Hidden Service Setup Guide for Newbies

Version .2

So, you've decided to set up a hidden service and join the information underground? That's fantastic, but be aware of what you're getting into. If you're setting up a hidden service, you're probably doing so because whatever you're publishing could put you at risk. Maybe it's legal risk, maybe you might lose your job, maybe your friends might disown you, or maybe you might end up in prison for the rest of your life or worse. It could also be that you're a Tor fanatic and that you're setting up a hidden service to help those who are taking the risk.

Tor is an amazing technology and there's lots of technology that when combined with it can make your hidden service almost bulletproof, but that doesn't mean you can remove risk from the equation. Whatever it is you're doing, you need to accept the risk and the potential consequences. If you aren't prepared to take the fall, you might want to reconsider what you're doing or how you're doing it. Nobody has ever been caught for running a hidden service and those who will be will most likely do so through their own stupid mistakes - Tor won't be at fault. Tor is beautiful and so is the resistance to the current system that is inherently built into it. If you're ready to join the revolution and fall in love, be my guest. All good things in life, all struggles must be won. There is no easy way, this is a conflict. Since you've taken it up to learn how to arm yourself, I'll show you how to use your weapons.

Just as a clarification, I am not a lawyer. I'm not qualified to give legal advice. You should learn about what you're doing before you do it. This guide is not the be-all end-all. You're going to have to use your brain and some common sense if you're going to survive out here. This guide is written for laypersons, but that isn't an excuse for you not to do your research. I didn't write this myself, it comes from decades of research and work by individuals too numerous to name. People have gotten hurt, people have been thrown in jail, and some people have been killed simply for taking the red pill.

I didn't write this guide to help people break laws. If you do something stupid with this, it's your fault. This, like the Tor software, is a tool and how you use it is up to you.

There is no warranty of fitness or accuracy on this guide whatsoever. You are using it at your own risk and if you mess something up, it's not my fault. By using this guide, you agree to hold its author(s) harmless for any damage that may arise as a result.

This guide is anti-copyright. You are free to mercilessly update it, edit it, share it, etc. If you give me attribution, that's great but absolutely not required or expected. If you have any questions/clafifications/ edits to the guide, you can reach me at ringo{at}hackbloc.org. My PGP key is below:

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.9 (GNU/Linux)

mQGiBEniUKIRBADfn8kULsRd3si+zPnVbeVp4C/cjxfOxvPURPjRMDPRZPuDuEI5
QIiMP+lZs0Y1BS/zubrwJ/R+knZW0dfkCbd0IBqhtcci4ZiDXRCNxxYow0MysweG
sbZE0QY4T2u40ffOLs9m/ENiDebUxknTyAg8/Jim9aBdEDgurCc7HCX+iwCghfLh
1POMWQRkXB4zUmXQfp+u+0MD/j5SUN6ct6fH4ex3L/WeIHRA+PZXBEpQv5HCwcYO
9VAtS0KYTtrBePXuhabjmiyhWIVsPHa8A+5RW3ONkK4gQ71E7sh2nu44p0rOSVkz
9/ZQiHVCjxZJNhvCsabIFT2/G8OFo2XPnJ0+8Gfluueb5a/HKArUWHIvkws82kQ5
75RJBACJp436/Bvk/CpKDkIG8v/4dQkyNKhv5AEAbx3jNjdOAxNSK0tBaQAulgCk
GFNkk+wpv6OWaawgQzFh71KvmEswSLObXk+S6WZgC+Epy4XmfzzDG/gIHD0VuBQ+
2D8JzFT/TiDMu6wdYu4kgDg5sO4a5Yzn7xoYMF5YWzXnPKhXi7QacmluZ28gPHJp
bmdvQGhhY2tibG9jLm9yZz6IZgQTEQIAJgUCSeJQogIbIwUJAeEzgAYLCQgHAwIE
FQIIAwQWAgMBAh4BAheAAAoJEFUc7QiIWsvrdtkAn3KtPdxxC/qWmmIFZ4Nc4cFE
as42AJoDwdk/N9I3sPvc91wTTlbsKhoHLrkEDQRJ4lCiEBAAs2JYGr1k1Dgi3DMy

h0ziX+22tIWWyIJoGKWKFspA7nGeniOBodLBvR+POtqqGCh+bkm9I0X/YMF9oVcP
xXBql7H6E4JSgtCk7xtohDpLlfcCpsddVxcJdXYLynTUMcmJtCER0bCNIkTmYoV7
uNXAqmUNAp4zaI70yWsidpAVHme0+sBUYNinfBdlcaMddzslbDtRV7yGKgvW3E5e
hPNTJ0pWF6WJg4VsEOFoP7pldtQ4YWScskvuCk957K4t4Of3QZs13Nn9sQZleFJU
E2L1bxEHuSqY/f1F/pbKmc7in8qkoBBAyhUbzCNxxELdof3uJpBy0pw0468GvSyb
Z4jyh2XFvxFFAcelzc453y9GOylIC0OQczkrzOa6QrIWQSmeCzn/byjLoi+TRFve
usRmJn5H9MJg+k+mG5LJM2mcyQJU2UOPDvSurKmk50vByBED6Qn5CvhXJp18H6Uk
2r+PICG4h8aN9KZpSrMAqYggyKgAxHTlCaQzGCwvJGiX6lx6iIm2GLoqeHdRHZZX
9XognVcbTwUWJkL0LR9nhm5U0GhFGM9eRdLw89C/Z/s1/Q/QLjoDh60qXcYo+vFS
5bJtiT52HnlA002opyi+Zn5mk9aXQiksOJruIdNw1rvJSe+uAIYQeBv+rinxzAyL
4f/p/+vvgnfgkEc2G1hLuGTvWMsAAwYP+gIhIgQ6UwQ0Bu1gyRN88Gs9H0fnQ74Z
RmFXDgUtpn1YrFzFfTNegQh8vvgo1pXV4ZDPc0w9Cs8QHrspnkYrvSymAEmwYtGd
nvnAVVROIJfN5d140Z1FJXCgFp/3m2SAX1omYyN3/5WX9ef1uaYWub48kSdqfHlr
xe8Z15nXQ9E6WMgDtP5jXpfCkAnweW6/WSGRrHlRyBUevCTyRSZ4dwtim0GHsls9
VbfDYWJVxiKWdgjtjg+PfsXrdQG2KICEHXprS9/tYCheWaHP4couXVHDPUNMGK/w
HSYXbr0/xA0i0JHpRzVCDweKZ32hgbYkTXp0U7ArBYLtbfpWlB8uWHFFAIS5yJQL
YMwc8/qFCgl5fUGMk4ZLTgbftQo/sfcOAIPQl2nVjhnvzucj8PgBBaJgH9ORTpW6
89zIzOtfXfju0dq4LC6Xj4h6SA/duh8dEiBzewNJ1FwnlrywvaQjsVdx5+5RolAk
gZKcT4hHCj+s2vCAyF5R70rfKkZkKhMuUzEWc4R4AzbkmI1eTtEl/FJVCzBsJRan
HC+YMgCdf2ujTxvBltytpWrs0nvzFVY6+RyihQsqlV6KeOtDBTv38a8Q5gdARK0j
5og+X3SWHW0p29PSKk6a3NeSB08J0wlXsrNOJ/JXlYw/yIifZdgl6fO8V7rPBoQt
xIQB5UKSXj8YiE8EGBECAA8FAkniUKICGwwFCQHhM4AACgkQVRztCIhay+vXkQCf
beWbtPmJOWbXn+9LEaJTqcN73REAn2MmtesdDs24QjWfZeTfc8dyEZ2n
=O0oE
-----END PGP PUBLIC KEY BLOCK-----

Because this guide is written for newbies, it's not the best way to do it. It's meant to be as easy and secure as possible. If you learn more and work on this knowledge-base for a few years, you'll look back on this guide as ridiculous. You'll realize that it could have been done easier, better, faster, and without such a nice computer. Since you are probably a newbie, let's define some important terms we'll be using.

