Deep Web Things

Interesting and Curious Things from the Deep Web and Beyond

How to Exit the Matrix

Source: The Uncensored Hidden Wiki
Source URL: http://kpvz7kpmcmne52qf.onion/wiki/index.php/Main_Page
Last update: 2015-07-25

"Protect yourself and your rights, online and off."

This document was created from original *wikitext* sources; some internal and external links may be broken and sometimes text formatting may be corrupted.

→ Read online (single page)

How to Exit the Matrix

Privacy and anonymity have been reduced to the point of non-existence in recent years (Thanks Obama). Our personal, private information is stockpiled and sold to the highest bidder like so much inventory at a warehouse. National Security Letters are written to make countless requests for records from our search engines, libraries, and book stores with no court oversight. Email and especially searchable data is practically unprotected from anyone who might ask to have it. All our electronic communications are tapped. Massive governmental data mining schemes are being built to record everything we publish on the web. In many workplaces, employers spy on and control their employees' Internet access, and this practice is widely considered to be acceptable.

These are dark times. The Fourth Amendment has all but disappeared, thanks to the Wars on Drugs, Porn, and Terror. Any practicing trial lawyer will tell you that you can no longer rely on unreasonable search to be the basis for excluding evidence, especially for digital evidence in the hands of a third party. Likewise the First Amendment has been shredded with exceptions and provisos, and is only truly available to those with the money to fight costly (and usually frivolous) court battles against large corporations. In short, you can say

what you want so long as it doesn't affect corporate profits.

How we got to a legal state where all this activity is the accepted norm, I'm not quite sure. It seems to stem from an underlying assumption that our function at work and at home is that of a diligent slave - a single unit of economic output under the direct watch and total control of our superiors at all times; that we should accept this surveillance because we should have nothing to hide from our benevolent overlords who are watching us merely to protect us from evil.

I believe this view is wrong. Moreover, I believe it is time to reverse the tide. This document seeks to provide the means to protect your right to privacy, freedom of speech, and anonymous net access even under the most draconian of conditions - including, but not limited to, both private and criminal investigation (which happens far more often to innocent people than one might like to think). "So what are you saying? That I can dodge bullets?" "No.. What I am trying to tell you is that when you're ready, you won't have to."

Welcome to the first day of the rest of your life.

Document Organization

This document is organized into seven chapters. The first chapter is an introductory philosophical discussion, and the next six are based on the six main ways you can leak information about who you are onto your network connection, or to an attentive individual.

1. The Matrix
   A discussion of what the Matrix is, how it functions, and how to resist and subvert it. This forms the philosophical underpinnings of this HOWTO and the driving force behind the author's motivation to work ceaselessly on this document for over a year, and then proceed to give it away for free. Not required reading, but strongly recommended.
2. Network Attributes of your computer
   This includes your network hardware (MAC) address, your IP address, and your 802.11 nickname. This section describes ways of obfuscating each of these attributes, as well as your network data itself
3. Local Programs and Services
   Various programs you run can leak information about you to the network. This section describes how to turn them off.
4. Web related leakage
   Even after you have taken steps to obfuscate your network attributes, it is still possible to leak a surprisingly large amount of information about who you are through your web browser. It is even possible for websites to determine your original IP after routing through a proxy (or even Tor), if you are not careful.

5. Intrusive Surveillance
   In some environments (public computers, labs, oppressive work places), your computer may be bugged and under direct deliberate surveillance from a third party. This section describes what to look for, and also describes how to use these same tools to your advantage to conceal your activities. It also covers measures you can take to mitigate information disclosure in the case of equipment seizure.
6. Anonymous Communications
   The previous 4 sections have dealt with how to access Internet resources without fear of divulging your identity. But what if you have something to say? This section discusses the ins and outs of publishing data and communicating anonymously.
7. Physical Interaction
   The ultimate goal in anonymity over the Internet is to carry it over into the physical world: to use money, and to be able to buy and sell items and otherwise conduct business without fear of surveillance. The means for doing this exist, yet most are prohibitively expensive for the average individual. In most cases, low cost, "good enough" alternatives are available with some extra effort, however. Hopefully, as the Anonymous Economy continues to grow, tools to aid in interacting with it safely will become profitable commodities themselves.

Where to find this Document

The latest version of this document can be found here or here (Wayback Machine copy). The Anonymity Portal also provides a mirror (Wayback Machine copy), along with several other documents. Those wishing to mirror or build their own copy can download this web tarball. This instance was built with xmlto html ExitTheMatrix.xml.

License

Credits

This document exists because of the hard work of literally millions of individuals working in concert to build a free, open world where all can meet, trade and converse without fear. One day The Man will burn.

At the same time, I would also like to thank The Man, because without him, the millions of individuals working in concert to build a free,

open world where all can meet, trade and converse without fear would not have such a fascinating hobby.

Furthermore, I would like to thank the dozens of contributors who have tipped me off to various news articles, software, FIXME solutions, and so on. Your help is much appreciated!

Feedback and Assistance

If I missed anything you feel is important, or if anything is unclear, please contact me via email at <aceevader]-a-t-[mailvault.com>. Particularly if you have any material to cover any of the FIXMEs found in the text, please email me. If you are someone who needs confidential anonymity advice or assistance, do NOT use my mailvault GPG key, since I have no control over preventing leakage of the passphrase. Instead, use this key. While mailvault is not located inside the USA (and thus not subject to the most likely form of assault: a National Security Letter), it is not outside the question that they could be coerced in some other manner. If you are unfamiliar with GPG, you may consider installing a graphical front end to help you along.

The Matrix

What is the Matrix

In my opinion, The Matrix films provide the best metaphor our society has for understanding why organized evil and oppression are allowed to exist, and so I will use it for this purpose. While my interpretation isn't the only possible one, I believe it to be valid, comprehensive, and most importantly, illustrative of the message I am trying to convey.

So let's begin by discussing what the Matrix is not. The Matrix is not the physical world. As far as I'm concerned, the physical world is actually real and is in fact governed ceaselessly by the laws of physics. Conversely, the Matrix is also not the Internet, despite what many seem to believe. The Matrix spans and transcends both these worlds. It has existed since the dawn of civilization, and it will continue to exist until its collapse.

So then, what is it? Well, that's complicated. Much like in the movie, it's nearly impossible to convey the size and scope of the Matrix to someone who doesn't already see it for what it is. However, unlike the movie, I believe it is an ethical imperative to try to convey it in a literal sense, even to those who are so dependent upon the Matrix that they would fight to protect it. At worst, they won't understand or believe and will continue on about their business. In a sense, I believe Cypher was right to resent Morpheus for what he did, because Morpheus engaged in flat out trickery and deception to free people.

But I digress. The Matrix is the social structure that subordinates Humanity to its will. It is the machinery of society that exists solely to perpetuate itself, its influence, and its power independent of any human need. It insulates us from each other and ourselves through deception, and essentially transforms us into servile engines of economic and political output (power). The machines that live off this power are institutions: large corporations, governments, schools, religious institutions, and even non-profit orgs. Every institution will reach a point in its existence where its primary function becomes self-preservation and perpetuation, instead of serving human need. At this point it becomes a machine of the Matrix. For example, when they become machines, governments cease to serve people and instead seek to extend their power over them; corporations prioritize increasing shareholder value over producing quality products or otherwise serving the public good; schools view students as a means and not an end; religious organizations equate membership with salvation (and actively oppose other teachings and even independent practice); and non-profits and charities spend more budget on fund raising activity than on their original focus. Inevitably all large institutions eventually become machines. They become too big for Humanity.

In addition to the independent self-perpetuating machines that write most of our paychecks, the Matrix has several major cooperative and more actively sinister groups of machines subsisting off of its power and directly contributing to the structure of the Matrix itself. These groups are the Military Industrial Complex, the Political Industrial Complex, the Prison Industrial Complex, the Surveillance Industrial Complex, the Media Industrial Complex, the Academic Industrial Complex, the Agricultural Industrial Complex, the Medical Industrial Complex and the major religious organizations (not to be confused with actual religions, many religious organizatons have abandoned the underlying principles of the religions they claim to represent). All machines in these groups either actively oppress humanity, or enable the oppression to persist. It is through their combined efforts that the Matrix takes on some of its more distasteful qualities.

Resisting the Matrix

Resistance is a mental state. The Matrix is designed to make it easy to accept what it tells you, and to make it hard to filter the Truth from the lies. Resisting the Matrix requires understanding its operating principles and assumptions, rejecting them, and helping others to do the same.

The Matrix is fascist, the Matrix is deceptive, and the Matrix is bureaucracy. The Matrix is essentially the rule of the institution over the individual, and in it, the rights of the individual are subordinate to the rights of the institution. Individuals have to believe (or at least not actively oppose the idea) that large corporations have the right to protect their profits above all else, and thus dictate policy and law. They have to believe that this law is just, moral, and seemingly based upon reason. Or, they have to feel unaffected by the law on an individual level. They have to accept the program, and be satisfied with the rewards given for doing so. They have to do their jobs, pay their taxes, and be content with their salary (at least to the point where their salary and the stability it provides are appealing enough to deter risking leaving the Matrix). Rejecting these beliefs is the first step in resisting the Matrix.

Furthermore, people must be insulated from the creative process. They have to forget that they are able to produce craft as individuals independent of large institutions, and they must feel entirely dependent upon the system to provide them with what they need. It is mostly through the violation of this principle that many who work with computers come to free themselves, or at least come to see the Matrix for what it is. Despite being products of the Matrix (for the most part), computers and the Internet enable humans to create individual works on a global scale: independent media, self-publishing, Free Open Source Software, computer music, computer art and graphics, and so on. Computers also enable independent people to communicate and build human-serving social structures outside of the Matrix.

However, note that computers aren't the only means of accomplishing this, and this time period isn't the first one of Exodus. In the 1960s, for example, people departed from the Matrix en-mass and independently created art, culture, and music, largely catalyzed by psychedelic drugs. Unfortunately, much of this structure collapsed due to a number of reasons, the main one being the hasty, ill-considered and unsustainable manner of its construction and the subsequent institutional and legal backlash. Miraculously, however, many of the core ideas have persisted, and their proliferation is largely the reason I am Aware and able to write this document today. It would seem that the present catalyst is a combination of the Internet and again psychedelics. Both of these phenomena provide a way of disconnecting yourself from the programmed reality and assumptions of the Matrix and taking your perceptions into your own hands. However, your "perception" is nothing but your individual dream that you have created as you have gone through life. There is the dream of the society that has been passed down generation after generation and instilled into your mind by your parents, friends, schools, and institutions. And then there is your individual dream. Each step of the way in your life you have lived subjectively, and depending upon how and where you grew up, who you hung around with, and the habits you formed, you created your little dream. Your "theory" on life while you stay completely unconscious of this. Everything that you think is "you", "I", "me", and everything you believe that you identify with is simply not you. It is not the truth. It is part of the giant web of individual dreams that everyone is in the clutches of on our planet. This is one of the main reasons why this world is the way it is, why it is so chaotic. You will scorn anyone who does not dream what you dream of, and someone will do the same to you. It is impossible for us to live the same individual dream because we cannot know everything about each other down to our core. It is a constant fog that grows bigger and bigger and more

dense as each day passes. It is your ego and it is my ego. The dream is not real. If we want to even begin to understand what it means to exist as a human being on this planet and evolve, this truth must be learned and it is just the beginning. Somewhere along the way, the entities that have been running the show figured out how the ego works, and they have been doing a damn good job at distracting us from trying to find a way out of our dreams and the dream of society. Psychedelics are simply a tool, but one of many to truly explore and expand your consciousness. However, whether it be psychedelic substances or meditation, it is not the ultimate answer. It is simply showing you the door, but it is your choice whether or not you want to enter into the other side.

To persist, the Matrix requires control, and in democratic societies it maintains this control by filtering people's view of reality through corporate-owned mass media and television. In essence, the Matrix requires a form of thought control, but not in the science fiction sense. Instead, it achieves an effective enough manner of thought control by manufacturing consent. The large majority of the public has to "buy in". They have to believe that the news media give them an accurate picture of the world. And by and large, they do believe this. Everything the general public knows about the world, they know through the Matrix. The symbols and images the Matrix presents to them have become more real than reality itself. Hence the popularity of the ungodly abomination that is Reality TV.

Note that while some media outlets do actively promote a political agenda of domination and control, on the whole it is not through some grand conspiracy that this process (or any process of the Matrix) functions. It is simply the way mass media is organized. Mass media is a machine that exists as a profit maximizing entity, and the most profitable news (and the cheapest news to produce) is recycled soundbytes and pre-packaged press releases from corporations and government. Furthermore, in the interest of preserving its revenue stream, news media cannot allow the public to hold any opinion that may threaten the authority and policy of government or the profitability of their sponsors, which are also machines of the Matrix and almost always directly involved in the business of domination and control. Thus the media must perpetuate the status quo. No news is good news.

Understanding this bias in the media is key to undoing the filter it applies. Consider who the advertisers and sponsors are. Beware of press releases disguised as investigative reporting. When possible, confirm mainstream, corporate produced stories with coverage from places like IndyMedia (go local), Wikinews, GNN, Politech, Free Speech TV, Democracy Now, Free Speech Radio News, and FAIR. A lot of the time these sources also cover many eye-popping items that for some reason don't even receive mention on corporate news media.

Last, and most assuredly not least, the Matrix seeks to identify and know its members at all times, in a misguided attempt to maintain control. It demands total surrender of your privacy to function in it. It is by breaking this last property of the Matrix that we come to truly free ourselves from it; to create economies, communication, and culture independent of its control.

Of course, the ultimate form of resistance is to fully disconnect from any and all dependence upon and allegiance to government and

institution; to remove yourself from the power structure of the Matrix, and contribute your economic output to resistance economies. It is this form of resistance that faces the most violent opposition from the Matrix, since providing this economic power is the primary function of Humanity, as it sees it.

Unfortunately for many this form of resistance is simply unattainable due to family and social ties, especially starting from your first realization of the size and scope of the Matrix. However, unlike in movie, it is possible to liberate yourself gradually instead of immediately, and in some cases this can prove easier than an 'all-at-once' attempt. It starts with disconnecting. Cut out TV from your life entirely, especially TV news and Reality TV shows. You should be able to get all your information and entertainment from the web, or from real reality (or from the occasional movie). Avoid chain stores where possible, especially for food. Supporting smaller (especially sustainable) business keeps entrepreneurial and independent business spirit alive. Getting and staying out of debt (especially debt without equity, or rapidly depreciating equity such as car loans) is crucial, as debt is a primary mechanism the Matrix uses to ensure your obedience. Also, if you are a salaried employee, working a 40 hour (or perhaps even 35) hour work week can be a big start to declaring your freedom from the machine and the corporate American peer-pressure to be a diligent slave. It also frees up huge amounts of mental energy which is then available for resistance.

From here, a limited form of resistance whereby you leave the Matrix for short periods of time (long enough to conduct purchases, business transactions, and communications with the underground) is well within the reach of all computer literate individuals, and functioning as a consumer is sufficiently supportive of the Anonymous Economy for it to be sustainable. Moreover, the probability of discovery of this sort of activity can be reduced as much as you choose. Doing this effectively is the subject of this HOWTO.

As you progress, you will notice yourself developing one or more separate identities, or pseudonyms. It is best to build as much insulation between these nyms as possible. They shouldn't appear to know each other, shouldn't really talk about the same stuff or buy the same things, and above all should be diligently separated from your original physical identity. Maintain different wallets, bags, user accounts and possibly even computers. In short, develop one or more Tyler Durdens, except without all the insanity, self-destruction, and sociopathic behavior. Or with it, if it helps.

The adept and the entrepreneurial will find it an easy step from here to total freedom. The next stage is to go into business for yourself. It doesn't have to be an anonymous business, but those who manage such an achievement do enjoy the satisfaction that they are directly subverting the Matrix and helping to weaken its hold on everyone.

Subverting the Matrix

While resisting the Matrix is an act of mental rebellion, subverting the Matrix is an act of social revolution. It requires understanding the types of human communities that exist outside of the control of the Matrix. It also requires understanding what sustains them, and if and how they directly or indirectly weaken the structure and control of the Matrix. Once you understand this, it is possible to intelligently align yourself with communities that actively weaken the control of the Matrix.

Gift Culture

Gift Culture (also known as Free Culture, or the Gift Economy) is a social structure where your status is determined by how much you are able to give away. It is not mutually exclusive to any other economic system, and examples of gift economies exist on top of capitalist, communist and socialist economies.

Gift culture has brought forth some of the most astounding recent achievements of the human race, including the scientific research community, much of the World Wide Web, the entire Open Source movement, vulnerability and security research, and Wikipedia, just to name a few examples. Gift economies tend to function best in the digital world, where something can be given without reducing the inventory of the giver.

However, the Burningman project is a massive experiment in bringing Gift Culture back into the physical world, and quite successful at that. Well over 35,000 people populate Black Rock City in the middle of the desert every year to give as much as they can to each other. The event serves in part as a model for the time when energy becomes abundant and human beings are capable of interstellar space travel. Obviously the burning of The Man is the climax of the event.

This is no small coincidence either. Gift culture does subvert the primary mechanisms of the Matrix. The Matrix subsists by transforming human endeavor into economic output which it uses to maintain its control. Gift culture, on the other hand, releases human endeavor for the good of all who would receive it. When items are given instead of sold, the power and control obtained through ownership is eliminated. Furthermore, in the case of Open Source Software, the fact that full freedom over the source code is also given means that code that the Matrix would never willingly create is readily available for the purposes of this HOWTO.

It is interesting to note that even machines of the Matrix are motivated to participate in gift culture - especially in the Open Source movement. It benefits many corporations as well as governments to have a common reference platform upon which they can build their individual products and infrastructures. Their cooperation in building this common platform vastly reduces the cost they would have paid to develop their own platform in-house, and is also inevitably cheaper than paying a single entity to do the same. The combined experience and widely distributed expertise, as well as the flexibility of modifying the common platform to perform a wide variety of tasks, yields a better system for all, and cheaper. In the digital world where copies are free, capitalism compels Gift Culture.

Unfortunately, some companies, such as Amazon.com, reap tremendous benefits off of Open Source Software, yet have a company policy of zero contribution back to the community. Other symptoms of this problem include Microsoft's war on the security research community, and the tendency of (even State funded) University Professors to refuse to provide Open Source reference implementations of their work. There are mechanisms discussed in this HOWTO that enable this trend to be reversed, which leads us nicely into the next cultural segment.

