment to repressive governments. When a differentiator is found, a government can block Tor, then a software update improves the disguise, and service is restored.

Hiding in plain sight

Of course, there's no reason a government can't simply block everything that looks like a secure Firefox communication with an Apache server – except for the social consequences. As we've seen in Egypt and Turkey, such obvious censorship can lead to demonstrations and more.

The next step from the Tor project to make it harder to block is pluggable transport modules. These have created a framework that enables a variety of different ways to connect to the Tor network. For example, there's the Flash proxy (implemented in HTML5 rather than Flash). This is a way of starting a Tor bridge from inside a web browser, so it can be run on a far wider range of computers. In turn this means that the supply of IP addresses is much larger, and changes far more rapidly than with traditional bridges, so it becomes harder to block.

Other pluggable transport modules in the works include ones that try to disguise the traffic as a Skype call, and ways of making the traffic look like an HTTP stream with HTML, JavaScript, etc. As more of these become available, it'll become harder and harder to block them all.

Not all attacks focus on trying to block Tor. In an attack widely thought to be performed by the FBI (although not yet confirmed), malicious code was injected into a hidden service that managed to break out of the Tor browser and get the computer to reveal its actual IP address, and therefore location. The solution to this is simply better software, and much work has been done on browser security in recent years. Currently the Tor browser is based on Firefox – there is theoretically better security in Chrome, although there are some technical challenges to overcome before this can be used.

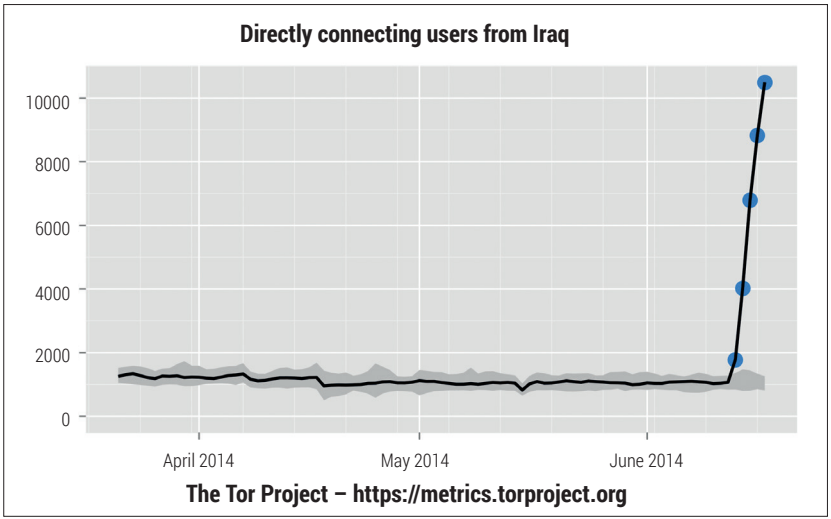
It takes a little while for your node to be picked up by the network, but when it is, you'll be able to find it by searching for its name on **https://atlas.torproject.org**. This will also give details about how it's running. There's more guidance on running an exit node at **https://trac.torproject.org/projects/tor/wiki/doc/TorExitGuidelines**.

The best way to keep an eye on a node running remotely is with the Arm command line tool. If you're using Debian, you can get it with:

```
sudo apt-get install tor-arm
```

It uses the Curses toolkit, so you can run it in an SSH session. Arm has five screens: Graph, Connections, Configuration, Torrc, and Interpretor. We've found it a bit easier to do the configuration outside of Arm, but the Graph and Connections screens are useful for making sure everything is working properly. With a bit of luck, you should soon see traffic flowing through your node (it can take a few hours). After your node's been live for a little while (around a week or two), you will be awarded a stable flag, which is an indication that your node can be trusted to stay running, and not break down in the middle of a communication.

That's all you need to start running your own Tor node. If you haven't run a server before, it's a gentle introduction to the world of server management. We've found it to be one of the easiest network services to run, and the developers deserve a good deal of praise for making it so straightforward.



The Tor project is constantly scanning for censorship events. This graph shows the number of users connecting from Iraq in June 2014 during an Islamist insurgency.

Abuse

Rather predictably, the Tor network is abused by some people who use it to conduct illicit activities. This is unfortunate, but unavoidable without compromising the core values of open access and anonymity. However, this abuse makes up a tiny fraction of Tor traffic (one common estimation reportedly based on an unpublished study by the US Department of Justice puts it at 3% of Tor traffic).

The Tor network also plays a part in fighting cyber crime. For example, the Internet Watch Foundation (a UK organisation that blocks child sexual abuse content) needs to use Tor, as all its IP addresses are blocked by many of the sites they are trying to investigate.

Ultimately, criminals have many methods of staying anonymous, but legitimate whistle-blowers, activists and journalists often have only one: Tor. That's why so many people are prepared to support the network even though it is sometimes used for nefarious purposes.

Ben Everard is the co-author of the best-selling *Learn Python With Raspberry Pi*, and is working on a best-selling follow-up called *Learning Computer Architecture With Raspberry Pi*.