Administrator - An administrator controls the computer. They can do system updates, install software, and mess with all kinds of internal system things. In the wrong hands (or in the case of a mis-typed command), the administrator account can wreak all sorts of havoc.

Anonymity - Anonymity is the ability to operate without anybody knowing who you are. It is not an absolute, you are simply more anonymous than the average person. Any adversary, given sufficient resources and time, can break your anonymity. If somebody can buy and control every internet router in the world (or even a fair portion of them), tracking somebody through Tor would be fairly easy. Fortunately, few people/organizations have this power and even fewer use it for this purpose. In terms of using Tor or running a hidden service, the goal is to obscure your browsing profile which includes things like your browser configuration and your IP address.

Adversary - The adversary is the enemy, the person or entity you use Tor to defend against. Maybe it is your government, the police, your boss, or your significant other.

Algorithm - A method for encrypting data. It describes how data should be encrypted and decrypted, kind of like a recipe.

Boot - To turn on a computer or operating system

Command - A command is something you tell a computer to do. We'll be issuing them through the terminal aka the shell or command line. After you type enter, the command is executed. For instance, typing “ls” and then the enter key causes the computer to list all the files/folder in the current directory.

Encryption - Encryption takes regular data (emails, files, etc.) and turns it into unreadable data. Only

those who know the secret (a password, private key, etc.) can theoretically access that data in a format that is readable. Encryption is strong these days.

Flag - A flag (also called an option) is something that's added onto a command to change the way it operates. For instance, the cp command (copy) when invoked (used) by typing 'cp oldfilelocation newfilelocation' will copy a file from one location to another. If you want to copy an entire folder, you would type 'cp -r oldfolder newfolder'. The -r is the recursive flag and it tells the copy command to go inside directories. Flags can also have values in this format (usually, but there are a few exceptions) 'command -flag anumberorsometext'

Identity – Somebody or something that an entity claims to be. This could be your name, a pseudonym you use, or the name of a corporation. Identity is important and when you're hosting a hidden service, and should probably be kept secret. Identity is authenticated by some form of credential, like an ID card, a passphrase, or the address of your hidden service. Your real world identity and the identity you use on your hidden service should never come in contact unless that's your goal.

Keyspace - A set of keys that could potentially be private keys (ie keys that could unlock encryption). The more keyspace you have, the more keys must be tried to crack your encryption.

Linux - Linux is an operating system. It's actually just the kernel of the operating system, but we'll call it an operating system for simplicity's sake. Most servers in the world run it (as opposed to Windows) for the simple reason that it works better, is cheaper, is faster, and is more secure. Linux (or at least the variants we'll be using) is free (as in freedom) software. You have the right to use it, copy it, give it to friends, modify it for your own purposes, and distribute those modifications. Being free (as in money) is a byproduct of this. It's designed for communities and built by communities. It's not built by a big corporation or monitored by a government. There isn't one person or entity who can backdoor it. It's ours.

Noob/Newb/Newbie - Somebody who is new at something and lacking or devoid of skill. This is who this guide is written for.

Operating System - An operating system is what your computer runs. The programs you run communicate to the operating system and it handles all of the hard stuff like writing to the hard drive, managing memory, etc. Windows, OS X, and Linux are all examples of operating systems.

Password - A password is used to protect unauthorized access to whatever you're protecting. You use a password on your email account, to enter a secret location, etc.

Passphrase - This is like a password, but longer, harder to guess, and much more secure.

Privacy - Privacy is the idea that a person has things or information that should be kept inaccessible to the rest of the world if the person so desires. Normally these include things like medical records, personal thoughts, and corporate records. Tor extends your privacy by giving you control over what you share and with whom. You get to decide if you want to be identified and if you want to reveal your true IP address. If you decide to post your personal information online, you're giving up a lot of your privacy. Like anonymity, privacy is not an absolute.

Private Key - In encryption, the private key is the secret that is needed (usually in combination with a passphrase) to decrypt information. You don't give this to other people as it's private. If an adversary

obtains this key and your password, they will be able to decrypt your data. If they only have the private key, cracking the password is still a fairly easy process. Divulging your private key will result in your encrypted data being unsafe.

Protocol - A standard procedure that is understood by more than on party. A protocol insures that two different entities can interact and produce a pre-designated result, kind of like a recipe. For instance, when you buy a hot dog, the protocol is that you pay for the food before you get it. This insures that the seller is not ripped off by you running away before paying. If the seller decides to take the money and run, you can always take the hot dog cart. Computers use protocols to communicate information.

Random – Unpredictable or difficult to predict. All modern encryption relies on obtaining random data to make private and public keys which in turn are used to encrypt data. An adversary would have a hard time guessing something random right? Random is also not a binary as some data are more random than others. The 'randomness' of a set of data is called its entropy.

Security - Security is the degree of protection from forces external from or internal to an entity. This could be a fence around a building, laws that protect data, or a firewall on your computer.

Sudo - A command that when used by allowed users executes a given command as if the user were the administrator.

Traffic - Stuff that goes over the network. This could be web browsing, Tor connections, hidden service downloads, or whatever.

Virtual Machine (VM) - A virtual machine is an operating system that runs inside another operating system. It is (often) completely separated from the other operating system. It can't see files in the host operating system, access the internal communications bus for the host operating system, or see the IP address of the host operating system. This is useful for testing programs, sandboxing users, or protecting information about your computer.

Ubuntu - Ubuntu is a distribution of Linux, think of it as a 'flavor'. A group of people took previous Linux flavors and packaged it together with the goals of it being newbie-friendly, compatible with as much hardware as possible, and being flexible.

User - An account usually associated with a single person on a machine. This user can log in and out of the machine and is granted specific abilities

Great, you're this far and about a day or so away from running a hidden service using this guide! This is going to take a while and it's going to take even longer if you have an older machine. I strongly suggest that you grab some snacks and coffee before you start this. As a note, this guide was designed for the Ubuntu 9.04 release. If you're using a different version, you'll have to adapt these instructions. The computer you're running this on should be recent, something made around 2005 or later. If your machine came with XP installed, it may struggle a little. Dual-core processors and processors with hyperthreading will handle the load much better. 64bit machines will as well, but many of the programs I'm expecting you to use do not have 64-bit versions available in the Ubuntu software repositories so you may spend quite some time finding them manually or even compiling them from scratch. You can run a secure hidden service on a machine that came with Windows 98, but it is much more complicated and requires more knowledge than I can fit in a guide that is already turning into a textbook. You'll also need a way to burn CDs.

In order to write this guide, a lot of things had to be assumed. We are assuming there is no all-knowing entity watching the internet. This not true. If you do some research about ECHELON, the UKUSA agreement, or the history of the intelligence trade you'll find this out. Rather, we're assuming that entities capable of being all-knowing are also not concerned with with whatever you're doing. Usually divulging what they know would compromise their purpose and therefore you have little to worry about. We are assuming that encryption will protect you. Again, intelligence agencies like the NSA can probably break encryption but they won't for the reasons stated above. In our model, we're also assuming that the Tor software will never have vulnerabilities that could result in an attacker running remote code on your machine and that, if put at gunpoint, the Tor developers would refuse to put in backdoors or that in such a case somebody would notice. We're also assuming that the Ubuntu operating system doesn't have any major backdoors or that if it did whoever used them wouldn't be interested in you. Assuming all of this is a risk but we're going to do it to keep both of us sane and within the realm of possibilities that reality offers as opposed to the possibilities that tin foil hat land offers.