Information Anarchy

A closely related social structure to Gift Culture is Information Anarchy. The idea behind Information Anarchy is that all information should be as widely and freely disseminated as possible. The cultural ethos is vehemently at odds with Intellectual Property, and refuses to recognize any such law (or suffer any code) that abridges free exchange of information.

Needless to say, the machines of the Matrix don't take too fond a view on this ideology. Unlike gift culture, which is an indirect subversion of the mechanisms of the Matrix, Information Anarchy directly challenges the Matrix's perceived right of ownership of human ideas. The past decade has seen an unprecedented decline in the freedom of information due to some of the more rabid elements of the Matrix. The machines of the Matrix now draw tremendous power from ideas and digital content/information. Recent examples include the DMCA, extension of copyright duration, the harsh criminalization of copyright infractions, and the resulting side-effects which lead to the criminalization of certain forms of technology. The legal climate for free speech and innovation has never looked darker.

However, hope is not lost. The future looks so dark precisely for the reason that Information Anarchy poses such a grave threat to a major power source for so many parasitical machines. On some level, the Matrix knows its hold is tenuous. At every opportunity, the Matrix will tell you that protecting Intellectual Property encourages creativity. It has even developed an amusing array of propaganda to promote this idea, even going so far as to begin the brainwashing at an early age. (Yes, the National Counterintelligence Executive is in fact a real office of the US government, apparently one of its major propaganda arms. Their stuff is hilarious. I recommend printing some out at your local copy shop before it becomes classic.)

All of this nonsense is observably false. Societies have always been most successful when communication and ideas were open to all. It is important to remember that the world didn't always operate this way, it was only when the ruling elite of the Matrix realized that ideas and creative expression are easily converted to economic power that they took claim over them. Economic systems can and will adapt to a form that is more profitable for human creators instead of their machine owners.

Five chapters of this HOWTO are devoted to protecting your digital identity and are easily applicable to contributing to the goal of

Information Anarchy and providing even more economic incentive to move towards alternate revenue models and/or Gift Culture. In every opportunity possible, do not support the system of Intellectual Property that the Matrix has created. Naturally as its power wanes, it will become weaker and less relevant, as content creators seek their pay through other means. The cancer starves, and dies.

The Anonymous Internet Economy

So as of late, a major source of the erosion of civil liberties stems from the fact that casual economic transactions are becoming increasingly difficult to conduct without permanent, identifiable information being associated with them. With the advent and increase in the volume of Internet commerce, casual purchases of personal items, books, software, and even medication are now irrevocably tied to your own personal identity. Bookstores such as Amazon now build complete dossiers of sorts on their customers reading habits, and much of this information is available publicly.

As a result, the natural reaction to these circumstances is to find methods to make Internet commerce behave more like physical commerce, where you have the option of anonymity by using cash or cash-backed identity free payment methods. An Anonymous Internet Economy.

The Matrix is providing massive economic incentive to create this economy as well. It has recently been revealed that the FBI writes over *30,000 "National Security Letters" in the US each year*. Consider how easy it would be for them to demand records of everyone at Amazon who might like to buy a particular book, or who has ordered "indecent" materials from websites? Amazon already does classification of consumer's interests for marketing purposes. Their engine can perform this classification instantly. What would they have to say about what books you like to read? How about Google, and the types of adword sites you are typically presented on the search website/via gmail? Google and many other search engines maintain indefinite logs of who searches for what keywords, along with lots of other data. These are prime targets for National Security Letters or just general government subpoena.

I provide the basics for conducting anonymous transactions cheaply in FIXME this section. You can use these techniques to get yourself started and comfortable with interacting with the Matrix anonymously. From there, the entrepreneurs in the audience may wish to start a business to start making some money in this new economy, and thus begin to fully escape from the control of the Matrix.

Markets of interest might include items in online games, anonymous web hosting, certain types of medicine, or even illegal electronics. For example, many people are too lazy to build a MythTV box, but personally I sure as hell would buy one over a crippled and ad-infested TiVo subscription service any day. If I watched TV, that is.

As you can see, most of these things go on above ground today, but for how long? And why at such high risk for consumers living in

less accepting legal climates? What about those who would pay more for more protection? For example, some customers may be attracted simply to the ability to free themselves from marketing and government profiling. Those who purchase certain types of books might prefer if Amazon and whoever else didn't have this information tied to their physical identity.

Yet another possible white/grey market to tap might be a physical anonymous remailer service for people who would like to conceal their street address from someone mailing them something in order to avoid becoming listed in a database for marketing spam and/or to avoid general profiling and surveillance, or to be able to order a product that won't normally ship to their geographic location. Basically the way the system could work is through a website where you create a temporary account number or unique pseudonym. The package is then shipped to a relay point where the account number/pseudonym is read off, and a new label printed onto the package. It is then mailed to its new destination, and any electronic and paper records are destroyed. It also has the advantage that extremely paranoid users can potentially chain multiple locations together for extra security, so that competition does not necessarily compete for market share, but instead cooperates for it. You might consider marketing this as a "Virtual Office Solution" to avoid liability, if done above ground. A useful technique for verifying that packages have not been opened/examined en-route is to create a unique multicolored wax seal swirl using two or more candles, photograph the seal, and transmit the photograph electronically via encrypted email. Delivery/payment can be ensured using normal UPS/Fedex/USPS tracking numbers, which can be encrypted to the senders public key and then destroyed.

The demand for such a system might not be immediately visible now, but once the next Patriot Act or similar legislation removes all anonymity from the mail, the demand should skyrocket. This business has the advantage that it is extremely low setup overhead and is very easy to start small with low capital, just to test market demand. Once the business is proved worthwhile, FedEx and possibly other major carriers offer bulk shipping rate accounts to merchants that could be taken advantage of, bringing the overhead work and cost to your customers potentially very low.

Taking this idea a step further yields a "ghost walker" contract market, much like the ones described in FIXMEToward A Private Digital Economy. Most of the P2P token-based nonsense there can be ignored, but his key idea could be transfered to a ebay-like auction site. Basically the idea is that people would contract the services of someone who is skilled at staying off the radar to conduct transactions that for various reasons they do not want linked to their identity (again, buying books, vitamins, medicine, regionally available items, web hosting, illegal electronics, and so on). Sort of like the inverse of a Private Investigator, these people would do anything from purchasing items, mailing and delivering packages, donating to charities, acting as couriers, business agents/representatives, mail forwarders, and so on. This can already be carried out in a guerrilla fashion on community/local city classified ad servers such as the nearly universal Craigslist (where it is possible to contract people from different state and country jurisdictions quite easily). In the ideal situation, a dedicated website would be created. Each "ghost walker" would have a nym (possibly paying a fee to do so, both to support

the site and to discourage morphing), complete with ratings and reviews, prices per task/risk factor, and so on. Contracts would be posted by clients containing a generic description of the task, and interested ghost walkers would contact the buyer with prices. The buyer would then select a particular ghost walker to reveal the complete details of the contract to, and terms of payment. Given the tendency to increased total surveillance, lots of regular people may be interested in using this service.

So there are numerous markets that can be potentially very lucrative while at the same time helping to build a social structure independent of domination and control. In general, any mechanism of state control creates markets for equipment or components that can be used to circumvent this control. Keep your eyes open for opportunities. If you have any suggestions or ideas, please feel free to contact me so that I can update the HOWTO for all to benefit as we work together to free ourselves.

Freedom Seekers are Not Terrorists

Essentially this HOWTO represents the hacker community truly claiming independence for itself from national and institutional rule. While many the ideas and techniques present in this document can be found elsewhere, I believe this is the first time such a coherent, consistent, and focused collection of these ideas has been assembled for a single purpose. 17 years ago, we made our Declaration of Independence. This body of code-law represents a manner of Constitution of Cyberspace. A basic set of rights we claim for ourselves through our use of technology.

Because the accusation will inevitably be made, I would just like to emphasize the distinction between freedom seekers and terrorists. The type of resistance and revolution discussed here is non-violent, and singularly focused upon allowing an individual to have real freedom through privacy and anonymity. While reactionary individuals might argue that the knowledge presented in this HOWTO could aid actual terrorists, the reality is they've already had much better training which has proved to be quite effective in practice. Furthermore, we're talking about people who are willing to give their lives in their quest: people whose families, friends, and their own lives have already been destroyed by Empire. These people will stop at nothing. They don't need this HOWTO to create new identities for privacy purposes when perfectly valid ones can be stolen readily. Nor do they care much about surveillance, since surveillance can't stop a suicide mission. What's worse, is that on some level the police state must know this. Even being generous with the reasonable doubt, all evidence seems to indicate that at best it simply used the tragic events of Sept 11 as an excuse for a long awaited massive power grab, with the resulting legislation doing far more to target the average citizen than any particular terrorist network.

I recently had a discussion on Usenet where it was asserted that taking action to protect yourself and withdraw from the Matrix might generate even more excuses for introducing oppressive legislation and policy. I believe this will not be the case, and that moreover this thinking is in fact defeatist and even dangerous for a few reasons.

First, the Matrix will not and can not overtly fight this behavior, as any public attention given to it will only provide it with more energy and momentum. The Matrix media filter won't even allow the individual pieces of information that lead to the conclusions of this introduction to be discussed for this very reason. There is no way it would willingly publicize this ideology in its entirety, even to attempt attack it. Furthermore, there is no need to. The Matrix already has more than enough material to drive through as much oppressively restrictive legislation as it likes in the name of fighting kiddie porn, the War on Drugs, the War on Terror, and in the name of protecting corporate profits. As stated above, it is already taking full advantage of this fact, to Orwellian ends. The interesting phenomenon is that the more ridiculous the regulations become, the more commonplace it will be for the public to want to circumvent them, which only serves to strengthen resistance economies.

Second, if you look at all of history, freedom has never been given to a populace. Left to its own devices, the state only ever grows more powerful. It never surrenders power over citizens freely. Take any instance in history where people have established rights for themselves, and you will see it was the result of a long, drawn out battle that the state simply lacked the resources to continue to fight effectively. Prohibition wasn't abolished because people acted good, honest and sober, it was because they got falling down drunk and alcohol consumption soared to new heights while the Puritan state was powerless to continue to oppress the newly criminalized middle class. Likewise, Civil rights weren't won because Africans obediently stayed in their designated rolls, complicitly accepting "separate but equal" facilities and politely tolerating discrimination. It was because they practiced civil disobedience and active resistance against injustice.

I believe in civil disobedience, and more importantly, the clear distinction between morality and law. I believe that it is defensive, defeatist thinking to say that "if we just be good, they will reward us and repeal laws." The laws - the DMCA, the Patriot Act, the REAL ID act, recent supreme court decisions and the now entirely conservative dominated court - are already essentially fascist and will only continue getting "tougher" on "crime" and "terror", stripping away the rights of citizens in the name of "safety" with little real gain except the hoisting of that Floating Eye on the dollar bill ever higher above the base. The total surveillance state has been a goal of the current cabal for time out of mind.

Third and finally, once again I'm not advocating violence here, or even any sort of crime that has a victim. I'm advocating creating a social and economic structure based on anonymity (not necessarily illegality) that drains the corporate state of power and thus weakens its ability to enforce fascist law and practice.

In my view, the only thing that will cause the state to rescind is the realization that much like in the 1920s with prohibition, it has criminalized a vast portion of its population that it is now powerless to control, and furthermore that its fascist law has done nothing to safeguard against the true monsters that its foreign policy has created.

Continuation on its current course of action will lead the Matrix to experience ever increasing instances of identity theft, repeated infiltration of data warehouses, massive underground surveillance rings, and so on. These actions are not advocated in this HOWTO, but they will inevitably become more commonplace as the Matrix continues to make it easier to steal an existing identity than to create a new one or otherwise escape from ceaseless surveillance.

At this point in time, the Matrix faces two options. It can either choose to allow us to be free, and create official sanctioned means for people who wish to free themselves from endless surveillance and total control to operate anonymously within the system, or it can choose to fight war after costly war on civil liberties and basic human rights until enough people are fed up with its behavior that they begin to depart en-mass. Again.

But we already know the choice the Matrix will make. Already I can see the chain reaction of propaganda, the sound-byte media precursors that trigger the onset of an emotion, designed specifically to overwhelm logic and reason. An emotion that the Matrix will use to blind the masses from the simple and obvious truth: We are going to be free, and there is nothing it can do to stop us.

Spreading the Word

Initially I had planned for a small distribution of this document to only a select few, to attempt to "stay under the radar". But as I indicate above, I now realize that is a flawed approach. Much like in a mixed network, the more people working to protect themselves and their identities, the better off the end result is. The stronger the support economies grow, the better able the resistance is to function autonomously.

An IRC friend of mine has designed some FIXME slick business cards that can be distributed at functions, protests or wherever. Online print shops will typically print 200-500 of these for around $20. If you're itching to give me some kind of donation for some reason, you can direct your funds towards that instead. It even works on a sliding scale. For example, you can buy some sticker paper for like $5 at your local office supply store and print out some stickers to put up at coffee shops, clubs, bars, Internet cafes, bookstores or anywhere intelligent people might be able to jot down or quickly visit a URL. If anyone wants to create another design, send it to me and I will post it here also.

Target Audience

This document is written at a technical level appropriate for "power users" - people who like to tinker with their computer configurations to get the most out of their experience. Novice computer users who are uncomfortable tweaking settings, editing configuration files, and

occasionally using the command line probably will struggle with much of the material regardless of OS, unfortunately (though at least one person has offered to help elaborate the more technically involved sections to help novices along - we'll see how that pans out).

I try to be as operating system agnostic as possible, providing information for Windows, Mac OS, and Linux, but due to the open and readily customizable nature of the system, the Linux material probably will be the most well developed.

As far as demographics, I expect this document to be useful to a wide variety of people from several walks of life. In particular, some of the major categories are:

Civil Libertarians

Those who are concerned with the gradual erosion of their personal freedoms will probably find nearly the entirety of this document useful and interesting, since it is intended to provide techniques and countermeasures to restore nearly every right that has been lost due to the Wars on Drugs, Porn, and Terror. It is now possible for the anyone (the US government, frivolous civil litigants, P.I.'s, and so on) to enumerate just about everything in your home without many warrants, simply by subpoenas or a National Security Letters. Somehow I doubt this situation was exactly what the Framers of the Constitution had in mind... This document can help you to keep as much of your personal belongings and reading habits actually private and out of numerous commercial databases (which are readily availably to law enforcement).

Whistleblowers

Whistleblowers who are interested in exposing wrongdoing, corruption, cover-up, or conspiracy at their workplaces will find this document useful for protecting their identities while contacting the press, or otherwise disseminating evidence over the Internet. When you are jeopardizing your job (and possibly your life) to expose wrongdoing, you must assume that NO institution will be able to protect your identity from someone who is determined to silence you. Your only option is to make sure no one knows who you are in the first place until such time as your safety from retaliation can be guaranteed. If followed carefully and diligently, this document will show you how to accomplish this.

Bloggers and Independent Journalists

Similar to whistleblowers, bloggers and forum posters often find themselves the target of harassment, especially when reporting on controversial material, or even when people who comment on their pages choose to do so. While it has recently been ruled that bloggers are afforded the same rights and protections as journalists, in cases where the blogger is exposing corruption or negligence at

their workplace, additional measures of protecting oneself may be desired. In some cases, it may be desirable to publish pseudonymously simply to avoid the stress (and expense!) of having to deal with frivolous lawsuits such as these.

"Political Dissidents" and Inquisitive Minds

Forget about China, even in the USA it's no secret that the FBI has consistently harassed those who have dared to speak out against the status quo. Targeted groups include vegans, Catholics, Quakers, peace activists, environmental volunteers, 3rd party candidates and campaign workers, independent journalists and bloggers, and even members of the mainstream press. What should be most disturbing to the average citizen is how easily it is to become mistaken for one of these "trouble makers" simply for buying (or being recommended) the wrong book on Amazon, posting on the wrong blog, buying certain types of food on a credit card, donating to certain charities, etc. With the advent of aggressive data mining and aggregation, it is all too easy to be lumped in with "the wrong crowd". If followed properly, this document will help you to retain your freedom to investigate alternative views and information without leaving an electronic trail of this activity to open yourself up to harassment. It should also help you minimize the damage a determined FBI agent (or vindictive ex-spouse or other enemy who has hired a PI) is able to do to you.

People with Enemies

As hinted at above, it's not just dissidents who need to be concerned either. Surveillance and draconian law can be used as dangerous weapons. All it takes is for one motivated enemy to hire a PI with access to the major data warehouses to dig into your life, find something that they can use, and then phone in to report you, or simply blackmail you. Anyone with a vindictive ex-spouse, political adversary, or even a feuding neighbor can be the target of this abuse.

Programmers and Security Researchers

Due to the DMCA and insanely broken patent law, programmers have found their freedom of speech horrendously restricted in the USA. Many security researchers have been afraid to discuss the privacy implications of copyright protection technology that is essentially spyware. Others are afraid to publish vulnerability assessments of cryptographic systems that may be tangentially related to copyright infringement, or even simply software in general. This document can teach such programmers and researchers how to conceal their identities and thus assist peers who are operating in legal climates that still respect the freedoms of speech and innovation.

The Video Game Underground

The DMCA has also been used to harass video game hackers and cheaters. Some gaming companies will ban you from their online services if they discover your involvement or subscription to "cheat forums". Most will not hesitate to issue bogus DMCA takedown notices to cheat/mod websites that operate in the USA. Software that modifies online games to provide additional features, cheats, or automation is also the target of DMCA harassment. Furthermore, programmers who publicly reverse engineer and re-implement open source game servers find themselves the target of lawsuits. This document should assist these people to continue to play and modify games how they see fit without fear of persecution.

Moonlighters, Double-Shifters, and Consultants

In a similar vein, those who are working multiple jobs may wish to conceal this fact from their employers due to fear of retaliation. I expect the most typical use case of this document will be programmers working as consultants, or who wish to contribute to Open Source projects in their free time.