When you use a machine for a hidden service, it's absolutely critical that you only use it for that purpose. NEVER use it for anything else, especially activities that have connection to you personally. It might be obnoxious to have a computer you can never use, but given the possible consequences it's probably worth it. The same goes for your virtual machine which we'll talk more about later. The more programs you install on your hidden service machine, the more avenues of attack you will create for your adversary.

While this guide seeks to protect you from all reasonable risks, you should realize that there are some attacks this guide doesn't cover.

1. Cold Boot Attacks

One of the problems with encryption is that in order for it to work, your computer has to know the private key and any other information needed for decryption. This information is stored in memory and while memory isn't a good place to store things long term, it does store data for an amount of time from seconds to minutes after your machine has been turned off. An adversary, knowing that they are facing a locked down machine with lots of encryption, may perform a cold boot attack. This involves turning off your computer, spraying your memory with liquid nitrogen (or something to keep it cold), and then recovering your encryption key from memory. Once frozen, data in memory can be retained (and then further reconstructed) for hours. If you feel this is a risk, you need to implement physical security measures that deal with the possible threat. This could be as simple as a laser tripwire on a door that triggers a shutdown.

2. Radio Leakage, TEMPEST, etc.

All electronics create radio interference as a consequence of their operation. While this radio interference is often useless it can also provide valuable information for your adversary. For instance, the radio interference generated by keyboards can divulge your passwords to an adversary sitting across the street from your house. RF shielding is the only solution for this problem and involves surrounding your machine in some type of metal. This isn't all though, as the power pull generated when you use the

keyboard, etc. can also be monitored through your wall socket. I don't know of any solutions to this. One idea would be to lock your machine in a box with a UPS to filter the electricity and a security scheme similar to the one used to prevent cold boot attacks but I'm not sure how effective this would be.

3. Physical Security

An adversary may put a camera, microphone, or some other recording device in the room with your hidden service machine. If they capture your encryption passphrase, your data will be compromised. Recently the FBI and Secret Service used this technique against a bust of the ShadowCrew carding board and it's been used for a long time by both law enforcement and intelligence. While using a blanket will deter a camera, the audio generated by your keyboard may not be sufficiently muffled to stop a microphone from knowing what's going on.

4. Traffic Correlation

If your adversary suspects you run a hidden service, they can watch your internet connection and try to use traffic analysis to determine if the hidden service is run on your network. If your adversary downloads a few 50 megabyte files from your server and every time around 50MB of encrypted traffic goes across your network, it's pretty good evidence. Combine that with shutting off the power to your machine and watching the hidden service go down and you've got somebody who knows what's going on. There are creative ways of dealing with this such as cover traffic, UPSs, redundant servers, and physical security.

Installing Ubuntu

The first thing we need to do is download the version of Ubuntu this guide was written for. In the event that you're doing this in a time so far away that Ubuntu 9.04 is an outdated version, you can use the newest version but realize that you might have to change some of what this guide tells you to do to get it to work. The fastest way to get the most recent release is via BitTorrent. Not only is this usually faster than downloading the file directly from Ubuntu, but it also reduces load on their servers. You'll need a BitTorrent program to do this. If you're doing this on Windows, I suggest Vuze (vuze.com) but a lot of people like uTorrent (utorrent.com). It doesn't really matter what program you use.

You can grab the Ubuntu 9.04 Alternate Install CD at:
http://releases.ubuntu.com/9.04/ubuntu-9.04-alternate-i386.iso.torrent

While you're waiting for it to download, it might help to look at some basic Linux commands. There's a good tutorial at http://www.reallylinux.com/docs/basic.shtml

Once you've downloaded the file (hopefully on a different computer to make your life easier), I encourage you to leave your BitTorrent program open (let it seed) so that you upload the file to other people. If there was nobody to help you get this file, you wouldn't have it so please do your part and seed for a day or two.

You'll need to burn the ISO image to a disk. Most commercial CD burning programs, such as Roxio or Nero will do the trick. If you don't have one, you can get a free ISO burner from http://www.magiciso.com or go to http://download.com and just search for ISO Burner.

Meanwhile, on the machine you're installing your hidden service on...

1. Put the CD into your computer and then restart. On most computers, this will cause it to boot off the CD. If it doesn't (ie goes into Windows, etc.), you'll need to change the 'boot order' in your BIOS. You can Google on how to do this. Basically, right when you turn your computer on (and before windows, etc. loads), just quickly cycle through keys F1-F12, esc, and del. If the computer starts beeping at you, you usually have to wait a bit more. Make sure the CD-ROM boots before the hard drive/hdd.

2. If you booted off the CD, choose the language you want (using the up/down keys and enter) and then select "Install Ubuntu"

3. You'll be asked to choose a language again at the next screen. Keep in mind that (without additional precautions), your webserver and any documents/emails/etc. you produce on this machine will indicate which language you chose to a trained eye.

4. Next you'll be choosing your time zone. Again, this is something to think about wisely as people will be able to tell what time zone you choose through your web server, other services, etc. Think about where you want your adversary to think you are ;)

5. You'll be asked if you want your keyboard layout detected. I suggest you choose no unless you've got a very weird keyboard. Then choose your keyboard's most likely origin and you're set.

6. If you get an error saying that "network autoconfiguration failed", it's probably because it doesn't recognize your wireless card or that you're not plugged into the internet. Just go to 'continue' and then 'do not configure network at this time'. This will get sorted out later.

7. This is where you choose your computer's name. Usually people name their computers after themselves, but this wouldn't be a good idea here. I suggest something generic like "computer" "laptop" "desktop" etc. You can also choose something deliberately deceptive to throw off an adversary who may obtain this through leaks such as “windows machine”.

8. The time zone seems like an obvious choice but again consider what an adversary could gain from knowing it.

9. CAUTION: THIS IS THE POINT IN THE SETUP PROCESS WHERE EVERYTHING GETS DELETED FROM YOUR HARD DRIVE. Before installing, you might benefit from "wiping" your hard drive as opposed to "deleting" the stuff on it, which is analogous with removing all the highway signs to New York City and hoping nobody will find it. The city is still there. A good wiping program called Darik's Boot and Nuke (dban.sourceforge.net) is available for those who are interested.

This is where it can get tricky. If you don't know what you're doing, it's best to just go with "guided - use entire disk and set up encrypted LVM". You can also make your own custom encrypted LVM but this can always be changed later.

After you select this, it will ask you which hard drive to partition. If you have multiples, you'll need to

know how big each of them is and which you want to install Ubuntu on. If you only have one (most people), just select the only one that's available.

To finish this step, just select yes at the next screen.

10. Encrypting Your Hard Drive
This is where you choose the passphrase to encrypt your hard drive. Under current US law, you cannot be forced to give up your encryption passphrase in a criminal proceeding (but that won't stop the judge from jailing you for contempt or using other illegal tactics to entice you) however in a civil proceeding if encrypted data is subject to discovery you may have to. Under UK law, you are required to give up your password in certain circumstances but there's nothing that can be done if you forget. In some countries, you could go to jail for life or worse for not giving up your passphrase.

If you're serious about this, you'll choose a good one and only type it in under a blanket. Authorities have been known to put cameras in vents, etc. to catch passwords.

DO NOT USE:
Words found in the dictionary (or combinations thereof)
Words or phrases that could easily be associated with you (your birthday, personal mantra, etc.)
Short passphrases

DO USE:
Letters, numbers, symbols, spaces, uppercase, and lowercase
A long passphrase

DO NOT:
Write down this password (unless to temporarily remember it and make sure you keep a damn good eye on it)
Share the password with anybody else unless they “need to know” it to administrate the server

For more information about passphrases and how to choose a good one (which is really important if you want your data to stay private) see these links:
http://www.queen.clara.net/pgp/pass.html
http://www.iusmentis.com/security/passphrasefaq/
http://www.unix-ag.uni-kl.de/~conrad/krypto/passphrase-faq.html

11. Next it may ask you the "amount of volume group to use for guided partitioning". Just use what it suggests as the default.

12. Tell it to write the changes to disk and it will start re-partitioning your hard drive. This basically means it's setting it up so you can put data onto it, dividing it up into the proper chunks and installing the file system (which keeps track of where files physically are on the hard drive, among other things). It will also start installing Ubuntu. This may take a long time, especially on computers with slow drives, big drives, or a slow cd reader. Be patient.

13. Choosing a Username
Once your system is installed, you'll need to configure it. The first thing you'll have to do is choose a username. It's best if this can't be guessed, so choose something random but also consider what your adversary might know about you if they saw it. When you choose your password, try and make it as

secure as your passphrase but don't make it the same. Also consider what somebody might know if they cracked your password. Under our model, the adversary will never be able to find out this account information but it never hurts to be safe.

14. Encrypting Your Home Directory
This doesn't really offer any additional protection and will just slow your computer down. Your entire disk is encrypted anyways.

15. Install Software
Now that you've answered a few questions, it's back to watching the loading bar.

16. Set the Clocks
Once you're done installing software, it's going to ask you about your system clock. In most cases, choosing Yes is the best option here. After this the CD will eject. You should remove it and then select continue.

Configuring Ubuntu

Once you've logged into your system, the first thing you'll want to do is select your software sources and update your computer. Go to System> Administration> Software Sources. You're presented with a list of software sources from which you can update your programs and install new ones. You should make sure that "Canonical Supported" software is enabled. If you have odd hardware (mainly laptops), Ubuntu will need special drivers which aren't open source. In this case, make sure you also enable "Proprietary drivers".

One of the things that makes Ubuntu so powerful is its community repositories. These contain programs and updates that are contributed by community members (other Ubuntu users). This is nice because it allows you access to lots of software but it's a security risk because you don't know who is delivering it to you or making sure security updates are available. Anybody can add any program to this list (for instance, there was an insecure outdated version of Tor in there for years), meaning that theoretically somebody could put a trojan, backdoor, etc. in there and you might accidentally install it. I suggest turning off these repositories and manually enabling them when you need specific programs. Everything that's installed on your system right now will update through Canonical (which we're assuming for the sake of simplicity is 100% trustable although this obviously isn't true).

Next go to the Updates tab and select "Install Security Updates without Confirmation". Unless you plan on sitting by your computer waiting for updates, this is the best thing to do. It will insure that your software is as secure and up-to-date as possible.

Now, go to Applications > Accessories > Terminal and type the following commands (followed by enter)

sudo gedit /etc/apt/sources.list

This will allow you to edit your software sources manually. Because Ubuntu's software repositories don't contain an up-to-date version of Tor, we'll be using the noreply.org repositories which are updated on a regular basis. Now, add the following two lines:
deb http://mirror.noreply.org/pub/tor jaunty main

deb-src http://mirror.noreply.org/pub/tor jaunty main

Now exit and save the file. Back to the terminal. Type this command:

gpg --keyserver keys.gnupg.net --recv 94C09C7F
gpg --fingerprint 94C09C7F

This should show you some text, mainly this:
```
  pub   1024D/94C09C7F 1999-11-10
        Key fingerprint = 5B00 C96D 5D54 AEE1 206B  AF84 DE7A AF6E 94C0 9C7F
  uid       [ultimate] Peter Palfrader
```

If it looks vastly different, something has probably gone wrong. Now, enter this command:

gpg --export 94C09C7F | sudo apt-key add -

This insures that when we download Tor, we're actually getting Tor and not a program that somebody has injected between us and the server we're downloading it from. This somebody could be your internet provider, somebody who has hacked into the software repository, etc.

18. Update Your Software
Go to System> Administration> Update Manager and click "check". You should have lots of updates available, so click "install updates". Depending on your internet connection and your computer's speed, this could take a long time. You may have to restart afterwards depending on what updates are available.

19. Install Tor
Now it's time to install Tor. Go to Applications> Accessories> Terminal and type the following command:

sudo aptitude install tor

Say yes to whatever it asks you. Great! Tor should be installed now.

20. Install Privoxy
Unfortunately, there is no good version of Privoxy in the Ubuntu 9.04 repositories so we have to add it manually. Go to privoxy.org, click on 'download recent releases', click on 'Debian' and download the i386/x86 version. Run this file once you've saved it and click install.

Preparing The Virtual Machine

A virtual machine is a complete operating system that runs inside another operating system. We will use this to protect your identity. This way, even if somebody hacks into your hidden service, they won't be able to find out your IP address, what's on your hard drive, or any other sensitive information. Instead, they'll just land in an empty sandbox that has ONLY hidden service things. It's important that you only use your virtual machine for your hidden service and NOTHING ELSE. Tor will run on the host machine. Tor needs to access the internet, but your hidden service only needs to access Tor. In this way, Tor can access the internet, connect to tor servers, etc. but the machine with your actual hidden

service can only communicate through Tor. This removes the risk that an attacker can force your server to divulge its IP address and therefore it's location/operator by requesting external files.

Open up the terminal (you should know where it is by now) and type the following command: (If you haven't enabled community repositories, you'll want to do so before issuing this command.)

sudo aptitude install qemu

Now it's time to restart!

You'll also need to grab a copy of Ubuntu 9.04 Server. I suggest you download this through the torrent they provide at:
http://releases.ubuntu.com/9.04/ubuntu-9.04-server-i386.iso.torrent

Please don't just be a leech and download. I suggest downloading these files and then uploading to other users so they can get it as well. A good general rule of thumb is to "seed" (share) until your share ratio is 1.5 or you've been seeding for 48 hours, whichever comes first. You can always run transmission, Ubuntu's Bittorrent program, later and it will remember what's up.

Once you've installed the software that's needed to install the virtual machine, you'll need to restart. I'll be here when you come back.

In order to keep the virtual machine safe, we're going to install Truecrypt. Ubuntu's encryption (which we used to encrypt your hard drive) is fairly weak in terms of the grand scheme of encryption options. It's also not deniable. Anybody looking at your hard drive can conclusively prove it's encrypted. Depending on where you live, you may be legally compelled to give up the password or a rubber-hose attack (imagine what somebody could do to you with a rubber hose) may cause you to give it up. It uses AES by default, which is approved for classified data in the United States if I remember correctly.

Encryption isn't foolproof, it's a deterrent -- something that will make your adversary work harder. Every encryption scheme people have devised has eventually been broken, and AES will be no exception. Right now, AES is still very secure. I believe Ubuntu uses 128-bit encryption. According to the National Institute for Standards in Technology (nist.gov), if you assume that every person on the planet owned ten computers, and that there are seven billion people on the planet, and that each of these computers can test 1 billion possible keys per second, and that on average you only need to test 50% of the possible keys to crack a 128-bit encrypted file, then it would take the entire world 77,000,000,000,000,000,000,000,000 years to crack a 128-bit key. This example is taken from
http://www.seagate.com/staticfiles/docs/pdf/whitepaper/tp596_128-bit_versus_256_bit.pdf
That's assuming you chose a truly random passphrase and that the adversary guesses your key in a random order. There's always a chance they could guess the key the first time around, it's all a game of chance. Additionally, there have been some attacks published about AES that reduce the keyspace (the amount of keys that need to be guessed in order for somebody to crack the correct one), so AES is probably on its way out.