Potential Victims of Identity Theft (Everyone)

This document can also help those who are interested in protecting their identities and/or financial information from being stolen due to their commerce online and elsewhere. As mentioned above, data warehouses such as Choicepoint are essentially making identity a commodity. It is inevitable that this data will be leaked and stolen again and again. With identity becoming an increasingly integral aspect of functioning in society, black markets that sell this data will continue to be extremely profitable. Just like the War on Drugs, the War on Terror fought through the politics of domination and control will lead to ever escalating levels of waste, destruction, and chaos. The best way to protect yourself is to minimize your digital footprint: Use anonymous forms of payment online, and conceal your name and mailing address.

Entrepreneurs

The last major category of people who are likely to find this document useful are those who are interested in providing privacy and anonymity services and software to others. Privacy and anonymity are difficult problems. There are many holes to be filled in, usability issues to be addressed, and markets to be built. To this group of people, every privacy problem and legal restriction should represent a potential market to get involved in. However, DO NOT SELL SNAKE OIL. If you cannot stand up to legal or other pressure, you need to inform your users of this fact clearly, so they are sure to take appropriate precautions while using your service (especially if you are located within the USA). Very few, if any, privacy services are capable of operating as stand-alone one-shot solutions. What is needed is a series of tools and components that can be combined arbitrarily. Focus on one component, and do it well.

Network Attributes of your computer

MAC Address

Every 802.x network card (wireless, ethernet, token ring) has a unique 48 bit identifier known as a MAC address. This address is burned into the EEPROM on the card, and oftentimes is used by networking equipment to track users as they come and go, frequently associating MAC address to hotel or dorm room #, credit card number, login info, etc. This means hopping on a network that has authenticated your hardware before and expecting to be anonymous this time around is pure idiocy.

In fact, even most consumer wireless gear will record the MAC addresses of all computers that have ever issued DHCP requests to them, and these logs usually cannot be purged, even by the owner! When you combine this with the fact that most Cable/DSL service providers will also record your MAC address and bind it to your billing information, and the fact that some of them don't even seem to wait for a court order to turn your info over, it becomes apparent that your MAC address essentially is your name. This isn't even counting the possibility of databases maintained by the major laptop manufactures.

One particularly useful hack I have discovered is that if you change your MAC address, cable and DSL providers typically will give you a new IP address via DHCP. This can be useful if you are a heavy user of P2P networks, since changing your IP regularly can help keep you off the RIAA/MPAA's "big fish" watchlists, since without access to the ISPs internal databases they will be unable to correlate your identity across IP address changes. That is, unless you keep the same "nickname" in whatever P2P app you use...

Thus, changing your MAC address is highly desirable for a number of reasons. However, note that you really only have to concern yourself with your MAC address if you do not wish the local network administrator to be able to identify you. MAC addresses do not cross router boundaries, so anything outside of your LAN will never see it. If your adversary is not internal to the LAN or cannot trace your FIXME IP address back to the LAN, don't worry about your MAC.

Here are the commands to change your MAC for the three major platforms:

Linux

In Linux, you just need to issue two commands, and then re-run dhcp or reconfigure the interface. This works for both wired and wireless cards.

```
[root@machine ~/dir]# ifconfig eth0 down
[root@machine ~/dir]# ifconfig eth0 hw ether de:ad:be:ef:f0:0d
[root@machine ~/dir]# ifconfig eth0 up
```

You can also use the tool 'macchanger' which is available in most of the repositories. It also allows you to set a random mac address.

```
[root@machine ~/dir]# ifconfig eth0 down
[root@machine ~/dir]# macchanger -r eth0
[root@machine ~/dir]# ifconfig eth0 up
```

Use --help to view many more options available.

Windows

Under Windows, however, things are a bit more involved.. There are a few ways to do it. One involves wading through your registry, and will not be discussed here. If you're lucky, you might be able to do it right from control panel. If this is not an option, you can try this ntsecurity.nu utility or this utility.

Mac OS

In Mac OS, for some reason it is easy to change the MAC address of your wired interface. One of the following two commands should work:

```
[user@machine ~/dir]$  sudo ifconfig en0 ether aa:bb:cc:dd:ee:ff
[user@machine ~/dir]$  sudo ifconfig en0 lladdr 00:01:02:03:04:05
```

However, to change the MAC of a wireless interface, you will need to patch your kernel, and then recompile it.

Some OS X users have informed me that USB wireless adapters often allow you to change the MAC address via the command line just like a wired interface. The one most commonly mentioned is the Belkin F5D7050.

I should also note that many routers will allow you to clone or specify a MAC address from their web interface.

802.11 "nickname"

The 802.11 Nickname field is a little-known feature of the wireless spec that sends your hostname to the AP. This is obviously bad.

Linux

```
[root@machine ~/dir]#  iwconfig ath0 nickname "Fucko The Clown"
```

Mac OS

Under Mac OS, your Computer name is your hostname. You can change it with sudo hostname -s "Fucko The Clown". However, your hostname resets every time you restart your computer. To permanently change your hostname, go to SystemPreferences-Sharing. From here you can change your computer name. Your default hostname will then be Example.local

Windows

I think your only option is to choose an obscure machine name. If you wish to change your hostname, you can either edit these registry keys or run the NewSID utility

DHCP Properties

Upon obtaining an IP address, your DHCP client will sometimes send information about you in DHCP requests. In many cases, this includes your hostname and possibly your MAC address, but can include your operating system and DHCP version, which can potentially be very damaging to your anonymity set on your local network. Once again, much like MAC address, if you are unconcerned about your local network discovering your identity, then you probably needn't worry too much about this. However, in many cases it is necessary to obscure this information.

Also, as part of the DHCP standard, all operating systems will provide their most recent IP address to the DHCP server. Usually this is harmless, since it is typically just an internal IP address, but if you use your laptop to DHCP directly to your ISP, it is possible it may then hand this IP to an open access point you associate to. If you are changing your MAC address to minimize risk for P2P activity, you may want to wipe previous IP leases on your router machine every time you obtain a new lease. Typically router devices have a 'release

DHCP' button somewhere on the web interface. If you need to change these settings on your computer itself, follow the appropriate instructions below:

Linux

Unfortunately under Linux, the details of DHCP client properties vary from distribution to distribution. In the general case for dhclient, the values are read from **/etc/dhclient-interface.conf**, but this is typically created at runtime. In Fedora, for example, only the hostname is transmitted, and this value is read from **/etc/sysconfig/networks/ifcfg-ethN** where N is the relevant interface number.

Gentoo (and likely any other distro that uses dhcpcd), however, will transmit the entire OS and kernel version. One contributor suggested changing your **/etc/init.d/net.eth0** (or equivalent) file to include

```
VID=`fortune -o|head -c 30|tr "\"'\n" ' ' 2>/dev/null`
/sbin/dhcpcd -i ${VID} ${dhcpcd_IFACE} ${IFACE}
```

Previous IP/Lease information is typically found under **/var/lib/dhcp/**. Blowing these files away between interface restarts **(/sbin/ifdown ethN ; [change mac addr] ; [change hostname] ; rm /var/lib/dhcp/* ; /sbin/ifup ethN)** should do the trick.

Windows

Windows sends your MAC address and an OS and version string (though the version is something nonsensical, like "MSFT 5.0", which means it might not map to exact Windows versions, but some DHCP implementation version). Unfortunately, Windows also transmits hostname. I don't believe there is any way to stop this, but you should not have picked a hostname that easily identifies you anyways. Unfortunately, it can be used to correlate successive connections, which makes it easier to track you down eventually. It seems as though the utility NewSID can be used to change your hostname on the fly. Alternatively, you can edit a few registry keys.

To kill a previous lease, usually all you have to do is go to **Control Panel->Network Connections** and right click on the interface, and go to **Disable**, and then **Enable**. This will blow away any previous lease state and IP address information. If you prefer the command line, **ipconfig /release** followed by **ipconfig /renew** does the trick.

Mac OS

Mac OS sends just the MAC address and hostname, but you should be aware that it also has some option codes that can serve to

identify the OS type as well. Again, to change the hostname, issue something like **sudo hostname -s "Fucko The Clown"**, and don't forget to change your FIXME mac address.

As far as leasing/IP information, go to **System Preferences->Network** and select **Network Port Configurations** under **Show:**. Then, click the checkbox next to the relevant connection to disable the interface, and then click it again to enable it, and the lease will be blown away and renewed. From the command line, **sudo ipconfig set en0 BOOTP** followed by **sudo ipconfig set en0 DHCP** will essentially the same thing (of course, substitute the appropriate interface name for en0).

If you want to be absolutely sure you have set everything properly, you can download Ethereal to monitor your network traffic. Set the filter line to bootp while a capture is running. This will display only dhcp requests and responses, which you can then inspect for information to make sure everything is OK.

IP Address

The most obvious way you can be tracked across the Internet at large is through your IP address. Yahoo Mail, for example appends an extra header on your email messages that contains the IP of the computer your web browser is on. So much for that anonymous email account, eh?

Luckily, there are at least three ways to change your IP address for Internet traffic. They are (in order of increasing preference):

Proxy Hopping

While easy, Proxy hopping fucking sucks. The basic idea is to find a reliable list of open proxies and change your browser's proxy settings to tunnel your connections through the proxy, thus obfuscating your IP address. BE CAREFUL. Some proxies will REPORT YOUR IP ADDRESS in the form of a cookie/session variable to hosts you connect to. Always test out a new proxy with either of these two proxy checkers. I believe the shroomery list only includes proxies that don't provide identifying information, so it is readily usable by the Firefox extension Switch Proxy. Unfortunately, Switch Proxy has a rather annoying bug that slows new window creation down to a crawl.. They don't seem like they are maintaining it anymore either.

If you're on a Linux machine, there is a utility that will allow you to also use all your command line applications with various proxy types, and will also allow you to chain multiple proxies together, for better protection. This utility is called ProxyChains. It is basically a library that you LD_PRELOAD to intercept socket calls and forward them to a list of HTTP or SOCKS proxies. Sadly, most of these proxies will only allow you to connect to port 80 or 443. Also note that the proxy chain is not encrypted. *THIS MEANS THAT A LOCAL OBSERVER*

AT ANY PROXY HOST CAN VIEW THE REMAINDER OF YOUR PROXY CHAIN, INCLUDING THE DESTINATION HOST!

SocksChains, a similar tool to Proxychains also is available for Windows. FIXME_WIN32: test

SSH Hopping

It is possible to accomplish the same thing as ProxyChaining by connecting numerous SSH connections together. If you have lots of UNIX shells, you may be able to use them to obfuscate your network connection by using ssh's -L and -D options. -L tells ssh to listen on a local port and forward those connections to another host and port through the ssh connection. -D tells ssh to open up a SOCKS 4 server where you specify. So an example session might look like this:

```
[you@home ~]$ ssh -L 4242:127.0.0.1:4242 user1@machine1
[user1@machine1 ~]$ ssh -L 4242:127.0.0.1:4242 user2@machine2
[user2@machine2 ~]$ ssh -L 4242:127.0.0.1:4242 user3@machine3
...
[userN-1@machineN-1 ~]$ ssh -D 4242 userN@machineN
```

So in this case, the first ssh command is run on your home machine. It opens up port 4242 locally, which you will then tell your web browser and other applications to use as a SOCKS4 proxy. Any connection made to this port will be forwarded to port 4242 on machine1. The ssh command on machine1 causes this to be redirected to port 4242 machine2, which has a SOCKS4 server listening, forwarding all connections out through machineN. Of course, if you only need one hop, you can just only use the -D command from localhost. Note that this technique should work on Linux and Mac OS natively, and also Windows, if you install Cygwin. If you set up ssh keys, you can pack this whole procedure onto one line for quick execution.

Apparently Putty can also be used as the first hop in the chain. Port forwarding can be done from the 'Tunnels' item under 'SSH'. You basically enter the local port and then 127.0.0.1:4242 as the destination (don't forget to click 'Add'). To make things quicker, you can save a profile named 'MyTunnel' with these settings (and perhaps also 'Don't start a shell' and 'Don't allocate a pseudo-terminal' if you just want to make the window real small and don't want to show up in the wtmp of the remote machine). Once a profile is saved, you can run putty from the command line or make a shortcut to run 'putty -load MyTunnel'. You can do this for a bunch of different tunnels so you can pick and choose which IP you want to have right from your desktop.

MetroPipe.net has packaged this whole setup behind a nice GUI interface, but before you sign up, you should be aware that the long-lived nature of these circuits, and the fact that a single path shares all your traffic makes this a far less desirable option than FIXME

Tor, discussed shortly. They do however also provide VPN service.

OpenVPN

OpenVPN is awesome. It provides an encrypted tunnel from your computer to the OpenVPN server. While it is not a perfect anonymity service, it is at least useful when you only need "one hop" of anonymous surfing, or when you need to dodge restrictive firewalls. For example, if your employer either blocks or watches your network traffic, you can create an OpenVPN tunnel going to port 53 (DNS, which should be open in just about any firewall) and route all your traffic through it.

In order to make use of OpenVPN, you will need an endpoint running the server. You can either set this up on your home machine as described below, or sign up for an account at a commercial VPN service provider. Vpn-service.us only seems to have a Russian language order page, but I have seen them advertised in English on certain boards. If you ICQ them, they may give you a price in $USD.

Here is a quickstart on setting up a Routed IP OpenVPN v2.0 tunnel for an endpoint you control. This quickstart generates two certificate authorities and 2048 bit keys, making it the most secure way to create an OpenVPN tunnel. It should be immune to MITM attacks, even if one of the hosts is compromised and its keys are taken.

FIXME Downloads are currently broken

Windows

1. Download this openssl config file and place it where you intend to keep your openvpn config files.
2. Generate the server and client keys. You can accept the defaults for all those bullshit info fields, but be careful to say 'y' to the question asking to sign the certificates. The default is # After that script is run, copy all the files that start with **client** to the client side, and all the files that start with server to the server side. All config files and keys go in **C:\Program Files\OpenVPN\config**. Be a dear, and use a secure channel, will ya?
3. Configure server to use 192.168.69.1
4. Configure client to use 192.168.69.2, and to connect to the server. Replace VPN_SERVER_IP in client.ovpn with your server's IP.
5. Place this script to rewrite the client's routes upon connect in your openvpn conf directory. It will be called automatically. However, you need to edit it and replace LOCAL_GATEWAY_IP with your Internet gateway IP (use the top line from **route print** if unsure) and also replace VPN_SERVER_IP with your server's IP.
6. Go to **Control Panel->Network Connections** and right click on the TAP network device to manually set the IP address to

192.168.69.2 and the gateway 192.168.69.1. If you need a DNS server, use one of these.

7. If your server is a Windows machine, right click on its network interface and enable Internet Connection Sharing.
8. Go to **Control Panel->Administrative Tools->Services** and start the OpenVPN service on both client and server.

Mac OS

1. Download the latest OpenVPN source. Install it by opening a terminal, and issuing:
   [user@machine ~/dir]$ tar -zxvf openvpn-2.0.5.tar.gz
   [user@machine ~/dir]$ cd openvpn-2.0.5
   [user@machine ~/dir]$ ./configure --disable-lzo && make
   [user@machine ~/dir]$ sudo make install
2. Download the Tun/Tap driver by Mattias Nissler. Install it with:
   [user@machine ~/dir]$ mkdir tuntap
   [user@machine ~/dir]$ cd tuntap
   [user@machine ~/dir]$ tar -zxvf ../tuntap.tar.gz
   [user@machine ~/dir]$ open tuntap_installer.mpkg
3. Download this openssl config file and place it where you intend to keep your OpenVPN config files (ie /etc/openvpn).
4. Generate the server and client keys. You can accept the defaults for all those bullshit info fields, but be careful to say 'y' to the question asking to sign the certificates. The default is No.
5. After that script is run, copy all the files that start with **client** to the client side, and all the files that start with **server** to the server side. Be a dear, and use a secure channel, will ya?
6. **sudo chown -R nobody:nobody /etc/openvpn** to get all config files to be owned by nobody, then **sudo chmod 755 client-osx-up**
7. Configure server to use 192.168.69.1
8. Configure client to use 192.168.69.2, and to connect to the server. Replace VPN_SERVER_IP in client.conf with your server's IP.
9. Place this script to rewrite the client's routes upon connect in your openvpn conf directory. It will be called automatically, but you might have to **chmod 755 client-up**. Replace VPN_SERVER_IP with your server's IP, and replace LOCAL_GATEWAY_IP with your local gateway to the Internet..
10. Start openvpn on server, and enable NAT
11. Start openvpn on client
12. Add a publicly available nameserver to /etc/resolv.conf.

Linux

1. Download this openssl config file and place it where you intend to keep your OpenVPN config files (ie /etc/openvpn).
2. Generate the server and client keys. You can accept the defaults for all those bullshit info fields, but be careful to say 'y' to the question asking to sign the certificates. The default is No.
3. After that script is run, copy all the files that start with **client** to the client side, and all the files that start with **server** to the server side. Be a dear, and use a secure channel, will ya?
4. '**adduser openvpn**' to add an openvpn user and group to both server and client machines.
5. **chown -R openvpn:openvpn /etc/openvpn** to get all config files to be owned by nobody.
6. Run '**modprobe tun**' or recompile kernels to support CONFIG_TUN (The Universal Tun/Tap Driver) as needed.
7. Configure server to use 192.168.69.1
8. Configure client to use 192.168.69.2, and to connect to the server. Replace VPN_SERVER_IP in client.conf with your server's IP.
9. Place this script to rewrite the client's routes upon connect in your openvpn conf directory. It will be called automatically, but you might have to **chmod 755 client-up**. Replace VPN_SERVER_IP with your server's IP, and replace LOCAL_GATEWAY_IP with your local gateway to the Internet. Replace eth0 with your appropriate interface.
10. Start openvpn on server, and enable NAT
11. Start openvpn on client
12. Add a publicly available nameserver to /etc/resolv.conf.

WARNING: An attentive and fascist network administrator will still be able to determine that you are tunneling packets over an openvpn tunnel by watching your traffic (rest assured, they won't be able to see what you are doing, just that you're doing something). If you work in an environment this oppressive, change the **proto udp** and **port 53** lines in your server and client configuration file to **proto tcp-server/proto tcp-client** and **port 443** (or **port 22**) to make your openvpn session look more like a secure web (or ssh) connection. Note that this comes at a performance price, which is why it is turned off by default. See also: Watching Your Back

For problems in general, the following checklist can help you narrow down the problem to the relevant component:

1. ping LOCAL_GATEWAY_IP
   Pinging the local gateway should be your first check. Sometimes it will be configured not to reply to pings, however it should reply to ARP requests. You should be able to check for ARP entries on all 3 OS's with **arp -a**.
2. ping VPN_SERVER_IP
   This should ensure you can connect to the VPN server through your specific route for that IP. Hopefully it responds to pings,

otherwise check arp.