TrueCrypt is an open-source encryption program. It works by creating 'volumes', which show up on your computer as separate drives. You can read/write to them like any other hard drive. It has a few very important features that Ubuntu's default options don't have. For one, it's deniable. There's no way (that anybody has figured out) to prove a Truecrypt file is actually a Truecrypt file. It could be just a bunch of random data. Another important feature is 'hidden partitions'. These enable you to create an

encrypted file that actually has two separate volumes with separate passphrases. In one, you can put sensitive-looking information should you ever be forced to divulge your passphrase. In the other one, you can put the actual sensitive information and there's no way to prove that a hidden section exists. Additionally, Truecrypt features 'super encryption', also known as cascading encryption. This means that your data is encrypted two or three times, not just once. This means that even if an adversary guessed a private key that worked, they'd have to guess more and they wouldn't know if that key was correct or just mathematically correct. The final important feature is that it has no default encryption algorithm. With Ubuntu's full-disk encryption, an adversary knows the algorithm the drive is encrypted with and what the keysize is. In Truecrypt, there's over a dozen combinations, forcing your adversary to spend much more time cracking it.

It's worth discussing quantum computing here. All modern encryption systems rely on the fact that factoring prime numbers for large numbers (that have 128 digits, for instance) is extremely difficult. It would take an average computer billions of years to factor a 128 bit key. With a quantum computer, you ask it to factor a 128 bit key and it gets you the answers within seconds. Traditionally, intelligence agencies have been at least a decade ahead of academics. Right now academics are starting to build very basic quantum computers (they aren't computers yet, they're just the basis for doing math using quantum computing) and I would put money on the idea that the NSA already has quantum computing. Needless to say, if you're fighting the NSA you've got bigger concerns than your computer's encryption software.

So, after much discussion, let's finally download Truecrypt. Go to truecrypt.org, click on download, and get the Ubuntu x86 version. One unfortunate part of the Truecrypt website is that it doesn't support SSL. This means that you can't verify that the truecrypt.org server is the actual truecrypt.org server. It could be your ISP, the Chinese firewall, etc. The site provides a PGP signature for verifying the downloaded file, but if you're getting that PGP signature in an unauthenticated manner, it won't do much good. One way to verify the files is to get an "md5 sum". This is way of making a unique 'signature' of a file. I downloaded Truecrypt (version 6.2a) through two different Tor servers and got this md5 sum:
7f16f069416b10b4455a7457a625771b

You can check the md5sum by opening the terminal and going to the directory where you saved the file using the following command:

cd /directory

It is probably in /home/user or /home/user/Desktop. Then type "md5sum filename" and it will print out the file's signature. Also realize that you probably got this guide in an un-authenticated manner.

Now, open up that file, extract its contents, and open the file you've extracted. Install it by clicking install (duh). Once this is done, open Truecrypt. It should be under your programs. If not, you can go to the terminal and type truecrypt. I suggest you use the 'hidden volume' feature, but I'm not going to explain to you how to do it. You have to do some of the work yourself. Truecrypt has some wonderful documentation available at http://www.truecrypt.org/docs/. The hidden service has to go in its own encrypted volume because we don't want the main (host) system knowing about it and vice versa. Tor is in a Truecrypt partition because we don't want an attacker with the main hard drive encryption key to be able to find out the address of your hidden service.

Once you've got Truecrypt open, follow these steps:

1. Choose "create volume"
2. Choose "create an encrypted file container" and go to the next screen
3. Choose "standard Truecrypt volume" and go to the next screen
4. Click "select file"
5. Navigate to the directory where you want the encrypted volume to live
6. Type in the file name you want. If you choose something like 'truecrypt.file', it's going to be obvious that it's a Truecrypt encrypted container. Something like "encrypted.test" or "hard_drive_image.raw" might be a little more deniable. If you want to have fun, create a few of these so your adversary won't know which (if any) to try cracking.
7. Click next
8. This is where you choose an encryption algorithm. I would suggest using multiple ones, but it's up to you and it doesn't really matter which one you pick. If you care about speed, hit "test" to see which ones will run best on your machine.
9. Click next
10. This is where we choose the size of the encrypted volume. For your hidden service machine, consider how many files you'll be storing and how big they'll be. I suggest at least 10GB. You can always change this later.
11. This is where you choose your passphrase. Make it as complicated, long, and random as possible while still being able to remember it later.
12. At this stage, you can also add a keyfile. This is a good idea if you think your adversary will never find your keyfiles. You can store them on your hard drive or on an external device (flash drive, etc.) in a secure or hard-to-find location. While it doesn't actually work like this, this example will explain it's effectiveness -- Using a keyfile is basically like using an entire file as your password in addition to your actual password, making it much more difficult to crack. It takes a while to guess a 20 character password, but even longer for a 5000 character password. Random files are the best, but you can use any file, including a huge zip file or the Ubuntu install CDs (not recommended, since they might expect this of hidden service operators who used this guide).
13. Click next
14. Choose "I will store files larger than 4gb on this volume". While you may never actually do this, it's important to keep it open as an option. This isn't something you'll want to have to change in the future.
15. Click next
16. Now you have to choose your file system. A file system is how the operating system finds out where files are physically located on a hard drive. FAT32 was used on Windows 98-era machines (and is still used on most flash drives). The problem with FAT32 is you can't store files larger than 4gb and if your system crashes without notice (power failure, etc.) there's a good chance it might not recover. Ext2 and 3 are better options for recovering from this kind of a disaster and you can store files larger than 4gb. Also, you can play more around with the permissions (only allowing certain users/programs to access certain files). I'm assuming you don't choose fat32 here because it's usually a bad choice.
17. Click next
18. Choose "I will use this file on other platforms". Again this is one of those things you'd rather have the option to do because if you don't, changing it will be a big pain.
19. Click next
20. Un-check "show" and move your mouse around a lot, for a while. It may also help to surf the web, open big files, type random stuff, etc. I'm not sure though. Once your hand gets tired or your patience runs out, click on format. This will take seconds to hours. It took half an hour on my test machine.

Now it's time to set up Tor for use with our hidden service. Normally Tor stores everything in /var/tor, but we'll be storing everything in our encrypted Truecrypt volume. Unfortunately, if we just edit Tor's

configuration file to store things where we mount the encrypted volume (where it shows up when we open it), it will provide pretty strong proof additional encryption is used on your machine. It wouldn't be much of a stretch to blame the Truecrypt volumes. The way to get around this is to make a script inside the encrypted volume that changes Tor's settings when it's opened. First though, we've got to open the encrypted volume.

Open up truecrypt and click "select file". Pick your encrypted volume you'll be using and hit "mount". Enter your passphrase, key files, and whatever else you need to. Normally Truecrypt mounts things to /media/truecrypt1, meaning you can access the volume at that directory.

Go back to the terminal and type these commands:
sudo cp /etc/tor/torrc /media/truecrypt1
sudo chmod 777 /media/truecrypt1/torrc
gedit /media/truecrypt1/torrc

Awesome, now we're editing the Tor configuration file. Add the below:

disable logging
log notice file /dev/null
use encrypted data directory
DataDirectory /media/truecrypt1/tor
set up the hidden service, 5022 can be anything you want but make sure you specify it right in the Qemu setup
HiddenServiceDir /media/truecrypt1/tor/hidden_service
HiddenServicePort 80 127.0.0.1:5022

This sets up a hidden service and redirect all traffic to it to 127.0.0.1:5022. On port 5022, Qemu is listening and will forward all of that traffic to your virtual machine (hidden service). Go to your Truecrypt volume and create a folder called Tor, which is where all of Tor's data will be stored including your hidden service's key and address. As a final measure, let's make sure Tor can write to the folder by typing sudo chown debian-tor /media/truecrypt1/tor.

The way we'll be making sure that your hidden service can only send traffic through Tor is by making a user that has all of their connections forwarded to localhost. This means that you can run programs over Tor from the hidden service if you set the proxy correctly and that if you don't, the connection will just get dropped. This means that if you're running any other software on your host machine that is listening on any other port, it will be accessible to your hidden service. You may need to block connections to it via iptables or completely disable it. In the example commands, I created a user called torify. If you give it a different name, be sure to change the commands accordingly. To create a new user, go to System> Administration> Users and Groups and select new user. You'll need to give them their own password. Go over to advanced and remember what it says in “user ID”. Once this is done, go to the terminal and run these two commands replacing id with the user id you found. This will allow the torify user to run the kvm modules which may speed up your virtual machine.
sudo adduser `id -un` libvirtd
sudo adduser `id -un` kvm

Once we've got your new user set up, let's install the software that will run your virtual machine.

In Linux, iptables is the best and most popular firewall (technically it actually isn't a firewall, but we'll

call it one for the sake of simplicity). We'll be using it to make sure your hidden service only uses Tor and can't communicate to the outside world any other way. If you're plugging your computer into the internet, your network interface is eth0. If you're using wireless, it's probably wlan0. Modify accordingly for the below. Anything starting with # is a comment to explain what's going on and everything else is a command which you should run! This is based off of some discussion on the or-talk mailing list. If you're looking for more explanations on how this works or other sample commands, see http://archives.seul.org/or/talk/May-2009/msg00067.html

```
#redirect all of torify's traffic to localhost
sudo iptables -t nat -A OUTPUT -m owner --uid-owner torify -j DNAT --to-destination 127.0.0.1
#allow vm to access privoxy, tor
sudo iptables -A OUTPUT -o lo -m owner --uid-owner torify -p tcp --dport 8118 -j ACCEPT
sudo iptables -A OUTPUT -o lo -m owner --uid-owner torify -p tcp --dport 9050 -j ACCEPT
#allow tor to access vm
sudo iptables -A OUTPUT -o lo -m owner --uid-owner debian-tor -p tcp --dport 5022 -j ACCEPT
#if we allow it outgoing, allow it incoming and don't interfere with prior connections
sudo iptables -A INPUT -p tcp -m state --state ESTABLISHED -j ACCEPT
sudo iptables -A INPUT -p tcp -m state --state RELATED -j ACCEPT
sudo iptables -A OUTPUT -m state --state ESTABLISHED -j ACCEPT
sudo iptables -A OUTPUT -m state --state RELATED -j ACCEPT
#don't let anything access vm on localhost
sudo iptables -A OUTPUT -o lo -p tcp --dport 5022 -j DROP
#don't let torify snoop around on listening localhost ports
sudo iptables -A OUTPUT -o lo -m owner --uid-owner torify -j DROP
#don't allow external machines to access vm
sudo iptables -A INPUT ! -i lo -p tcp --dport 5022 -j DROP
#drop all torify traffic failsafe and protocol agnostic
sudo iptables -A OUTPUT -o lo -m owner --uid-owner torify -j DROP
```

To make sure the rules stay after reboot, type the following command:

```
sudo iptables-save
```

Next we'll need to make sure Privoxy works with Tor. Type this command:

```
sudo gedit /etc/privoxy/config
```

Delete everything in the file and replace it with this:

```
# Generally, this file goes in /etc/privoxy/config
#
# Tor listens as a SOCKS4a proxy here:
forward-socks4a / 127.0.0.1:9050 .
confdir /etc/privoxy
log /etc/privoxy
#actionsfile standard  # Internal purpose, recommended
#actionsfile default   # Main actions file
#actionsfile user      # User customizations
#filterfile default.filter

# Don't log interesting things, only startup messages, warnings and errors
```

```
#logfile logfile
#jarfile jarfile
#debug   0    # show each GET/POST/CONNECT request
#debug   4096 # Startup banner and warnings
#debug   8192 # Errors - *we highly recommended enabling this*

user-manual /usr/share/doc/privoxy/user-manual
listen-address  127.0.0.1:8118
toggle  1
enable-remote-toggle 0
enable-edit-actions 0
enable-remote-http-toggle 0
```

Now, type this command:

```
sudo /etc/init.d/privoxy restart
```

You'll also probably want to lock your screen while you're not at your computer. This will stop your adversary from accessing your computer while it's on but won't stop cold boot attacks. To do this, run the following command:

```
sudo aptitude install xscreensaver
```

This installs the xscreensaver. Unless you want to manually start the xscreensaver program every time you turn on your computer, you'll probably want to add it to your startup list/fluxbox init file which we'll discuss later. The command to start xscreensaver is just xscreensaver and to lock it use the command xscreensaver-command -lock.

Speeding Up Your Machine

Running a virtual machine takes a lot of memory, processing power, and disk space. Before we install the virtual machine, it would be beneficial to slim down Ubuntu a bit so we've got more power to spare for the virtual machine. I'll be asking you to restart a lot while we make these changes. The reason for this is that if a change breaks your system, you'll have a much easier time figuring out which change it was and putting it back to the way it was before you made it. If you absolutely *have* to restart, I'll be sure to notify you otherwise it's probably safe to not restart if you want to save time. Most of these changes are small, but some are larger. Just as a reference, the system I used was a Dell with a Pentium 4 Processor and 244 megabytes of ram (actually 256 but Ubuntu doesn't seem to realise this). When I logged in and ran the System Monitor (with nothing else running), I was at 18-19% CPU Usage and 109MB of memory usage. If you installed a full desktop in your virtual machine, you can make these changes there as well. If you're doing these changes logged in as your 'torified' user, you'll have to run this command first:

```
export http_proxy=http://127.0.0.1:8118
```

This tells your user's account to use Tor to access the internet. If you don't do this, you won't be able to access the internet.

1. Disable Startup Applications and Services

Go to System>Administration>Startup Applications
Disable Bluetooth (you'll probably never want this, especially considering how insecure it is and you most likely don't have a computer with bluetooth anyways)
Evolution Alarm Notifier (unless you'll be checking your mail with evolution and using it as an alarm, you can safely disable this)
Print Queue Applet (We want to disable printing because quite frankly you won't be needing it)
Remote Desktop (You'll also probably never need to administer this computer from a remote location. If you do, there are better and safer ways to do so such as ssh)
Visual Assistance (Unless you have a disability which requires this, you can safely disable this)

System>Admin>Power Management
Select NEVER put the computer to sleep (You don't want your hidden service going offline because your computer went into sleep mode)

System>Pref>Display
If possible, choose a lower resolution. This requires less memory and CPU to manage. As a rule of thumb, go as low as you can go without making it unusable. After all, this is a server and you won't have to be on it that much.

Go to System>Pref>Appearance>Visual effects and disable any which are enabled.