3. ping 192.168.69.1
   This should check that the VPN tunnel is working. You can also check my.log in your OpenVPN config directory, which should tell if you if the remote VPN server passed the key exchange, etc.
4. ping DNS.SERVER.IP
   The last item you should check is that your DNS server is reachable. If it is reachable, but you still don't have net access, perhaps you forgot to update /etc/resolv.conf or enter a DNS server in your IP properties.

If you still are having no luck, consult the Gentoo OpenVPN FAQ.

Tor

Tor is magnificent. Plus, setting it up is surprisingly easy for what it accomplishes.

WARNING: DO NOT BE A JACKASS OVER TOR. Recently there have been a couple of reports of people using Tor to hack and deface websites. This is not cool, and usually ends up forcing whatever Tor node(s) the attack exited from to shut down. If you insist on using Tor as a cracking/penetration testing utility, PLEASE read the [Combining Approaches] section for some pointers on how to put a proxy or ssh host after your Tor exit.

Both Windows and Mac OS/Linux setup should be pretty similar. For the Windows folk, Tor comes prebuilt as an installer, and these instructions will hold your hand through the process. All you have to do is download the exe file, and run it, and Tor should be installed. Mac OS people can follow the these instructions for their package. Linux people will have to compile and install the Tor source code.

From this point, any user on the system can run tor. Once you do so, your machine is connected to the network as a client node. If you have at least 1Mb of upstream bandwidth, you are encouraged to run a tor server, which is more involved. Note that you do not have to run a tor server to run a tor service, which can come in handy if you have material you wish to [publish anonymously].

The next step is getting applications to use Tor. Tor creates a local Version 4A/5 SOCKS server, listening on port 9050 by default. This means any application that supports SOCKS can be told to use localhost:9050 as your proxy server. However, this is not always desirable or possible. Most web browsers don't support forwarding host name lookups over SOCKS. *This means that it is possible for a local observer to determine which hosts you connect to.* So for full protection, you must install a go-between proxy that does forward host name lookups over SOCKS, such as privoxy. Windows and Mac OS users will have Privoxy already installed and configured as an HTTP Proxy listening on port 8118 as part of the Tor Bundle.

Linux users and DIY folk will want to download this config file for privoxy that starts the proxy server on port 8118 and forwards your requests to port 9050 for Tor. (The thing that has to be changed from the default is to add a **forward-socks4a / localhost:9050 .** line to it).

Newer web browsers such as Firefox 1.5 can be instructed to use SOCKS to do DNS lookups. Go to [about:config about:config] and search for "dns". Set the option **network.proxy.socks_remote_dns** to true. If you do decide to abandon privoxy, you need to be sure to install several extensions to protect yourself from cookies, ads, javascript, etc.

Once you decide how to handle remote DNS, you can then tell your web browser to use HTTP localhost:8118 (if Privoxy) or SOCKS localhost:9050 (if remote DNS) as its proxy server, and you're good to go. Again, you may want to verify your setup with this handy checker or whatismyip.com.

If you intend to use Privoxy to access non-web ports (such as IRC, you may want to edit default.action and replace the line containing **limit-connect** with **+limit-connect{1-}** to allow privoxy to connect to all ports.

If you are looking for an easy way to switch back and forth between Tor and a direct internet connection, the Firefox extension ProxyButton is pretty nice. Switch Proxy is also available, and is a lot more flexible in switching between multiple proxy settings and being able to use a list of proxies in rotation. Unfortunately it slows new window creation down to a crawl.. They don't seem like they are maintaining it anymore either.

If you run Linux or one of the other free Unixes, you can install tsocks, which is an application layer SOCKS proxy. There are several patches available for tsocks, including patches to support Mac OS X (does not work out of the box, sadly). This config file for tsocks tells it to use port 9050. Once tsocks is installed and configured, you can use it to enable any application (ssh, telnet, netcat, nmap, etc) to connect over the Tor network simply by prefixing the command with "tsocks". Note that while the default tsocks does not support SOCKS4A, the Total Information Security version does in fact route DNS requests over Tor.

Some command line applications such as wget, links, lynx and curl will honor the http_proxy environment variable. So you might want to **export http_proxy=http://localhost:8118** if you use these commands often, so that they run through Privoxy.

Windows users may have some luck using TorCap2, which provides SOCKS4A support transparently to applications that lack SOCKS support. Presumably since it does SOCKS4A, DNS requests are not leaked, though I have not verified this personally.

Some web sites have taken to banning the Tor network. Most recently google has started routinely presenting catchpas to Tor users, which require cookies to persist and also javascript to be enabled. In the case of google, Scroogle is available to circumvent the

catchpas, cookies, javascript and other nonsense google throws at you. For other sites, you may want to consider writing custom privoxy rules to combine Tor with an HTTP proxy.

Additional information on connecting specific applications to Tor is covered in the Torify HOWTO and also later on in the [Anonymous Communication] section. One thing you should pay particular attention to is that various browser plugins might not pass through Tor.

In some environments, you may only be able to connect to web or ssl through your firewall. If this is the case (or if you have a network administrator that you believe may watch for "suspicious" employee behavior), edit your torrc and add **FascistFirewall 1** to it. This will cause Tor to only make outbound connections to port 80 or port 443. You can tune which ports it uses through **FirewallPorts**.

> **Note**
>
> Just about everywhere on the Tor website, you are advised that Tor is not to be used if you require true anonymity. Further sections of this HOWTO address (almost) all of the application level issues involved in anonymity, so pretty much all that remains of that paranoia are attacks on the tor network itself, which, unless you are trying to hide from the NSA itself, you really don't have much to worry about.
>
> You should, however, be aware of the possibility of malicious exit nodes. Exit nodes can both observe and modify traffic, which means they can do things like keep AIM/non-SSL web sessions open after you close them. You should also be aware of is that it is somewhat dangerous to mix traffic that can identify you with traffic you wish to be completely anonymous. Since Tor multiplexes multiple TCP sessions over a few circuits, it is possible for the exit nodes to determine that the two were sent from the same host.

I2P

I2P is a complementary network to Tor. While Tor is useful for establishing anonymous connections to the external Internet but doesn't have the bandwidth/architecture to support Peer to Peer and bulk traffic, I2P is designed specifically to provide an anonymous internal network upon which you can run any normal Internet server, including Peer to Peer applications.

While the install itself is typically straightforward, I2P's setup is a little more involved than Tor's, mostly because every client node is also a relay node, and as such you may need to have a hole punched in your firewall to forward port 8887 for UDP. I2P does have NAT and firewall traversal mechanisms, but they do not work in all cases (especially symmetric NATs). Once you have it installed, you can go to the config page and check to see if it is able to connect.

I2P opens a proxy server on localhost:4444, but most likely you will want to add a line to your privoxy config so you can use both I2P and tor. Placing **forward .i2p localhost:4444** (no period) below your Tor line will allow all requests to .i2p domains to go through I2P. *DO NOT configure privoxy to only filter .i2p domains without also using Tor for everything else.* It is possible for eepsites to have images, cgi, or iframes from non-i2p servers that they control, and can thus discover visitor's IPs that way.

Note that I2P does have gateways to the external web, so you technically don't need a Tor line at all if you want to forward everything to localhost:4444, but there are only two I2P exit points, and they are not nearly as reliable as Tor's. Once Privoxy is set, you can go to Orion.i2p to browse the list of all .i2p sites on the net, or to search.i2p or eepsites.i2p to search the I2P web. Note that by default, you will not have the host entries for many sites listed in orion and the search engines. You can fix this by adding **http://orion.i2p/hosts.txt** to your addressbook subscriptions

If you would like to search/browse the I2P web without installing I2P itself, you can use tino's DNS-based I2P in-proxy by appending "tin0.de" to any i2p domain name to access it from the public internet.

The really beautiful thing about I2P is that it is effectively an anonymous networking layer that applications can be developed on top of, including Bittorrent and Gnutella. In addition, through a feature called I2PTunnel, you can create a tunnel to just about any TCP based service you wish, including anonymous .i2p site of your own. In particular, you can point your IRC client at localhost:6668 and you will then be on the I2P anonymous IRC network (note that your IRC client does not need a proxy server for this).

Once you're all set up with the basics, you can browse over to The Ugha Wiki and check out some HOWTOs for doing various tricks over I2PTunnel and I2P in general.

SLIRP

SLIRP is a magical tool that converts a normal non-root UNIX shell account into a PPP session. While obviously the conversion isn't 100% (for example pings don't work), it does allow you to set up a pseudo-VPN to a machine you don't have root on (and thus couldn't run OpenVPN). Unfortunately this technique is likely to only work on Linux/UNIX and Mac OS, since it requires that your PPP setup use an ssh session as the serial device. I doubt Windows PPP can do this, but I could be wrong.

For our limited purposes, we'll just discuss using SLIRP over ssh. To do this, you need to do a couple things. First, you must download and compile SLIRP on your UNIX shell (not your home machine). This is pretty straightforward and does not require root. You can just run configure and then copy the slirp binary to someplace like ~/bin. If you need to be covert, you could try calling it zsh or mutt or something.

```
[user@machine ~/dir]$ tar -zxvf slirp-1.0.16.tar.gz
[user@machine ~/dir]$ cd slirp-1.0.16/src
[user@machine ~/dir]$ ./configure && make
[user@machine ~/dir]$ cp slirp ~/bin/zsh
```

Next, you will have to make sure the host you connect to is in your known_hosts file. Usually this means you have to connect to it using root on your home machine (ie **sudo ssh user@shellhost**). If you get an error to the effect of "Host key verification failed" during connection, this is the reason for it.

You might also have to set up passwordless ssh to your UNIX shell. It makes things more convenient, and for some broken versions of ssh it is required (such as the one that ships with MacOS 10.3). If you don't get a password prompt or get some ssh error when running the following scripts, you probably need ssh keys.

Once this is accomplished, you can run a script to start pppd using ssh: Linux version, MacOS version. FIXME: The Mac OS version is causing problems with pppd either hanging up (10.3) or not connecting at all (10.4). Is this an issue common to FreeBSD as well? Could any BSD users give this a whirl and report back?

Run this script as root on your local computer. It runs PPPD, calling SLIRP through ssh on the other side. Note the variables you must set at the top. You can obtain your local router in Linux or Mac OS by issuing **netstat -r** and looking for the entry at the bottom for default. For **PUBLIC_NAMESERVER** you can cat /etc/resolv.conf on your UNIX Shell, or you can use one of the publicly available nameservers. I've already filled in two public nameservers for you. Also, the pppd should attempt to grab the remote shell's DNS server and write it to resolv.conf for you, but this doesn't always work. Hence I write to resolv.conf in the script before calling pppd, just in case pppd fails to write it.

Double Black Magic IP Wizardry

Sometimes one approach is not sufficient by itself to both protect your identity and grant you access to all the services you would like. For example, Tor does not carry UDP traffic, nor is it really an all-encompassing VPN solution, which can lead to lots of problems with browser plugins ignoring your proxy settings, spyware, webbugs, etc.

In some cases, it is also not possible to use Tor to connect to services due to banning. In other cases, it may actually be desirable to conceal the fact that you are exiting from the Tor network to protect it from abuse complaints, which usually have the side effect of forcing Tor nodes to shut down.

Tor followed by HTTP Proxy

This is probably the easiest way to conceal the fact that you came from the Tor network. If you are using Privoxy, you can chain a regular HTTP proxy from a list mentioned above (this one provides IRC capable proxies, while this one automatically builds a list of currently active "high-anonymous" proxies) after the Tor line in your config to bounce off the HTTP proxy before hitting the server you wish to access.

To connect to fascist.torhater.com through an HTTP proxy at somehost.net:8080, add **forward-socks4a fascist.torhater.com localhost:9050 somehost.net:8080** to the end of your privoxy config file. Note that with Privoxy 3.0.3, somehost.net:8080 must be an HTTP proxy and cannot be SOCKS.

If you are having problems locating a proxy that is not already banned by your desired service, you can attempt to scan for a fresh one using either ScanSSH or YAPH. Once you find a proxy, you can either google for the IP to see if it's in any proxy lists, or if you're ambitious, use a perl script to check to see if it has been listed yet in the DNS RBLs by doing DNS queries for it. (For those writing their own script from scratch, note that the IP must be backwards. So query 5.13.42.23.dnsrbl.org to check 23.42.13.5).

Tor followed by SSH-tunneled SOCKS4 Proxy

Unfortunately, many IRC servers ban known proxy servers, and some will even scan your source IP for an open proxy before allowing you to connect. However, what you can do is combine the SSH hopping approach with Tor if you need Tor-caliber anonymity. Basically the procedure is to obtain a UNIX shell (using Tor and an anonymous email address), and then use tsocks to connect to it with **tsocks ssh -D 4242 user@someshell.net**.

Once this is done, you can add **forward-socks4 fascist.torhater.com localhost:4242 .** (the dot is not a typo) to your Privoxy config if the service is a website, or otherwise inform your IRC client to use localhost:4242 as a SOCKS4 proxy. In this way, you are connecting to your locally listening ssh client, which routes it through the Tor connection to someshell.net, at which point your traffic exits on to the Internet with the IP address of someshell.net.

Once again, note that SSH only supports SOCK4, and thus DNS queries will be made locally and thus can be observed. Also, if you are using one of the free UNIX shell accounts, please be courteous and don't make a nuisance of yourself. These people don't want to and shouldn't have to take time out of their day to answer abuse emails about your account. They do this for free.

Also, note that Putty can be used with Tor to perform this same technique by setting localhost:9050 as your SOCKS4 proxy. Then you

can set up the SSH proxy by going to 'Tunnels', and filling in 4242 as the destination port, and clicking 'Dynamic', with no destination or hostname. When you click 'Add', D4242 should show up. As discussed before, you can save this profile and make a shortcut to 'putty.exe -load myprofile' to quickly establish your tunnel. Once you run putty, you can use localhost:4242 as your SOCKS proxy, or add the above line to your privoxy config.

OpenVPN over Tor or HTTP Proxy

A few people have mailed me asking about using OpenVPN over Tor. It turns out this is not as hard as I originally thought. There are several reasons you might want to do this. In my opinion, the main one would be to obtain protection against plugins/non-SOCKSified applications making connections that reveal your address. If you are running OpenVPN, all of these applications will go through the VPN.

Essentially the main problem is that you want your normal Internet traffic to go over the OpenVPN interface, but you need your Tor traffic to travel over your regular Internet interface. Essentially this involves setting up routes for every Tor server you intend to use as an entrance point. This can be very problematic, because there are many Tor servers. However, you can limit Tor's choice of entrance servers with the torrc config options **EntryNodes nick1,nick2,nick3** and **StrictEntryNodes 1**. I would recommend picking a couple high bandwidth servers off of the Tor Network Status page. If you are running Tor 0.1.1.x or greater, you should also use **LongLivedPorts** and add your VPN server port to the list (or just run the VPN server port on one of the ports mentioned in the manpage, such as 5190, 5050 or 6667). Remember that Tor exit servers block most ports below 1024, and also common P2P ports. *Be careful not to choose one of these as your OpenVPN server port, or you will be unable to connect.*

Once you have selected a few nodes and edited your torrc accordingly, you basically can follow the OpenVPN instructions exactly, except for 3 differences:

1. You need to edit the server.conf (Linux, Mac OS, Windows) to have the line **proto tcp-server** instead of **proto udp**.
2. You need to edit the client.conf (Linux, Mac OS, Windows) to have the lines **proto tcp-client**, **socks-proxy 127.0.0.1 9050**, and **socks-proxy-retry**.
3. You need to modify the client-up script (Linux, Windows, Mac OS) and instead of just one **route** line with your VPN_SERVER_IP, you need a route line for the IP of each node you chose for **EntryNodes**.

After that, you're pretty much good to go. Doing this over HTTP proxy is essentially the same as Tor, except you only need one route, and the config options are **http-proxy** and **http-proxy-retry** instead of socks.

Note

Using OpenVPN over Tor can be considerably weaker than simply using only Tor. On the one hand, you don't have to worry as much about applications/plugins not using Tor, but on the other hand, you do have to worry about OS-level leaks through the VPN interface, and also the fact that there is a fixed endpoint to correlate traffic through. If you use OpenVPN for more than just one-hop encryption, I would recommend only using it over a throwaway computing setup, to avoid risks of attacks, trojans, and other leaks through the interface.

SLiRP over Tor or HTTP Proxy

Running SLiRP over Tor is desirable for similar reasons as running OpenVPN over Tor, but has the benefit that you only need a shell account on some UNIX machine. Unfortunately, it has the disadvantage that it is only possible under Linux and Mac OS. As far as I can tell, Windows cannot use an arbitrary terminal to run PPP over. However, it is still possible to set up a Linux box doing SLiRP over Tor, and have it NAT for your windows desktop, if you were so inclined.

But I digress. Again, like OpenVPN over Tor, you have to choose a few high-bandwidth, reliable and trustworthy entrance nodes from off of the Tor Network Status page. Then you have to add these servers to your torrc with **EntryNodes nick1,nick2**, and also set **StrictEntryNodes 1**.

Then, follow the instructions for SLIRP setup, but instead use a different client script: Linux version, MacOS version (FIXME: again, MacOS version has bugs.. See SLIRP for details). The script needs the variables at the top to be filled in, as well as the routes that are added for FIRST_TOR_IP, SECOND_TOR_IP and so on. These must be replaced with the numerical IP addresses for the Tor nodes you added in your **EntryNodes** list.

Remember that pings will not work, so test this with telnet or something, rather than ping.

SLiRP over SSH hopping (and Tor)

An alternative approach is to ssh hop to one shell, and then use **tsocks** to ssh to the next shell where you either are using -L to forward to an OpenVPN server, or where you are running a SLIRP host. You then tell the OpenVPN/SLIRP client side that localhost:4242 (or whatever) is your server and adjust the routes accordingly.

Here is how to accomplish this for SLIRP, which I expect to be the more common use case. If anyone does this for OpenVPN, please send me your scripts and I will post them. I imagine it should be pretty similar. You start by constructing the typical SSH hop chain:

```
[you@home ~]$ ssh -L 4242:127.0.0.1:4242 user1@machine1
[user1@machine1 ~]$ tsocks ssh -L 4242:127.0.0.1:4242 user2@machine2
...
[userN-1@machineN-1 ~]$ ssh -L 4242:127.0.0.1:22 userN@machineN
```

You then can use this script (Mac OS) to launch SLIRP across the SSH hopped link. Notice that the parameters are a bit different than with the vanilla SLIRP setup. **FIRST_IP** is the IP of your first SSH hop (machine1). The rest of the hosts do not matter. The rest of the options are the same as above.

One thing that may improve your security is to make the first SSH hop to a box you control. From there, you can install tor and use tsocks for the SSH connection to the host that will be running SLIRP. This is a good way to get Tor-level security for UDP applications and apps that don't support SOCKS. Technically the first hop doesn't have to be your box, since you don't need root to install Tor as a client, but the first node might be able to reveal who you are if they decide to cooperate with the destination host of your tsocks connection for some reason (which they may do, if they believe you compromised them or something).
[[Category:Computer network security]]

Local Programs and Services

The programs running on your computer can give away information about your identity. Particularly those involved in file transfer and logging in to other computers.

ident lookups

ident is the TCP identification service. It allows a remote host to determine the local username associated with any TCP connection involving that remote host. Naturally this is a concern, especially if your username reveals your true identity. So when do ident lookups happen? Well technically your machine's ident server will answer any request for a (server, client) port pair for which the destination IP is the same as the ident request source IP. However, normally is only used with FTP, SMTP and IRC traffic, if that. Some web and ssh servers also have it enabled. The best thing to do here is to kill your ident server, or add a firewall entry for port 113. Even better, you may wish to create an obfuscated or common username for regular use. Something like bob, jane, Acidburn, or ZeroCool, perhaps. ;)

```
    [root@machine ~/dir]# iptables -A INPUT -p tcp --dport ident -j DROP
```

ftp logins

Be aware that your FTP client may also transmit your username or email address as the anonymous password independent of your ident response. So far I've tested Firefox, links, and ncftp, and none of them report your username OR hostname in the login. So that's good.

Telnet

It is possible for a telnet server to query any arbitrary environment variables from your telnet client. These include USER, HOSTNAME, DISPLAY, etc. However, the default behavior of Linux telnet is to only send DISPLAY and PRINTER. Note that in some cases, DISPLAY may contain your hostname.

ssh keys

The major threat with ssh keys for Mac OS and Linux users is your ssh host key. This can be used to fingerprint you by connecting to port 22 of your IP to verify that you are using the same machine as some other previous IP, either at your ISP or over VPN.

On both operating systems, you should be able to regenerate new ssh host keys with the commands:

[root@machine ~/dir]# ssh-keygen -t rsa -f ssh_host_key.pub
[root@machine ~/dir]# ssh-keygen -t rsa -f ssh_host_rsa_key.pub
[root@machine ~/dir]# ssh-keygen -t dsa -f ssh_host_dsa_key.pub

On Mac OS, these commands should be issued while you are in the /etc directory, and you should use a sudo before them. On Linux, run them as root in the /etc/ssh directory. Use no password for the keys.

ssh login attempts reveal information about your machine only if you have created ssh private keys and the ssh client attempts to use them to log in to the remote host. Password-based login leaks no information about the client other than the IP address.

Realistically, even the scenarios for proving a client's identity via an ssh key exchange are very unlikely. In the case of unsuccessful private key attempts, the only way it could be done is if the attacker's ssh exchange were logged, and then the attacker's private key

was seized and demonstrated to provide the same signature as given to the remote host. To do this would require an obscene amount of data collection at the remote host end, just waiting for the attacker to connect. However, if an attacker logged in successfully via an ssh key, all that would need to be shown is that the ssh key existed on your machine to prove you were the attacker. Your local username is never sent as part of the ssh key exchange, even if it is a part of the public key.

Terminal Services/rdesktop

By default, both rdesktop (for Linux) and Microsoft's Terminal Services Client (mstsc.exe) will send your hostname and username to the machine you connect to. In rdesktop, you can override the username with the -u switch, and the hostname with the -n switch. In the MS Terminal Services Client, you can change your username in the "Options" button, but it's not clear that there is any way to avoid transmitting your machine name. Again, in Windows you can change your hostname via either NewSid or via the registry.

SMB/NMBD

Machines running windows file serving broadcast their computer name and description in SMB Master Browser Elections. You probably don't want this traffic spewing across your network connection if you wish to be anonymous. This is how you turn it off in Windows.

In general, it's a good idea not to name your machine something like "Bob Fnord's Evil Hacking Box of Doom", or "JoeSmithsLinuxBox".

Under Mac OS, these services can be turned off under the "Sharing" icon in System Preferences. I would turn just about all of those off if I were you.

Under Linux, you can either remove smbd and nmbd from /etc/rcN.d/ or you can run chkconfig smbd off and chkconfig nmbd off. Note that this just prevents the services from starting. To shut them off, run /etc/init.d/smbd stop (and again for nmbd).

FIXME_WIN32: Server and TCP/IP netbios helper? Is netbios EPM?.. Also check snmp.

mDNSResponder (Bounjour/Rendezvous/ZeroConf)

mDNSResponder is Apple's implementation of ZeroConf, which is used to configure your computer on a network automatically. It also can be used to announce information about your iTunes, iPhoto, and iChat profiles. Obviously this may be undesirable. To turn it off:

[user@machine ~/dir]$ sudo /System/Library/StartupItems/mDNSResponder/mDNSResponder stop

And to re-enable it:

[user@machine ~/dir]$ sudo /System/Library/StartupItems/mDNSResponder/mDNSResponder start

To permanently disable it, you can erase or move the mDNSResponder directory from the StartupItems folder.

It should be noted there is also an mDNSResponder installed by default on some Linux systems. You probably want to remove it from /etc/rcN.d, or run chkconfig mDNSResponder off. Don't forget you also have to stop it with /etc/init.d/mDNSResponder stop, since chkconfig only removes it from bootup.

An mdnsresponder.exe is also installed with the Windows version of iTunes. You probably want to remove it/rename it so it is not started. You can check with Task Manager or Process Explorer to see if you have a copy running.

UPNP

UPNP is Microsoft's half-assed attempt at a ZeroConf protocol. It basically does the same thing ZeroConf does, and probably should be disabled. Here is a utility to turn it off. Note that you don't need their utility to turn it off. You can go into Control Panel->Administrative Tools->Services and first STOP and then DISABLE the "Universal Plug and Play Device Host" service. Do the same with "SSDP Directory Services".

[[Category:Computer network security]]

Web related leakage

Your web browser leaks a frighteningly large amount of information about you. For example, even after you protect yourself by concealing your IP address through Tor, it is still possible for someone to use a Java program to obtain your actual source IP and hostname. And this is only the beginning. An excellent test utility to test all the different types of information that can be obtained from your browser is available at BrowserSpy. Most of these can be handled by the excellent Firefox extension NoScript. You are encouraged to test out your browser there in addition to looking over the following material.

Cookies

Cookies can be used to track your web usage across even a Tor session, where each connection originates from a different IP. This can be both a blessing and a curse. A blessing because if you are actually logged in, you usually won't have to keep doing so. But a curse in that if someone then obtains your computer, they can use the cookie values to prove you were at a given website at a given time. They are also extremely dangerous if you have a tendency to turn off Tor or your proxy config from time to time, and then wander back to a site that has a unique ID for you.

In particular, sites with ad banners can catch you off guard, since often they can have a small ad banner with a piece of javascript you don't notice. If they set a cookie in this banner, they can easily correlate your Tor traffic with your non-Tor traffic. For sites like these, the best thing you can do allow cookies from the originating website only in your web browser preferences window.

There also are a few Firefox extensions you can use to mitigate an arbitrary site's ability to track you via cookies. The first, Cookie Culler provides a toolbar button that allows you to purge all but selected "protected" cookies, and also provides you with the option of blocking cookies you have deleted before. Add N' Edit Cookies will allow you to search for cookies by site, modify, remove, and add them. Also useful is CookieButton, which is a handy toolbar option that allows you do access control and clear cookies for a given site right off of the toolbar. You can permanently disable cookies for entire domains, such as doubleclick.net or google.com.

Browser User Agent And Capability Info

Sometimes a very unique User Agent string (Ex: "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.5) Gecko/20041107 Firefox/1.0 RealMedia 1.1.3") can be a giveaway. How many people will have that Gecko build date combined with that version RealMedia? On Linux? The solution is the User Agent Switcher, which will allow you to set your user agent to whatever you wish (be sure to pick something common to avoid fragmenting your anonymity set). Be sure to check the "Reset User Agent When Browser Closes" option, or bad things will happen. Even still, you may find yourself stuck with a browser that doesn't want to start. In which case, you might have to edit prefs.js by hand, and remove all the lines that contain "agent".. It's been a while since this has happened for me, so perhaps they've finally gotten all the edge cases where it can exit with the wrong agent set.

As mentioned above, another possibility is a malicious site can use Javascript to detect all sorts of information about your browser. This can also be used to track you. The best defense against this is to use NoScript.

Referrer Url

Likewise, your referrer can be used to track your session path from page to page as well. In order to prevent Firefox from transmitting referrer information, go to [about:config about:config] and set **network.http.sendRefererHeader** to the value 0.

Tab Mix Plus can also be used to block referrer forwarding for the lifetime of any given tab. Simply right click on the tab, and go down to Permissions. The option to block referrer is there, along with several other options we will use later.

Sometimes, however, it is useful to outright lie about where you came from. For example, if you frequently visit your own blog/website by typing in the URL, those visits are distinguishable by the lack of referrer URL. In those instances, you may wish to use refspoof instead. Refspoof adds the ability to specify your referrer in a URL. For example, *spoof://nytimes.com;ref://google.com* connects you to the New York Times, making them think you came from google.

Browser History

This should be obvious. Delete your browser history, cache, cookies and other personal information after doing sensitive things. A frightening javascript hack] can actually reveal if you have been to select sites, which can be used to fingerprint you. The best way to deal with this is to run NoScript.

Web bugs

A common technique (used by the FBI and others) to discover someone's identity is to send them an email with an image or some other document attribute that their email client or browser will attempt to load. Usually, this image will be located on a server that the sender controls, which means they will then have your IP address once your browser makes the connection. Of course, for normal images, these will obey your proxy settings. However, note that even using Tor is sometimes insufficient. Make sure your proxy settings have an entry for FTP, especially.

You need to be particularly careful about Java plugins. Have a look at this clever java applet that turned up at this odd website. This is an improvement on previous Java bugs in that it is able to select one of several potential ways to make an external internet connection and bypass your proxy settings based on capability information provided by the JVM. Even if the JVM has been secured (most, including Sun's, are not), it can still query the local interface of your machine and get the IP address without even making an internet

connection. It then submits results back to the server that served the applet. In the case where the JVM is not secure, the JVM will IGNORE YOUR PROXY SETTINGS AND MAKE A DIRECT CONNECTION TO THE ORIGINAL WEBSITE.

You basically have two options when dealing with web bugs of this nature. One is to install the NoScript Firefox Extension, which allows you to whitelist Java, Javascript, and Flash objects on an as-needed basis. This is the recommended option, as it covers all your bases all the time unless you say so. The one downfall is that if you enable permissions on a site, you enable it for everything. This means that if you enable Javascript for your email provider (most don't work too well without it), and they do not scrub HTML properly, someone could still feed you this Java applet. According to the NoScript Website, the ability to split permissions for Java from Javascript is being developed for an upcoming release. Your best bet until then is to globally disable Java from your Firefox preferences.

The other option is to use Tab Mix Plus tab permissions to disable plugins on the current tab (right click on the tab). You get finer granularity here, but your choice only persists in the current tab. This is annoying and easy to forget.

Also be aware that some media objects can reveal your IP address, depending on how the plugin was written. On Windows, I have tested Windows Media Player, Realplayer, Quicktime, and Flashplayer. Of those, I have discovered that only Realplayer did not honor proxy settings, since it launched an entirely new application. I have tested mplayer-plugin and Flash on Linux, and unfortunately the mplayer plugin does not obey browser proxy settings (but does obey the http_proxy environment variable). In general, the best way to determine if a plugin/media type is obeying your proxy settings is to use Wireshark to watch network traffic. The display filter *'tcp.port == 80* or *tcp.port == 443* can make it easier to find traffic that is bypassing Tor, though note that some media apps will use other ports and possibly even UDP.

Desktop and Web Browser Extensions

The major threat with web browser extensions and desktop plugins is that they will transmit a unique user id over the same Tor circuit that you happen to be using to anonymously access a website. Weather monitoring extensions are particularly dangerous because they can transmit zip codes or even address information to retrieve local weather conditions. Likewise for link collection services such as delicious, stumbleupon, and flickr.

It is also possible to install a malicious Firefox extension to track your whereabouts on the web. The most surefire way to watch for this is to manually view the Extensions.rdf file in your extensions directory under your Firefox profile. Each cryptically named subdirectory of the extensions directory should have an entry in the RDF file. Make sure that the plugin name is something you remember installing.

Intrusive Surveillance

THOMAS: What is this thing?

TRINITY: We think you're bugged. Try to relax. Come on, come on...

TRINITY: CLEAR!

THOMAS: Jesus Christ! That thing is real?!

Intrusive Surveillance basically means any type of surveillance that is occurring due to some form of intrusion into your machine. As such, it is the most difficult form of "forensics" to defend against, since doing so involves the securing and hardening of your operating system against attack. While keeping up with security patches is a necessary condition to be secure, sadly it is not sufficient. In spite of this, there are several obvious indicators that someone is investigating you/watching what you do with your computer.

I should probably start this section off by saying that if you need anonymity, you should probably reinstall your system now. Especially if you use Windows and Internet Explorer. It is way too easy for those machines to become infected with spyware if you haven't been practising safe computing up to this point. I've even seen spyware that modifies the Internet Explorer user agent string to contain a unique 128bit identifier.

Root Kits

Root Kits are the most intrusive and stealthy form of system surveillance around. They are typically designed to take complete control of your operating system kernel, causing it to lie to you about what processes are running, what network connections it is making, system diagnostics, and so on. Luckily, if someone just wants to spy on you, they are much more likely to only install a keylogger rather than a full blown rootkit. However, knowing some basic info about how to detect rootkit installation is helpful for finding keyloggers as well, especially since as keyloggers grow in sophistication, the line between them and complete rootkits will blur.

For most users, this section probably covers a threat model they do not need to worry about (although with rootkits being used by RIAA goons, this is rapidly changing). Many users will want to just skip to the section on watching your back, which describes how to use

rootkits to hide various items on your system from a fascist administrator and for plausible deniability ("a hacker did it! Who installs a rootkit on their own machine?"). Still, even in this case a read over this section is recommended, since it will tell you how to undo what you have done.

Linux

On Linux, there are a couple of things you can do to protect yourself from rootkits. The easiest, least technical method is to do an **ls -laR / > dirty.list** on your live system, and then compare this to an **ls -laR /mnt/hdd > clean.list** on a Knoppix live CD. The command line utility diff -u clean.list dirty.list can be used to compare the two listings. The utility xxdiff may make this easier to view if there are lots of differences due to an encrypted filesystem not being mounted, proc missing, udev, etc.

If you intend to audit your machine while it's live, the very first thing you need to do is ensure that your kernel hasn't been modified via a rogue driver module. While **lsmod** will list loaded modules, it is possible (and quite trivial) to remove module names from this list while keeping the module in the kernel. For this reason, if you are concerned about local surveillance, you are advised to build a kernel without module support (preferably with grsec, which adds protection against ways to force modules into a module-free kernel). This will prevent someone from loading a module that could hide processes and network connections, log keys, etc.

If this is not an option (it is very difficult to get a monolithic kernel running, especially on systems that require closed-source drivers), you can perform a manual audit. I provide a couple Linux kernel modules that can aid in the auditing process. They will help you check for common ways rootkit modules hide themselves, and also can help you to ensure your kernel syscall path has not been tampered with. The ideas are based on this Security Focus article and this article on rootkit operation. All methods discussed there can be examined with those modules. The reason I created these modules is that Fedora's policy of disabling /dev/kmem has broken existing utilities that performed these functions. In particular, samhain is capable of performing these checks automatically for you on non-Fedora and *BSD systems.

Once you know your kernel hasn't been subverted, you can check the output of **netstat -natup** to see if any strange programs have established network connections to external hosts. **ps xa** will show you all the processes running on your system. If you don't recognize something, scroogle it.

To help watch for modification of your userland programs, you can run Chkrootkit and/or Rootkit Hunter, and also periodically run **rpm -Va** to verify checksums of your installed packages. TripWire and AIDE are also options for maintaining system integrity.

Mac OS X

If you have a second Mac, the easiest way to scan for a rootkit that is hiding files is to run an **ls -laR / > dirty.list** on your live system and compare this to an ls -laR /path/to/disk > clean.list run from a "clean" Mac that has mounted your hard disk drive in Target Disk Mode. The command line utility *diff -u clean.list dirty.list* can be used to compare the two listings. Compiling the utility xxdiff may make this easier, but you have to run it in the X windows emulator, which is painful.

As far as live inspection and countermeasures, unfortunately, Mac OS doesn't have the advantage Linux has in being able to compile a monolithic kernel. This means that it is much more difficult to ensure that your system integrity is still valid. Just like on Linux, it is possible to write a kernel extension that goes in and verifies that none of the system calls have been hooked, and that the keyboard interrupt vector is still intact, but I don't think such a extension has been written yet. FIXME_MAC: any takers? FIXME_MAC: how about gdb + the syscall table? Is there a map file?

There is a piece of malware that exists for Mac OS X called Opener. It functions as a trojan, spyware, and a keylogger. It's worth searching your filesystem for it's presence. FIXME_MAC: more details on how it works..

However, since opener is not a kernel-level rootkit, it can be detected with system monitoring software such as CheckMate. Additionally, you can check for suspicious TCP and UDP connections via **lsof -i TCP -i UDP**. However, note that kernel-level rootkits like WeaponX are able to make themselves invisible from these checks.