In System>admin>services, make the following changes:
Automated crash reports off (You absolutely want this off. Crash reports (data sent to a program's developer when it malfunctions) often contain sensitive information such as log information, system configuration information, and occasionally even memory dumps (a copy of what is stored in your RAM) which are extremely dangerous as they can contain passwords and even the private key used to decrypt your data!
Bluetooth device management off
Printer service off (Again, you won't be using the printer. Additionally, this should disable CUPS. CUPS has a web interface which is accessible from localhost, including your hidden service VM. It's better that nobody is able to access information about your system)
Remote backup server off (You shouldn't need this)

Now, open a terminal and type "sudo gedit /boot/grub/menu.lst". Find where is says "splash" and take it out. This will cut down on the time it takes to boot your computer. The downside is that this means your computer will print all sorts of messages to the screen that the splash screen (Ubuntu logo) normally hides. If somebody watches your screen during the boot process, they could gain a significant amount of information. While passwords won't be printed, information about system internals and logs may be.

It's probably about time to restart, let's see how fast we can get it.

Now that we've changed some basic preferences, let's dig a little deeper into the system.

Most programs on your system don't run all on their own, they use external programs and information called 'libraries'. Normally, each time the program runs, it has to figure out where these libraries are

stored. We can speed up the system by removing the need to do this. This also has security advantages as we can tell the 'prelinker' to load these libraries into the memory at random locations. This is useful because an adversary looking to mess with your memory and inject their own commands will have a harder time finding places to do so. So, let's start by running this command

sudo apt-get -y install prelink (you'll need community repositories enabled to do this)

Now type the command 'sudo gedit /etc/default/prelink'. Change PRELINKING=unknown to PRELINKING=yes.

Now type the command 'sudo /etc/cron.daily/prelink' (this might take a while)

This would be a really good time to restart your system.

Systems manage things called 'locales'. Locales define what language to use when talking to the user. Usually you only need one language, but all of those extra languages could be taking up your memory and disk space. Run 'sudo apt-get -y install localepurge' to change the locales you'd like to use. When it asks, read the instructions and follow them. If you're using English, you should keep EN, EN_US, and EN_US_UTF8. I strongly suggest you restart after this.

You may also decrease CPU usage by putting your Truecrypt containers on external storage devices. When you load them from the hard drive they actually have to be decrypted twice (the first layer of encryption is the full-disk encryption and the second layer is Truecrypt), wasting additional CPU resources. The downside of this is that it will significantly decrease the speed at which you can transfer data to and from your Truecrypt volume, which will likely slow down your virtual machine significantly. As an added benefit, you can quickly hide or destroy the data in the event that you need to whereas breaking into your computer to remove the hard drive and then destroying it would take significantly longer.

You can also speed up your machine by using alternative applications to the ones installed by default. You can switch gedit (your text editor) for mousepad and nautilus (file manager) for thunar just for an example. If you installed Ubuntu (not Xubuntu), you're using a lot more memory than you need. Even Xubuntu can be a little heavy compared to a system with a more lightweight window manager like Fluxbox. Since this is a server, you don't need all the pretty functionality that Ubuntu or Xubuntu provide by default. To install Fluxbox (with community repositories enabled) type 'sudo aptitude install fluxbox thunar'. Then log out and before you log back in again, click on 'sessions' and choose Fluxbox. To access the menu, right click anywhere on the desktop. To start up the terminal, go to applications>terminal emulators>xterm. To start up a file manager (to move files, copy them, look around, etc.) type thunar at the command line. As an added bonus, Fluxbox comes with all sorts of cool themes that barely take any memory. If you are using wireless internet (or even if you're not), it's very helpful to have the network manager enabled. In order to make it work when you start Fluxbox, open up the terminal type "gedit ./fluxbox/startup". Look for the line 'exec fluxbox' and in the line above it add 'sudo nm-applet&'. You also should also add the line 'update-manager&' so that you get updates.

After these changes, my CPU usage is the same but my memory usage has dropped down to 76MB, which will be very useful when running the virtual machine

Install Your Virtual Machine

Now it's finally time to install your virtual machine! Restart your computer, log in as the torify user, and mount your Truecrypt volume. Run the following commands

sudo tor -f /media/truecrypt1/torrc

Next we have to make a hard drive for Qemu to use. You'll want to open a terminal and change into your Truecrypt volume's directory. You can change 5M to whatever you want as this will be the size of your hard drive (10M, 100G, 30G, etc.). This should be big enough to hold everything you're going to put on your server and should probably be a minimum of 5G. File.disk is the name of the file to store the disk in. You can change file.disk as well but make sure you remember that you changed it! Here's the sample command:

qemu-img create -f raw file.disk 5G

Now that we've made our hard drive, let's boot up the virtual machine! The hda flag tells qemu where our hard drive is. The m flag says how much memory to allocate. You can get a feel of how much you have left by running gnome-system-monitor from the terminal. The server can run on 128M but bigger is better unless you run out of it on your host system. The name can be anything you want. I added -no-acpi because I was getting errors about it but you may not. My system also gives a segfault error on shutdown. From what I can tell, this is only in the part where it actually turns off the power so it shouldn't be a problem or a security risk. The redir flag tells qemu to forward traffic to localhost (host) on port 5022 (TCP) to the guest os on port 80 (TCP). Here's my sample command, which you would need to run in the directory where you downloaded Ubuntu Server Edition. Remember: Most times before you boot up the virtual machine you'll need to run the command “xhost +torify”.

qemu -hda /media/truecrypt1/file.disk -m 127M -name TorMachine -no-acpi -redir tcp:5022::80
-cdrom ubuntu-9.04-server-i386.iso -boot d

You should install it pretty much the same way you installed your host system. You shouldn't use disk encryption (as it's already encrypted twice). When it asks, you want a LAMP server which stands for Linux-Apache-MySQL-PHP.

After you install the server, you can run it using this command:
qemu -hda /media/truecrypt1/file.disk -m 128M -name TorMachine -no-acpi -redir tcp:5022::80

For when you're working in the virtual machine, you can access your host machine at 10.0.2.2. This means that privoxy (your web proxy) is at 10.0.2.2:8118 and Tor is at 10.0.2.2:9050

If you want any of your programs on the virtual machine (such as the auto-updater, wget, etc.) to work, you'll have to run this command:

export http_proxy=http://10.0.2.2:8118

This setting will stay this way until you restart your system so if you want it to stick around, you'll have to edit your bash configuration file by running the command nano ~/.bashrc. If there is already a line that says http_proxy, modify it like we did in the previous command. If there isn't, you'll need to add it. Once you're done editing the file, hit Control and W.

Next we'll need to update your server. Run the command:
sudo aptitude update

If you get 404 errors or network connection errors, something has probably messed up with the http proxy variable. It could also be Tor not working well so try updating a few more times combined with restarting Tor. If nothing went wrong, run the command sudo aptitude safe-upgrade and wait for it to finish. Once it's done, you'll want to restart and depending on how many updates are available you may have to do this several times. To turn off your server (and then turn back on to restart) type “sudo halt” in the virtual machine.

The final thing we'll want to do is be extra sure that we're sending our traffic through Tor. Type the command “w3m http://torstatus.kgprog.com/” and you should be greeted with a page that says you're sucessfully using Tor.

Running Your Virtual Machine

Now that you've installed your virtual machine, it's time to start working on it! If you followed the guide verbatim, here are the instructions to get your virtual machine up and running

1. Log in as the torify user
2. Open Truecrypt, mount your hidden service in /media/truecrypt1
3. Start Tor using the command sudo tor -f /media/truecrypt1/torrc
4. Run the command qemu -hda /media/truecrypt1/testing.img -m 128 -name TorMachine -no-acpi -redir tcp:5022::80
5. Log into your server
6. Lock your screen!