Windows

There are a few rootkits for Windows available which are capable of hiding processes, services, files, etc. In addition, most rootkits come with some form of device driver that is installed in order for them to do their cloaking by directly manipulating kernel objects.

For general rootkit detection, Sysinternals puts out a tool called Rootkit Revealer that uses various Windows APIs to check for inconsistencies (ie comparing raw registry data to API reported, etc). It does not use this information to detect rootkits by name, but instead prints a list of anomalies. If you have a clever rootkit implementer, be aware that they can hide among the noise.

If you are interested in looking for hidden files, or executables, your best bet is to create a bootable CDROM from your Windows XP install media. You can then look through your directories after booting from this CD and ensure they match your system under normal boot. UBCD has a tool called RootKitty which basically compares file listings of your computer from the CD versus while it is running on a normal boot.

Specific rootkit removal tools exist also. Currently the most popular of these are BlackLight and ICESword which both check for changes made to kernel memory to detect rootkits like FU, as well as hooks to functions in kernel space. In 3rd place is VICE, which

checks for hooks in commonly targeted kernel and userland functions. The main issue with VICE is that for usermode, it will spew out one hell of a lot of false positives, as it is common for the Windows DLLs to "hook" one another.

The RKDetector utility linked off of this Hacker Defender Removal page is also pretty nice. While it has been written primarily to detect the Hacker Defender rootkit, it is also capable of finding hidden processes and services that may have been cloaked by other rootkits (such as FU).

Also available is UnHackMe, a shareware tool designed for AFX removal.

Lastly, just like Linux and Mac OS, Windows enables you to list active Internet connections and their associated processes. **netstat -nab** should do the trick.

FIXME_WIN32: Do AppInit_DLLs show up in procexp? How about other injected code? How do you know what is injected?

Keyloggers and Spyware

If someone is out to spy on you, by far the most likely thing they will do is install a keylogger. Some keyloggers can be easy to find, some almost impossible. There are two types of keyloggers, software and hardware.

Hardware

The main thing to watch for is an extra extension jack coming between your keyboard and the back of your computer. However, also be wary of internally installable keyloggers. If your physical environment can't be trusted (note that depending on your situation, this may or may not include your home), buy a new keyboard and seal it with epoxy, or some tamper evident mechanism. Also periodically check the inside of your computer for dangling pieces of electronics coming between your keyboard port and your motherboard. Normally there should only be wires or nothing at all. People have been prosecuted using keylogged data as evidence.

Software

Windows

Your best bet for guarding against software keyloggers in Windows is to install some form of anti-spyware software. My personal favorites are Spyware Doctor, Microsoft's AntiSpyware, AdAware and SpyBot (in that order), since they all provide free versions that are

fully capable of removing spyware they discover and also have received good reviews on the web. Although I have not tried it, Symmatec's Antivirus+AntiSpyware is supposedly very good as well, though pricey.

Because these scanners are unlikely to detect custom spyware, you probably should also give Sysinternals AutoRuns a try, to catch anything that may be scheduled to run that has not been signed/verified. And of course, don't forget to use the techniques discussed in the rootkits section to detect anything that may be attempting to hide itself via a rootkit.

For the technically curious and/or those concerned about custom keyloggers, there are two main types of windows keyloggers:

1. Message-Hook Based
   Being relatively easy to code, message-hook based keyloggers are the most common form available. It is estimated they comprise 90% of keylogger installs.
   Their simplistic nature means that they are relatively easy to detect as well. Follow the same steps involved in looking for a Windows root kit: Check for strange processes, and check the AppInit_Dlls registry key. Alternatively, you could install this anti-keylogger. It will run for 4 hours per reboot for 10 reboots before you have to pay for the registered version. It actually watches members of your message queues and notifies you when processes attach to listen for events. FIXME_WIN32: Are there more subtle ways to inject DLLs using hooks such that you can hijack, say, explorer.exe to do your keylogging? Yes. How do we enumerate them? SetWindowsHookEx with WM_DEBUG. Also !chkimg in kd.. CodeMe.
2. Device Driver Based

   To date, I've only run into one commercially available implementation of a device driver based keylogger on Windows. Basically the way it works is it hooks itself onto your keyboard device driver and then writes the keys to a file, which can be viewed with an external program.
   While in theory a kernel mode driver can be made next to impossible to detect, in this case it is quite easy to find. You have a few options. The easiest thing to do is go to Control Panel, click on Keyboard, and then click on the "Driver Details..." button. That will list all the .sys files involved in making your keyboard work. The ones with a green checkmark are signed by Microsoft, and clicking on them will tell you so. If you see any that lack a green checkmark, they are most likely a keylogger. Alternatively, you can run **regedit.exe** to search your registry under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class for either "Keyboard" or "kbdclass". This is where all your device drivers live. Once you find the keyboard driver, check to make sure it ONLY has kbdclass under the UpperFilters value. If it has anything else after kbdclass (specifically the unsigned driver from control panel) edit it and remove it. Then search again for that name under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services and delete any subtrees that come up. FIXME_WIN32:

Screenshots

Linux

For Linux, the rootkit detection techniques described above apply for keyloggers as well. Most Linux keyloggers will take the form of rootkits. Also, you should watch out for trojaned binaries (use **rpm -Va** or install tripwire or AIDE), especially ssh. You should also keep an eye on your aliases and shell rc files (eg ~/.bashrc and ~/.bash_profile) to make sure no one is sneaking an LD_PRELOADed library in on you. Last, but *DEFINITELY* not least, you want to make sure your ~/.Xauthority file is not world readable, and that you don't have any bizarre xhost entries. It is possible to remotely capture X events (and thus keystrokes).

General Techniques Against Keyloggers

Keyloggers are typically pretty blind. Especially kernel and hardware keyloggers. If you are at a machine you cannot trust and do not feel like making it trustworthy, you do have some options to at least protect your passwords. For example, switching windows mid-password, cutting and pasting characters, and using the mouse to delete sections of text randomly are all effective against hardware and kernel keyloggers, which will only focus on actual keyboard events.

Message-hook and other application-level keyloggers can also be fooled in this way, but they can attempt to do things like sort keystrokes by destination window, target only specific apps, take screenshots, and even attempt to capture cut and paste events. In fact, most commercially available software keyloggers have advanced to the point where they are able to track both cut and paste activity and sort keystrokes according to their destination window. You can still attempt to confuse them by entering keys into other fields in the same window, however.

Watching Your Back

Ok, now that we know how to find and remove keyloggers and rootkits, we're going to talk about how to use them to conceal what you are doing, and to watch for evidence of nosy coworkers. Using Hacker Defender with this config file will hide an OpenVPN installation. copy **c:\windows\system32\cmd.exe Desktop\mycmd.exe** will give you a command shell on your desktop that is still able to see the Hacker Defender config files.

Be aware that antivirus software may detect hacker defender, especially before it has a chance to run. The README advises you insert <, >, ", and & characters randomly into the config file fields to help avoid detection, though obviously this is not fool proof. To conceal

the Hacker Defender executable, you may wish to run it through Morphine, which is an executable encryptor. Even this is probably not foolproof, so use with caution if you are in an environment where the network administrator receives virus information on your PC.

FIXME: In a future revision, write up how to turn a webcam (and/or audio mic) into a security camera, to watch over your computer while you are gone in order to catch a 3rd party in the act of installing/retrieving a hardware or software keylogger at your computer. An excellent writeup for Mac OS can be found at engadget. For Linux there is Gspy, SCRAP, and Motion. A HOWTO on building a mini computer dedicated just to video surveillance on Linux is available at MagicITX. For windows, Dorgem has been discontinued, but it allows you to capture video when the camera detects motion.

On Windows, it is also possible to install monitoring software such as those listed here to monitor your computer. Promising candidates include PCSpy, SpyMyPC, All in One Keylogger and BlazingTools Perfect Keylogger.

FIXME_WIN32: maybe also write a section on how to elevate privs on your own machine. Can OpenVPN be installed from a BartPE? Many people will not have Administrator access on their machines. Test on vmware. You probably can't add yourself to Administrators, since that is a Domain group.. Can you create a RunAs.. shell?

Throwaway Computing

In certain situations where you have freedom over the computer but do not trust it, you might consider using a bootable CD. The obvious choice is Knoppix or a Knoppix derivative. I prefer Auditor Linux since it automatically supports my wireless card (unlike Knoppix) and comes with lots of useful security tools right on disc. It is possible to customize Knoppix using a USB key, so a Knoppix CD+tor on your USB key might be helpful if you frequently find yourself in shady labs. The Knoppix Wishlist includes Tor, so maybe that will happen someday.

For pre-configured, pre-tested Tor environments, you can try out Tails Live or Anonym.OS, which are boot cd that include Tor and have everything preconfigured to use it. Alternatively, you can save yourself some effort and purchase a bootable USB Key with Tor and other privacy software pre-installed.

Alternatively, you can build a BartPE image for the same purpose if you prefer Windows. While there are a couple of massive plugin directories, my favorite route is to use The UBCD Installer (which contains many of those plugins) and then just keep any extra apps I want (such as Tor) on my USB Key. The TorPark project is excellent for this purpose.

Either of these methods provide maximum protection and assurance against software trojans and viruses. In addition, you

automatically get protection against cookie logging and browser history data for free.

An alternate (and possibly more convenient for home use) method is to use VMWare (or Xen) to create an innocuous looking operating system to interact with the real world. This has the advantage that if you need to use your original setup for something, you can, but for all other communications you have a system that you wouldn't mind being attacked. Furthermore, VMWare has a feature that allows you revert to a known safe snapshot of the OS at any time, which can be useful to ensure you haven't been trojaned or acquired any persistent cookies during your session. The snapshot feature of Xen is still in development, but various hacks are possible to get the same effect.

Search and Seizure

The last and most perilous threat to your privacy is when The Man busts in and takes all your gear (and they will take all of it). This sucks, and will often leave you without computer equipment for the better part of a year. There are limits on scope of warrants, but the courts have proved to err on the side of The Man. Of course, it is always advisable to practice good Kung Foo so that he's never able to trace you in the first place, but luck favors the prepared. You never know when some scumbag decides to turn states evidence, or some enemy of yours decides it might be funny to see you sweat out an investigation for no reason. As such, lets spend a bit of time discussing how search and seizure functions in the US.

Warranted and Warrantless Search

There are several methods by which an Agent can obtain the legal right to search and seize your digital goods, most of which are conveniently outlined in the US DOJ Search and Seizure Manual (local copy). There are a couple instances where they can get away with searching you WITHOUT A WARRANT that you might not have anticipated:

Private Searches

These are probably the most dangerous type of warrantless search, because it is most likely to catch you off guard. Essentially a private search is a search conducted by someone who is not acting as an agent of the government (ie a vigilante). In this case, a neighbor, roommate, officemate, janitor, repairman, maid, sysadmin, etc can be snooping around on your computer, discover what they might think is criminal evidence (such as this HOWTO :), and call the police. The police are then legally authorized to repeat the search conducted by the private citizen without warrant (unless it was a search of a residence), and arrest you for both the original and any additional contraband (within the scope of the original search). Bad news all around.

Searches/Consent by Employers/Coworkers

In this case, not only can employers and coworkers conduct "private searches", but they can also consent to search of your office space for you. How nice. Don't forget: you're their slave, coppertop.

Implied Consent

This is another beauty. If you've signed away or otherwise have been warned of a reduced right to privacy (ie through a work contract or login banner), you can be considered to implicitly consent to search. I wonder if this is why AOL changed its Terms of Service (before changing them back due to prompt public outcry, and zero corporate media coverage).

Exigent Searches

If the feds have reason to believe that there is extreme urgency in obtaining the data due to either danger to person or threat of its destruction, they can search without a warrant. Thankfully "laptop batteries might be about to run out" was ruled not to be an exigent circumstance. But note that some Agent did try to use that as an excuse.. Gotta love The Man.

No-Knock/"Sneak and Peak" Searches

If the feds can show that there is either danger of violence or threat of destruction of evidence if they announce their search, they can obtain a "No-Knock" warrant, which allows them to barge right in if you are home or not. Also, with the passage of the Patriot Act, they are also able to conduct "Sneak and Peak" searches. Typically "Sneak and Peak" search cannot involve seizure, and the authorities must typically notify you within 90 days. For defensive techniques against "Sneak and Peak" searches, consult the section on watching your back.

Subpoena of ISP Records

A subpoena is a court order to testify or produce evidence. The downside is it can be used if you are not suspected of a crime but instead may only have evidence relating to a crime. It is possible for the feds to obtain your ISP records, files, and "opened" emails via subpoena. However, they do have to notify you of this fact immediately, unless there is fear of danger to person, flight, destruction of evidence, or otherwise jeopardizing the investigation.

Deception of Purpose

This little doosey was a fun one to discover. How to Be Invisible cites a January 1994 FBI Law Enforcement Bulletin that states agents may disguise themselves as utility repairmen, delivery personnel, distressed motorists, etc, and ask for entrance into your home to make a phone call, look up something on the web, etc. Once you grant them entrance, anything they happen to see (or claim to smell) in plain view can be used to go back and get an actual warrant. Good times!

Trash Inspection

Ruling that there is no expectation of privacy for discarded items, the Supreme Court has held that trash is not protected by the 4th amendment. This ruling has allowed the government to tell garbage collectors that certain trash bins are to be delivered directly to their doorstep, without warrant. They do not even have to collect it themselves.

Retroactive FISA Warrants

The USA Patriot Act has expanded the ability for the FBI to obtain secret warrants against "terrorism" suspects. At the direction of the Attorney General, the FBI is able to conduct a secret search of your residence, phone communications, and/or Internet activity and then 72 hours later, apply for a secret search warrant to do so from the FISA court. Since the FISA court is secret, and has only turned down 4 requests in the past 5 years, this power has essentially destroyed any guarantee against unreasonable search. The Patriot Act also amended 18 USC 2518-7 to allow for "specially designated" law enforcement officers to conduct warrantless search and surveillance for 36 hours in situations of suspicion of organized crime, national security interest, and threat of serious physical injury to a person.

Search of Corporate Records

In an astounding display of new advances in Constitutional interpretation, the Supreme Court ruled in 1976 that since corporations are not natural persons, they do not (nor even their owners!) enjoy any protection from the search and seizure of their records. My oh my, that certainly makes it convenient to fetch just about any information on anyone. With this ruling, the US government is able to enumerate just about everything in your home simply by demanding purchase records from your credit card company. The ruling also extends to subpoenas of logs and other files at colocation providers, and also provides the basis for the "constitutionality" of National Security Letters.

It should also be noted that anything uncovered while executing a warrant is admissible in court, even if it was not what the agents were looking for.

Civil Procedure

In addition to the above, you need to be at least peripherally aware of the rules of discovery in civil procedure, especially if it is likely that someone may seek to sue you for damages instead of (or in addition to) pressing charges. In civil procedure, if there is reason to believe that you may have evidence supporting the plaintiffs claim, the process of discovery enables them to demand evidence/records from you. If you destroy this evidence, then not only are you potentially liable for criminal charges, but the plaintiff also is allowed to assume that the destroyed records contained the proof they sought. In the case of civil litigation by large corporations against individual people, the police are sometimes called in to immediately seize relevant materials without warning. Isn't that great?

Encrypted Filesystems

The solution to these perils hinges on cryptography, and each system has its own way of accomplishing this. Depending upon your threat model for The Man, you may want different levels of assuredness that he cannot obtain your data. As described above, your two main classes of threat are civil action, and criminal action.

In the case of threat of civil action, it may be desirable to employ some form of steganographic filesystem so that the process of discovery cannot be used to assume you have destroyed incriminating evidence. Your best bet for this is TrueCrypt, which has an appealing hidden volume mode which can provide deniability for civil situations where you are compelled to give up the key during discovery. It exists for Windows and Linux, but since it is the only non-broken implementation of an encrypted filesystem for Windows, the writeup for it is right below the Windows sub-section. If you decide to use TrueCrypt for Linux, you should be aware that there are secondary logs (ie bash history on Linux) that can be used to demonstrate that files exist if they are not carefully purged.

In a criminal situation the rules are a bit different. In the US, you may have some luck in claiming that your encryption key is protected by your 5th amendment right to not incriminate/testify against yourself. However, be aware that if you are subpoenaed to testify against another individual, you can be ordered to give up their key, unless they are your spouse. In the event that you are ordered to testify against someone else, you can request immunity from the prosecution to protect you from any incriminating evidence found as a result of the key disclosure.

Your ability to assert your 5th amendment right (and thus be eligible for immunity) ultimately rests with the decision of the judge. If he thinks your 5th amendment right does not apply to the key due to the lack of real threat of incrimination or some other twisted legal logic and you still refuse to surrender it, you can be sentenced to up to 6 months in jail for contempt of court. Note that you can still be charged with contempt for refusing to obey even if you believe a higher court would rule in your favor. However, refusing to comply

would get you a good deal of (most likely positive) press attention. Nobody likes to see people imprisoned for refusing to testify against themselves, even if some legal loophole would allow it.

There has been recent political maneuvering in the UK to attempt to enable Part III of the Regulation of Investigatory Powers Act to give the government the power to demand your keys even in the cases where it may incriminate you.

This is particularly short sighted for a number of reasons, the most obvious being: the ramifications of the damage of trust in SSL certificates and banking communications; the inability to discern what is encrypted data and what is simply random application data; the inability to discover or prove with any certainty exactly how many passwords there are; the unspecified language as to whether key files count as password, and what happens if they are lost or destroyed; and the inability to prove that the subject hasn't legitimately forgotten the password (which, with the infrequent use patterns of filesystem passwords, is entirely possible and even common among users). The rest of the world should thank the UK if decides to take it upon itself to prove the stupidity of this action for us. I have no doubt that this measure cannot survive in any country with a legitimate constitution or other declaration of human rights, for good reason. Get ready, hilarity is about to ensue.

So enough of that. Lets discuss filesystem cryptography on each of the 3 major platforms, as well how to erase data securely as you move it from non-encrypted storage to encrypted storage. As usual, *BSD users are left to fend for themselves. If anyone would like to submit a quick and dirty BSD writeup for this HOWTO, don't hesitate.

Under all of the following systems, you will need to make one or more passwords for each encrypted volume. *You should avoid writing these passwords down at all costs*, but note that filesystem passwords are particularly easy to forget, since they are used infrequently. As mentioned above, this fact makes mandatory key disclosure particularly short sighted. It is very possible to forget filesystem keys and risk complete data loss. For this reason, you should mentally rehearse your passwords every day for several weeks after you create them to make sure you do not forget them. I consider myself fairly mentally competent, but I have still lost more than one encrypted volume after creating it because the password was used once for creation and then forgotten.

Linux

There are two main cryptographic filesystem solutions for Linux: dm-crypt and TrueCrypt. TrueCrypt setup is covered below, and arguably it has some more appealing features than dm-crypt, but since it is not included in any major distributions, typically you will have to recompile your kernel to support it.

Setting up dm-crypt is relatively easy to do (at least for a simple loop-back filesystem), and their wiki has several good HOWTOs.

Unfortunately, setting it up right can be extremely hard and involved. *Please read this guide carefully, as there are many subtleties than can catch you off-guard.*

To protect against the legal snafus mentioned above, I prefer a bit different approach than that given on the Wiki for actual device creation. I prefer to use GPG to encrypt the filesystem key, and have the passphrase I type into the keyboard be the password to the GPG key. This enables you to change the password without having to rebuild the filesystem. It also enables you to carry the key with you on a USB microdrive to ensure its safety and also to prevent anyone from mounting the fs even if they know the password. In emergency situations, the USB key can be destroyed, and the data can never be recovered. In this way, you can be in full compliance with a court order requiring the password for the filesystem and still not reveal your data. Note that if you destroy a key after a court demands to see it (or simply refuse to give up the password) you can be held in contempt of court and sentenced to jail time (in the US, this is 6 months or less, however). However, if the key/data is a substantial portion of the prosecution's case, and the sentence you are facing is more than 6 months (or if you are a hardcore civil libertarian type), you may want to tell them to fuck off anyway. Probably would get you a good deal of (most likely positive) press attention. Nobody likes to see people imprisoned for refusing to testify against themselves, even if some legal loophole would allow it.

> **Note**
>
> Since The Man will usually attempt to take all of your electronic gear right away, you will have to find some mechanism to either store the key some place safe or have an instantaneous mechanism to destroy it as soon as you hear the knock. Be advised that if your key media fails, you will lose all your data. Floppies are a no-no, but can be used to provide plausible deniability.

Here are the steps to generate such a key that can be destroyed on a moment's notice:

```
[root@machine ~/dir]# dd if=/dev/random bs=4k count=1 | gpg -a --cipher-algo AES256 -c - > /mnt/usb/keys/fs.gpg
[root@machine ~/dir]# gpg -q -o - /mnt/usb/keys/fs.gpg | cryptsetup -v -c aes create cryptfs /dev/hdxN
[root@machine ~/dir]# mkfs.ext3 /dev/mapper/cryptfs
[root@machine ~/dir]# mount /dev/mapper/cryptfs /crypto
```

So basically what this does is get some random data for the fs key material, and use gpg and AES256 to symmetrically encrypt (-c) it with your passphrase. The next command then decrypts your key file and uses the key material to initialize the dmcrypt driver using /dev/hdxN, where x is one of a-d, and N is the partition number. Note you can also use files instead of partitions, but it is not recommended, especially if that file resides on a journaled filesystem. After that, the /dev/mapper/cryptfs block device will appear, and

you can format it for whatever FS you like, and then mount it.

For added safety, I prefer to move /var, /tmp, and /home to /crypto and create symlinks back to /, so that .bash_history and system logs aren't available to someone who might want to prove you have certain files or access times. You should run **telinit 1** before doing this, to ensure that no daemons are running and actively using those directories when you move them.

```
[root@machine ~/dir]# telinit 1 [or reboot into single user mode]
[root@machine ~/dir]# [killall rpc.idmapd]
[root@machine ~/dir]# [umount /var/lib/nfs/rpc_pipefs]
[root@machine ~/dir]# mv /var /tmp /home /crypto
[root@machine ~/dir]# ln -s /crypto/* /
[root@machine ~/dir]# [vim /etc/selinux/config]
[root@machine ~/dir]# telinit 3 [or reboot]
```

On Fedora Core 4 systems, you'll need to **killall rpc.idmapd** and possibly **umount /var/lib/nfs/rpc_pipefs** before the **mv**, or just reboot into single user mode. In addition, this whole setup is likely to cause SELinux conflicts, so you should probably set **SELINUX=permissive** or **SELINUX=disabled** in /etc/selinux/config (or add **selinux=0** to the kernel boot parameters in /etc/grub.conf).

Once this is complete, you'll want to make sure that your crypto fs is mounted before anything tries to use /var. The way I prefer is to create a script interface to gpg that has the right options to enable it to work from /etc/rc.d/rc.sysinit. For Fedora Core users, you can typically just call that script right after the rest of the local filesystems are mounted. Search the rc.sysinit file for "mount -a -t" or "Mounting local filesystems". You should end up somewhere near a bunch of mount -f lines and an SELINUX relabeling call. Stick a call to **/path/to/mount-crypto** right before the SELINUX stuff. If you prefer to run your system in runlevel 5 (with graphical login), you will need to edit /etc/grub.conf and remove the **rhgb** option from the kernel config line in order to be able to enter your FS password. Note that you will probably want to have a boot disk handy or be ready to do **linux init=/bin/bash** from the boot command line in case something goes wrong.

Alternatively, if you don't wish to be prompted for a password at bootup because the machine is a remote server, you can use **chkconfig** or edit /etc/rc.d/rcN.d (where N is your runlevel -- type **runlvl** as root if unsure) to remove syslog, sendmail, crond, atd, and any other daemon that shows up in an **lsof -n | grep var and lsof -n | grep tmp**. In summary:

```
[root@machine ~/dir]# runlvl
[root@machine ~/dir]# lsof -n | grep var
[root@machine ~/dir]# chkconfig --level 3 syslog off
```

```
[root@machine ~/dir]# chkconfig --level 3 sendmail off
[root@machine ~/dir]# chkconfig --level 3 crond off
[root@machine ~/dir]# ...
```

Unfortunately, there are likely to be a crapload of daemons you're going to have to do this for, especially if you run Fedora Core 4. Once you finish this, you'll want to make a script that mounts and then starts all the needed daemons. As a starting point, you can have a look at my crypt-start and crypt-stop.

Note that if your system is running remotely, it may not be too happy about brining sshd up for you after the unmount or upon bootup, unless you **mkdir -p /crypto/var/empty/sshd/**, and **mkdir -p /crypto/var/lock/subsys** in the *unmounted* dm-crypt directory.

It should go without saying, but if you go through all this trouble to encrypt your harddisk, you shouldn't leave backups lying around on unencrypted media. If you have to transfer a backup to unencrypted media, tar it up, and then use **gpg --cipher-algo AES256 -c** to encrypt it. GPG does compression before encryption.

Last, but not least, you should also consider encrypting your swap so that pieces of programs you run aren't recoverable after shutdown. I prefer to use the lazy route and just make a swapfile on the encrypted filesystem.

> **Note**
>
> An alternate procedure to protect /var and /tmp is to encrypt your entire root filesystem and place the decryption scripts on an initial ramdisk. I have not done this, because it means that you cannot reboot your servers remotely, but it requires a hell of a lot less hacking with initscripts and SELinux permissions. As you saw above, this process can get pretty involved.
>
> The Gentoo Wiki has a page on setting up an encrypted root filesystem for Gentoo, and Linux Journal has an article on setting up an encrypted root for Fedora Core 3. This Ubuntu Forum Post describes the same process for Ubuntu.
>
> Debian's installer has the capability to create partitions that are encrypted with a key that is randomized and reformatted every boot. This can be used for swap and /tmp partitions that cannot be decrypted once the system is shut off.

Mac OS

In MacOS you have two options. If you trust Apple (and your sysadmin, where applicable), and really believe there is no master password set, you can use the built-in FileVault feature to encrypt your entire home directory with 128 bit AES, or you can attempt to do

it yourself. Given the amount of work involved in doing it yourself, I would suggest trusting Apple and going the FileVault route.

Windows

In Windows, everything is easy. Unfortunately, everything also sucks. For some reason, Windows implements encryption at the filesystem layer, and you can enable it by right clicking on a file/folder and going to **Properties->Advanced**. Unfortunately, all of the file names in an encrypted directory are still viewable without the key. Worse still, the Administrator account has access to all these files through a special recovery key, and there seems to be no option to disable this.

If you do choose to run Windows on your desktop, an alternative you might consider is building a Linux fileserver that houses all your sensitive documents on an encrypted Samba share, but be aware that Samba does not encrypt traffic. You can set up an OpenVPN tunnel between your Linux and Windows machine, however, if there is danger of someone monitoring your network.

TrueCrypt/FreeOTFE

Another alternative is to use TrueCrypt or FreeOTFE, which actually provides the benefits of both crypto and steganography in that it has an emergency password that you can use if ordered to reveal your filesystem password under threat of force. Both of these programs are good, and more than makes up for the pile of suck that is NTFS encryption, and now also has a Linux/Windows Mobile version. Creation of hidden volumes is pretty straight forward. There is online documentation for both of them (truecrypt - freeotfe), but really you only need to be aware of a couple of things. First, it is best to create the hidden and the outer volumes at the same time. The outer volume is created first, and is populated (by you) with non-sensitive files and encrypted with the emergency password. Once the volumes are created, you can mount the same volume file/partition with either password to test it out.

Additionally, they both have support keyfiles. The support for this feature is particularly excellent. This allows you to create a collection of both fake and real keyfiles such that your adversary has to know which subset are actually required (assuming they can even find any of them), in addition to knowing your password. It can be seen from this formula (or by visualizing keyfile "fake/real" as a binary string), that for N keyfiles, the total number of combinations of keyfiles is 2N. Thus you can easily provide for a large number of possible combinations of both keyboard and mouse input (which is particularly comforting if keyloggers are a possibility). Couple this with hidden volume support and clever concealment of encrypted volume files amongst other large and unintelligible program data files, and you have yourself a pretty secure and undetectable encrypted setup. Just remember to frequently mentally rehearse all of your passwords (instead of writing them down), as mentioned above. Filesystem passwords are used much less often than login passwords. It is easy to forget them.

Newer versions can also encrypt the entire system partition including the Windows Swap file, standby/sleep file, hibernation files and all the other little things that can save sensitive information across session.

On Linux, however, an encrypted volume can of course contain a swapfile as discussed above, and on PDAs swapfiles don't exist

SeizeD

For the extra paranoid, you can write a quick perl script to monitor network connectivity, and immediately unmount and remove the crypto device as soon as pings fail (or execute any other arbitrary command). I've done this already for you. My first cut was a simple script that pinged a series of hosts and executed your crypt-stop script from above. This script gave The Man one second to move your machine to a network that would also respond to all those pings while he transported it.

This is problematic in that The Man could simply throw in a hub with a bunch of machines that would also respond to those IPs and turn it on as soon as they disconnected your box from the network (assuming they figured out what to do about power). So after thinking about it for a bit, I decided I didn't even want to make it that easy for that bastard. So I wrote a pair of scripts you can run on various machines on your LAN (or across the Internet) to ensure network connectivity.

The way this works is there is a client script and a server script. The client script is the one you run on your secure machine, and the server script you run on any host on the Internet. The scripts are written in standard perl, and depends upon the Unix utility md5sum, which is available on Linux, Mac OS, and cygwin. You should be able to replace md5sum with any command line hashing program, such as FSUM if you do not want to install all of cygwin.

When you start the pair (start the server first), they ask you for a password to be used for that session. The client script then periodically (every 0.25 seconds by default) sends the MD5 (or SHA1) hash of a random number (from /dev/urandom) to the server script, and the server script appends this random number to your password. It then hashes this combined value, and sends the result back to the client. The client compares this value to one it generates locally via the same manner. If they match, the process is repeated with a new random hash, if they do not match, a script you specify (such as crypt-stop to unmount your drives) is run. The script is also run after a timeout period (1 second) or if the TCP connection otherwise dies.

This is a common cryptographically secure authentication technique that is used to prove two people know a password without revealing it to a third party (The Man).

Note that there is nothing stopping you from running multiple copies of this program on a given machine to connect to multiple servers with different passwords, in case there is concern a password could be recovered by attacking a particular server. It won't hurt your

volume to attempt to unmount it twice.

Secure Deletion

Oftentimes you will have old or temporary copies of data left on your hard disk after you finish making your encrypted filesystem. Sometimes applications will save data to unencrypted locations by default before you realize what they are doing. In these cases you need to have a mechanism to wipe traces of this data clean. Simply deleting files is not enough, since deletion only removes files from the directory listing and does nothing to actually remove their contents until they are overwritten by some new file.

Whenever the topic of secure deletion comes up, an argument will inevitably be raised as to how many times a file must be overwritten and what must it be overwritten with in order for it to be truly gone. The tinfoil hat crowd will tell you all sorts of horror stories about this or that government agency that has the power to read through N+X layers of random overwritten data, where N was the number you asserted was secure and X is some arbitrary additional amount they made up to make you feel bad.

My personal opinion is that somewhere between 2 and 5 really is all you need. As drives become larger, the cost factor involving finding data below an arbitrary number of writes over the span of the entire disk grows tremendously. And then who's to say that the data wasn't there from a previous owner, possibly even someone who returned a drive back to the manufacturer because of some defect that was corrected and the drive resold. The other factor is that if secure deletion takes forever, you will find yourself doing it less and less, and postponing it more and more because it will interfere with your real work. This is obviously much worse than minimally wiping something quickly right away and getting it over with.

Flash drives (and SSDs) are difficult to wipe due to the internal wear-leveling preventing the OS from writing to the same flash block repeatedly. The file may be made inaccessible through the drive interface, but if the flash chip is removed from the drive, it can be completely dumped out.

Linux

On Linux, the relevant utility is called wipe. There are two versions of this utility. The more popular one is hosted at sourceforge, and the other is available here.

For most uses, I would just accept the defaults. To wipe a file, **wipe filename** should be fine. **wipe -r directory** will get an entire directory recursively. To wipe all free space on a drive, **wipe -a some_file** should do the trick.

Modern Linux filesystems cannot be wiped at the individual file level due to the internal journal saving changes to the files. These include reiserfs, ext (>=3) and btrfs filesystems.

Mac OS

On Mac OS, secure file deletion is built right in to the trash bin. You need to ensure that you do not interrupt this process, however, or you may lose access to any FileVault volumes you may have. Keep the power plugged in.

It is also possible to erase free space on your Mac if you emptied the trash bin without using the secure file deletion option. For instructions on how to do this, go to **Applications/Utilities** and select **Disk Utility**, and select your Macintosh HD. From here, under the "Erase" header, select Erase Free Space. This overwrites all the empty free space on your hard drive. However, if there is material on your hard drive that is unreadable by OSX (i.e. Windows Partition, etc.), this will remain untouched.

Windows

On Windows, far and away the best option is Eraser, since it adds right-click context menus to wipe a file or directory, and allows you to schedule tasks to wipe all free space as well. Very nice piece of software.

OpenBSD

Use '-P' parameter of rm command for secure wiping in OpenBSD.

Anonymous Communications

Anonymity is hard. One f**k up and the game is up. The art of remaining anonymous is constantly evolving and what works one day may not work the next. There are very few people that know *all* of the ways communications are monitored and how to protect your privacy. There are however some best practices that you must use in order to give yourself the best chance.

Anonymous Email

Low-grade anonymous email cannot effectively be achieved by using a mainstream email provider. Yahoo and Hotmail services will append your IP to the mail headers. So be absolutely sure to never send any messages (or even log in) without using Tor.

Worse still, Google and other mainstream webmail services typically offer other services that "conveniently" allow you to share the same account/cookie between them. Cookies can grab your MAC address on your network interface, GeoIP locate your exact address, and keep a log of all actions ever made with that MAC address.

Of course you need to be careful with things like entering your street address/zip code into their corresponding mapping services, or for that matter, ever using a yellow pages. Careful and judicious use of various cookie control mechanisms or throwaway computing is required. It is also rumored that hotmail will pull your browser time info and place it on emails, thus narrowing your geographical location.

As the final nail in the coffin against Google and company, these providers scan in real time each and every email ever received to your account in storage for advertising classification and government agencies. Every you have thought you had deleted can be obtained in a court order.

Your last option for anonymous mail is to use a proper mix network. However, these networks require a good deal of configuration and setup to join, and once you do, they are only one way. There are two main anonymous remailer networks in existence, MixMaster and MixMinion MixMinion is designed to succeed MixMaster, but it is still in development and thus has debug logs, etc in place that can be confiscated and used to betray anonymity. There are web gateways available to use, but again they are only one way. Note, however, that the last mixminion release was in 2007 and the last mixmaster release was in 2008. The project lead of mixminion says that it is "mostly dead" (http://archives.seul.org/mixminion/dev/May-2010/msg00001.html).

It is also possible to set up a return path, or Nym through certain mix networks. Hushmail provides nym service as part of their paid accounts, and Panta Rhei maintains a list of NymServers as well. As of 2011/08/13, www.panta-rhei.eu.org does not resolve and panta-rhei-eu.org has no whois entry. It appears that nym.mixmin.net is up as of 2011/08/13.

If you only need a throwaway email address for or for signing up for a google groups or other forum account, you can use Mailinator.com, dodgeit.com or pookmail.com. Note that these temporary mailboxes have no passwords. Also don't forget to use Tor or some other IP obfuscater

Note

If you use a webmail account, you should expect that your email is NOT PRIVATE. According to the ECPA (see also), after

180 days it becomes possible to demand email from a server without a warrant, and for non-criminal matters. This means all that has to happen is a civil attorney decides they want to see your email because they might have a reason to sue you, so they write a subpoena demanding all email older than 180 days from your provider, and it is theirs.

A few interesting anonymity/privacy mailing services have also arisen lately because of this loophole. StealthMessage, Self Destructing Email and MailJedi all provide "self-destruct" capabilities for email, so that you don't have to worry about messages you send sitting in someone's inbox to be discovered later. StealthMessage for some reason does not work for me, however. It also requires Javascript and is pretty clunky.

Once again, *I would not rely on any of these services to actually destroy your mail or otherwise keep it private, especially in the case of subpoena, National Security Letter, or coercive tactics.* If you need this level of assurance, you must manage your own GPG key using a front end or plugin to your mail client.