Locking Down Your Server

There's a reason that many hidden services don't use PHP or active scripting: it's a huge security risk. Because your server is running in a virtual machine, you'll be much safer but give any adversary command-line access for long enough and they might just be able to break out of your sandbox.

A popular saying in the IT and computer security field is that 'obscurity is not security'. This basically means that you can't assume that a system is secure just because your adversary doesn't know about it. The phone companies learned this lesson the hard way. When we're dealing with a hidden service, obscurity can provide some security. If your adversary can't determine which server software your machine is running, breaking into it will take a little longer.

The first thing we need to do is change some of Apache's settings. Apache is the web server we'll be using. Most of the files you'll need to edit for Apache are in /etc/apache2. It would probably be beneficial to go through all of the files in there and tweak the security settings. Here's a few tweaks I thought would be useful.

In conf.d/security, change SecurityTokens to “Prod” so that your server gives out less information about itself. Also, change ServerSignature to Off so that Apache doesn't tell people it's running. If you change TraceEnable to Off, you'll get less wordy error messages. This is good because we don't want

attackers getting a view of your machine's configuration.

We'll also want to disable the “server status” module. Unfortunately, Tor's traffic will appear to come from localhost (the local machine) and we therefore don't want to allow localhost to see anything sensitive. In mods-enabled/status.conf comment the lines that begin with Allow.

If you're going to be hosting lots of files in a single directory (or a few), you'll probably want to enable indexing for those directories. Indexing is nice because it displays all the files in a directory. As a fail-safe, it's best to disable indexing everywhere and then manually enable it where you want or else you might accidentally enable indexes for folders you didn't intend to. There's a good guide for how to do this at http://www.ducea.com/2006/06/26/apache-tips-tricks-disable-directory-indexes/ that's pretty easy to understand.

While your server will never know its real IP address or hostname, revealing it's virtual IP or hostname tells an attacker that you're in a virtual machine. If you want to change this, you can do so in the envvars file. I suggest adding this:
export SERVER_ADDR=127.0.0.1
export SERVER_NAME=localhost

I also suggest changing some of PHPs default settings (in /etc/php5/apache2/php.ini). You can spend all day hacking PHP to be more secure, so I suggest looking up guides on securing or “hardening” PHP. Here are some important ones:

#don't tell people PHP is running
php.ini expose_php = Off
#if an error happens, don't tell the user anything sensitive
display_errors = Off
#make sure errors go to logs
log_errors = On
#set your maximum size for people uploading files. If you're running a service where people can upload files, this one is important. Just use 10M, 50M, etc.
upload_max_filesize = XM

Uploading Files To Your Server

There are a few ways you can get files onto your server. If you're looking to grab files from the web, such as the install package for Drupal, you can just run the command “wget http://www.example.com/file.zip” from the server's command line. If you want to upload your own files, it gets a little trickier.

Most hosting companies allow users to upload their files through FTP or SSH. The best way to upload files to your hidden service is without either of these services as they open new avenues for attackers. The safest way to upload files is to stick them in a disk image. A disk image is an entire disk on a single file which when opened properly will look like a disk (similar to the concept of a zip file). We'll be making these disk images with a program called Brasero. On your host machine, run the command brasero. Click data project and then add the files you want to transfer over. You can add as many as you want since you won't actually be burning this to a cd. Once you're done, go to project > burn and select image file from the drop-down menu you select. Save it anywhere (your truecrypt volume would be good). When you boot up your server next time, add the -cdrom /wherever/you/stored/your/file.iso flag to the qemu command. Once you're booted up, run these commands:

sudo mkdir /mnt/cdrom

sudo mount /dev/cdrom0 /mnt/cdrom

cp /mnt/cdrom/* /where/you/want/to/put/everything

This should copy over all of your files from the iso. Once you've done this, and shut down your virtual machine, you can delete the iso file.

If you want to install a FTP/SSH server, there are several guides that will help you through doing so. There's a good FTP guide at http://www.ubuntugeek.com/settingup-an-ftp-server-on-ubuntu-with-proftpd.html Setting up an SSH server is even easier to set up than a FTP server and requires almost no configuration http://www.cyberciti.biz/faq/ubuntu-linux-openssh-server-installation-and-configuration/. If you want to allow access to these services from your onion url, don't forget to edit your torrc and qemu command to reflect it.

Additional Resources

Free Software

http://www.gnu.org/philosophy/free-sw.html
http://en.wikipedia.org/wiki/Free_software
http://www.fsf.org - Creators of many core linux apps, the GPL license, etc.

Encryption
http://www.truecrypt.org/docs/
http://en.wikipedia.org/wiki/Public_key_encryption
http://www.pidgin.im - A good IM client that can use AIM, ICQ, YIM, Jabber/Google Talk, IRC, SILC, and many others .
http://www.cypherpunks.ca/otr/ - Encrypt Instant Messages with Pidgin
http://www.getthunderbird.com - A mail application that integrates well with PGP (use the enigmail add-on)
Crypto: How Code Rebels Beat the Government by Steven Levy
http://english.ohmynews.com/ArticleView/article_view.asp?no=381337&rel_no=1 NSA AG-Crypto Sting

Privacy/Anonymity

https://www.freenetproject.org - Anonymous distributed data-storing and communications system

https://www.torproject.org - Home of the Tor Project
https://ssl.scroogle.org - Encrypted anonymous Google searches also see cuil.com
http://www.eff.org - Legal defenders of electronic rights, the ACLU of the internet
http://www.epic.org - Electronic Privacy Information Center - a group that defends privacy online and in real life, has been pivotal in many cases
http://www.tor2web.com - For when you just can't install Tor
Network Forensics Evasion: How to exit the matrix - A how to guide about privacy that was taken offline.
http://www.nickyhager.info/ebook-of-secret-power/ (UKUSA Agreement)
https://forum.nonvocalscream.com/ Unofficial Tor Forums

Security

https://www.cacert.org - Free certificate authority
2600 Magazine - 2600.com - A hacking magazine with a lot of useful information. Only DeCSS defendant to fight the MPAA
Secrets and Lies: Digital Security in a Networked World by Bruce Schneier - One of the best security books ever written
Activist Security Guide – activistsecurity.org – A very in-depth guide about security for activists
Security for Activists - http://security.resist.ca/personal/securebooklet.pdf
Security at Resist.ca - http://security.resist.ca/
Security at Riseup - http://help.riseup.net/security/
http://citp.princeton.edu/memory/ - Information about Cold Boot Attacks

Tech Collectives/Resources

https://www.hackbloc.org – Mainly US-based
http://www.riseup.net – Mainly US based
http://www.aktivix.org/ - Mainly UK based
http://www.tao.ca – Mainly Canada Based
http://www.resist.ca - Another Canadian riseup-like collective
http://www.linux.org/groups/ Linux User Groups
http://lug.org.uk/lugs/all UK Linux User Groups

Software For Your Hidden Service:

Wordpress.org – A blogging platform with no regard for privacy but much regard for usability. Could never get this to work in a virtual machine
Drupal.org – A PHP content management system that's very popular and versatile
http://www.acme.com/software/thttpd/ - Extremely lightweight webserver
http://www.mediawiki.org – Wiki software that is used by Wikileaks and Wikipedia
http://moinmo.in/ - Lightweight python-based wiki that is popular in the onion.
https://www.yacy.net - A distributed search engine
http://www.phpmyadmin.net/ A great SQL Database Manager with web interface

Other Links:

https://secure.wikileaks.org - Censored and leaked information from around the world

http://www.cryptome.org - Censored and leaked information from around the world, mainly pertaining to intelligence