Question: What about Tor Mail? The operators of TorMail are anonymous and should *not* be trusted with your private messages. There are allegations online that TorMail is run by the Russian government, use proper methods (throw away addresses and throw away GPG keys) for safety measures.

Posting to Usenet

For the benefit of the unwashed: Usenet is a massive collection of discussion groups spanning all sorts of topics. Just about any type of discussion you might imagine takes place on Usenet, and you can browse and search all posts ever made via google. There's just one problem. You can browse and search all posts ever made via google. This means that if you ever post something to Usenet, it remains there. Forever. Thus anonymity is highly desirable.

Posting to Usenet is actually easier than writing anonymous email, because you don't have to set up the return path. In this case, you can simply use a Mixmaster web interface (see also) or some other remailer (use Tor) and send mail to one of the mail2news gateways. You can then view your results on Google Groups or one of any number of public NNTP servers.

To post a reply to a given post, you need to enter an "In-Reply-To: <MessageId>" header line with the Message ID of the message you would like to reply to (in addition to the usual "Re:" subject prefix). You can find a message's ID via google groups by clicking on "Options" and then "Show Original". Since it's relatively easy to screw this up, please practice this in misc.test or alt.test before posting to real groups.

Unfortunately, many newsgroups are unavailable via the mail2news gateways. To post to these groups, you will either have to create a google account (which is problematic due to a universal cookie google creates discussed previously), or sign up to a commercial Usenet provider and pay via Money order. If you plan on signing up to a commercial usenet provider, you should ensure that they enable web access, because there are few Tor exit servers that will allow you to access the NNTP (Usenet) port. Some examples that may meet your needs include Usenet.com, NewsFeeds.com, NewsGroups.com, Binaries.net, and MegaNetNews.com. Be sure to avoid the temptation of using the same account for anonymous posting as you use for downloading warez/movies/etc from the binaries groups, since most likely you will be unable to do the latter over Tor.

Also be aware that there are two limitations with google's news server. The first is that google.com keeps a cookie that tracks which groups you have visited. This cookie persists for multiple sessions and is potentially shared with their main search page, and every other google service. It's not too much of a stretch for them to also track IPs that have used that cookie as well (or worse, save info about map queries), meaning that if you forget to use Tor and access google groups, or any google service, they can potentially correlate your interest in one particular anonymous post to your IP via the cookie that was used both times. The solution is to either use a bootable CD for this sort of work, or be diligent about purging cookies. The same goes for posting to web forums.

The second issue with google groups is that some people configure their clients to append an X-No-Archive header, which prevents google from keeping the post on its servers. This means you may be unable to see replies unless you use a public NNTP server, especially in some privacy conscious newsgroups.

IRC/Instant Messaging

If you need to talk to a bunch of people quicker than Usenet allows, or wish to meet with a particular person anonymously, then IRC is probably your best bet. You most likely want to avoid Instant Messaging, since it is too easy for a third party to profile your social network. Furthermore some IM networks enable you to put in an alias for your friends. Many people will just set this as your real name. The problem with this is that it is transmitted to the IM server, which means all that has to happen is for anyone who knows your real identity to set an alias, and bingo, your real name has been revealed.

Choosing an IRC Client

For Linux and Windows IRC clients, I recommend Gaim / Pidgin. Veterans may balk at my choice, but Gaim / Pidgin is nice for a few reasons:

1. It supports OTR and Gaim Encryption
   OTR and Gaim Encryption are person-to-person encryption methods. This is useful when you need to be on an IRC server that you can't trust. Gaim Encryption works a lot better as far as enabling itself automatically, but it tends to be worse off at handling two locations for the same buddy than OTR is. Both of them have the nasty property of getting confused when either you or your buddy use different clients (like at work and at home, for example), but OTR is easier to reset.
2. It supports Tor hidden services
   Gaim speaks SOCKS5, which means you can use any of the Tor hidden service IRC servers. You can either set a global SOCKS5 server under Preferences->Network, or you can set it per account that you add, under "Show More Settings".
3. It doesn't respond to CTCP TIME
   CTCP TIME is a request you give to a client to ask it what time it thinks it is. This can reveal your timezone and thus general geographic location. In general, when discussing time or planning meetings with people, you should give the time in UTC, to be both considerate of their timezone being different than yours, and to avoid giving your location away.

A close second to Gaim is X-Chat, which is available for Linux, Windows, and Mac OS. X-Chat doesn't support OTR or client-based encryption, but it does support IRC over SSL, where as Gaim currently does not. X-Chat supports SOCKS5, so hidden services should be accessible. X-Chat WILL respond to CTCP TIME, but it has a convenient menu option that allows you to edit it (hidden under Settings->Lists.. CTCP Replies).

Also for Mac OS, Adium does support OTR, but most likely won't support IRC until the v1.5 milestone for group chat is reached.

Diehard command line users can use irssi or any other client with tsocks (that version supports hidden services!), but they should remember to do **/ignore * CTCPS** and **/ignore * DCC** to block CTCP and DCC as well. irssi can also be configured to use privoxy as an http proxy directly without the need for tsocks. Note that for this to work, you need to dig through the privoxy default.action config file to change the **limit-connect** line to be **+limit-connect{1-}** (to instruct privoxy that it is OK to forward non-web ports).

> **Note**
>
> Both Gaim and X-Chat *WILL send both your username and your hostname to the IRC server by default*. Both can be configured to send a different username, however. X-Chat's config is right in the server list menu, where as Gaim's is under "Show More Options.." in the preferences for the account. I'm not sure how to solve the hostname problem, short of running **hostname foo** as root on Unix, or editing the source. Hopefully you followed the advice above about not naming your machine after your self or your street address.

Also, you want to make sure your IRC client never responds to DCC file transfers or chats automatically. A DCC connection is a direct connection over the Internet to your IRC client. Naturally, this will give away your IP address. Gaim typically will ask you if you want to accept the connection, where as X-Chat users will need to enter **/ignore * DCC**.

Choosing an IRC Network

Unfortunately, a few of the major IRC networks, have been abused by script kiddies to the point where they had to ban Tor. Brain dead solution if you ask me (what's wrong with an email-confirmed NickServ?), but when you're dealing with monkeys fighting monkeys, what can you expect but that they hurl shit at each other. Unfortunately, legitimate folks in need of anonymous communication get caught in the crossfire. However, if you need to get on to either of these networks, you can try to use a regular open proxy, or for stronger anonymity you can try to bounce off one of these proxies after Tor, and/or bounce off a UNIX shell.

If you use X-Chat, you can conveniently choose a network from the "Server List" menu. Otherwise, pick a network from that site and go to its website for a server list.

You can also pick one of the hidden service servers listed on the Hidden Wiki. I'm a fan of the OFTC site, because it also has a public interface so that non-tor users can still talk with you.

Creating Web Content

On the Public Web

If being on the "real web" is your goal, there is at least one hosting provider that will accept Bitcoin or Liberty Reserve, and will register your domain and provide anonymous hosting for you. Alternatively, by using a combination of money orders and other physical interaction techniques, it may be possible to achieve the same end from cheaper hosting providers who do not explicitly offer anonymous service.

However, if you are hosting content that may anger a large US corporation or otherwise could be construed to violate US law (even if you believe you are doing something completely legal, the First Amendment is no protection against lawsuits from a company with far more dollars than sense), you are best served by finding hosting in another country.

A search for offshore hosting yields several hits. Incidentally, you should verify that any offshore hosting provider you go with is actually offshore, especially if you are seeking offshore hosting to escape censorship (some companies provide offshore hosting, but are in fact

incorporated in the US, making them subject to DMCA takedowns and the like). The best way to verify this is to query their domain name via whois and their IP address via ARIN.

Yet another option you might consider using to add an extra layer of obfuscation is to get an account with one of the aforementioned OpenVPN providers. You can then host your website at any physical location you choose, independent of the server IP address. Note that it won't take much work for someone who can monitor traffic at the server to determine your source IP, so this technique should only be used in combination with an anonymous co-location account above if true anonymity is required. And then at this point all you buy yourself is a little advance warning when the VPN service shuts you down before the Colo provider does.

Another interesting (but ultimately not very effective) option is to take a page from the spammer's book, and combine a VPN solution with some ARP wizardry. This only buys you minimal anonymity though. It is probably only a matter of hours before jack disconnection trial and error at the ISP reveals the real destination of the packets. But an interesting technique nonetheless.

If you are dead-set on using OpenVPN, one possibility is to connect to your OpenVPN provider over Tor or HTTP proxy, allowing you to host content as if your IP was at the OpenVPN provider's network, yet your server is at some other anonymous location concealed by the Tor network. This process is only slightly more complicated than setting up OpenVPN by itself, although the resulting connection will most likely be neither speedy nor stable.

Over Tor

To serve content on the Tor network, you have to set up your own web server and configure a tor hidden service. When you start tor, it should print where it expects to find a torrc and that one doesn't exist. Copy torrc.sample to this location, and uncomment the **HiddenServiceDir** and **HiddenServicePort** options. HiddenServiceDir should be an empty directory. The first time you start tor after this modification has been made, two files will be created in this directory. The hostname file will contain your hidden service name.

Once your service is configured, you will need to set up Apache. If you run a Linux box, doing this should be pretty straight forward. There is some Windows documentation for it as well.

After your hidden service is configured, it should be available to anyone who uses Tor via the .onion hostname, and also via the proxy gateway at serifos. This means if you link to your hidden service from the hidden wiki, it should be searchable via google.

When running a hidden service, you have two major threats: Intersection attacks, and predecessor attacks. Intersection attacks narrow in on your identity by using ("intersecting") various characteristics deduced from your uptime, update frequency, web server version information, etc. The most dangerous type of intersection attack applies if you run a Tor node on the same machine as your hidden

service. In this case, it is possible for an attacker to record uptime/reachability of all Tor nodes in the database, and find the node that most closely matches the reachability history of your hidden service. If you want to run a Tor node, it is best not to run it on the same machine as your hidden service.

Predecessor attacks are most applicable if you are not running a Tor node. Essentially the adversary will make repeated requests to your hidden service in some detectable timing pattern, and attempt to correlate this with how often one of their malicious Tor nodes is used to create a new circuit and sends this timing signature of encrypted cells. Given the number of users on the Tor network, this attack is probably very difficult to mount effectively (though it supposedly has been done). It can be mitigated by choosing trustworthy entry nodes from the Tor Node Status page for use in an **EntryNodes nick1,nick2,nick3** directive in your torrc (also, remember to set **StrictEntryNodes 1**). The torrc option EntryGuards can be used to simulate this effect, but it is not as reliable as explicitly picking trusted nodes. The two can be used in combination, however.

You might also want to take the extra step to only allow SSL connections to your service. This may be excessively paranoid, since there is end-to-end encryption for hidden services, but nonetheless it may be desired to provide another layer of authentication of the hidden service itself. To do this, you will need to install and configure mod_ssl, and generate a self-signed certificate with the Common Name being the same as your .onion hostname.

Over I2P

I2P calls a hidden service an "eepsite" and is very similar to a Tor hidden service, except they have a web console that allows you to create the public key. This HOWTO walks you through using that web interface, and this forum post then gives you the relevant vhost section to add to your Apache http.conf.

Once you are set up, you should post your key to the i2p forum (use either Tor or I2P).

Note that I2P has its own set of possible vulnerabilities. I personally regard I2P eepsites as less safe than Tor hidden services, at least currently. Their network has much fewer users, and its distributed node directory makes it vulnerable to partitioning attacks that can gradually narrow in on eepsite hosters. I2P also does not support the ability to choose your trusted peers (ie the **EntryNodes** option in Tor) or to not be listed in the node directory. Supposedly these features are planned eventually, but I wouldn't recommend hosting extremely sensitive material on I2P until they are implemented.

Apache Tidbits

There are a few Apache config file tags you should make sure are set to reasonable values. These include **ServerAdmin**, **ServerTokens Prod**, **ServerSignature Off**, and for Tor/I2P, make sure all your virtualhosts bind to localhost. Also, you will probably want to disable modification time reporting if you allow access to directories without index.html files, since this can be used to narrow in on your location. **IndexOptions IgnoreClient SuppressLastModified** will do this. Note that HTTP/1.1 **HEAD** requests will still reveal modification time, but these times are given in GMT.

It should also be noted that for both I2P and Tor, any vulnerabilities in your web server/web applications are direct threats to your anonymity. All an attacker needs is a way to execute **ifconfig** through your cgi scripts, and your anonymity is gone. So take great care to secure your website if you are going the I2P or Tor route.

Anonymous Blogging

For those who don't really want a full website, but instead a forum to post information, setting up a blog account is a good alternative. Typically all that is involved is creating an email address, and then creating an account at a blog provider such as Blogger or LiveJournal.

Blogger seems to not display the signup link unless you have javascript enabled, which is annoying if you use NoScript. However, the rest of posting functionality, etc seems to be just fine without Javascript, which is comforting, especially when reading comments to your entries. Also, they don't require the email address to be valid, which is a plus.

Invisiblog is also a potential place to host your blog as well, though it is considerably more involved than Blogger.com or LiveJournal. The anonymity comes from the fact that you post via the MixMaster anonymous remailer network. However, in my opinion, this service has three pretty sizable problems:

1. You have to use Tor anyway
   As they mention numerous times in their docs and FAQ, you should not make a habit of visiting your blog to check if posts arrive, since your IP would thus show up more often than anyone else's, especially for new posts/a new blog. In my opinion, you should not access your blog unprotected *at all*, because any hits without a refurl indicate that that visitor is either a regular or a maintainer.
2. They do not allow web-based remailer gateways
   This is a major stumbling block. Mixmaster is really difficult to set up for your average user, at least compared to Tor. I'm not quite sure why they ban web based remailers. Perhaps they are not aware that people can access them through anonymous means.

3. The URLs are cryptic and hard to communicate
   I realize it probably was easier to just take the GPG key ID as a unique ID than to allow users to try to pick a unique title and handle rejection of duplicates, but this is a barrier to communicating the URL effectively.

I should also mention that the EFF has published some information about anonymous blogging. All of the anonymity stuff is covered in this HOWTO, of course, but they also give some legal information that may be of use to you.

> **Note**
>
> Blogging in general is seeing increased mixing with social network software. As such, you need to be especially careful about your Social Network and your audience of your blog. If most of the people who end up viewing and posting to your "anonymous" blog are your friends, family and coworkers, you don't really have any anonymity.

Document Metadata

Many document formats conveniently embed personally identifying attributes called metadata,and this data is analyzed by companies like Google and Facebook when you upload them. This can be problematic to whistle blowers who need to produce/deliver incriminating memos and photos to journalists, and also to academic researchers who wish to electronically publish their work anonymously. For image files, search for "delete Exif data" for applications that can do this for you.

Microsoft Office

Do Not Use MS Products

Microsoft Office embeds your name, machine name, initials, company name, and revision information in documents that you create.

According to Microsoft's knowledge base article on the Metadata, the best way to remove all personal metadata from a document is to go to **Tools | Options | Security Tab | "Remove personal information from this file on save"**. Be warned that this does NOT remove hidden text and comment text that may have been added, but those tasks are also covered in that article.

Microsoft also provides the Remove Hidden Data Tool that apparently accomplishes those same functions but from outside of Microsoft Office.

This NSA Guide to sanitizing documents might also be of some interest, but I think the Microsoft KB articles cover the info better and in

more depth.

LibreOffice/OpenOffice

By default, users of LibreOffice/OpenOffice are not safe either. Both of these programs will save personal information in XML markup at the top of documents. It can be removed by going to **File | Properties** and unchecking **"Apply User Data"**, and also clicking on **"Delete"**. Unfortunately it does not remove creation and modification times. It's not clear how to do this without editing the file raw in a plain text editor such as notepad.

Document DRM

Document DRM can come in all shapes and sizes, mostly with the intent to restrict who can view a document and how many times they can view or print it (in some cases even keeping track of everyone who has handled a document). For whistleblowers who need to circumvent DRM to distribute a document, the most universal approach is to use the "Print Screen" key to take a screenshot of your desktop with each page of the document and paste each screenshot into Windows Paint and save it. Some DRM software will attempt to prevent this behavior. This can be circumvented by installing the 30 day trial of the product VMWare Workstation and installing a copy of Windows and the DRM reader onto it. You can then happily take screenshots using VMWare's "Capture Screen" or even the "Capture Movie" feature, and the DRM software will be none the wiser. With a little image cropping, you can produce a series of images that can be distributed or printed freely.

The VMWare approach may be problematic for DRM that relies on a TPM chip. The current versions of VMWare neither emulate nor provide pass-through access to the TPM. However, TPM-based DRM systems are still in the prototype stage, and since it is possible to emulate and virtualize a TPM, it should only be a matter of time before some form of support is available in VMWare.

Depending on the DRM software itself, cracks may also be available to make this process much more expedient. Casual searching doesn't turn up much, most likely due the relative novelty (and public scarcity) of document-oriented DRM. Note that when doing your own google searching for this type of material, be sure to check the bottom of the page for notices of DMCA 512 takedowns censoring search results. It is usually possible to recover URLs from chillingeffects' C&D postings. That, or use a google interface from another country such as Germany.

Image Metadata

Metadata automatically recorded by digital cameras and photo editing utilities may also be problematic for anonymity. There are three

main formats for image metadata: EXIF, IPTC, and XMP. Each format has several fields that should be removed from any image produced by a photographer or depicting a subject who requires anonymity. Fields such as camera model and serial numbers, owner names, locations, date, time and timezone information are all directly detrimental to anonymity. In fact, there is even a metadata spec for encoding GPS data in images. Camera equipped cell phones with GPS units installed for E911 purposes could conceivably add GPS tags automatically to pictures.

The WikiMedia Commons contains a page with information on programs capable of editing this data for each OS. My preferred method is to use the perl program ExifTool, which can strip all metadata from an image with a single command: **exiftool -All= image.jpg**. MacOS and Linux users should be able to download and run the exiftool program without any fuss*(for Ubuntu install package libimage-exiftool-perl)*. Windows users will have to install ActivePerl and run **perl exiftool -All= image.jpg** instead. Running exiftool without the -All= switch will display existing metadata. The -U switch will show raw tags that the tool does not yet fully understand. As far as I can tell, the -All= switch is in fact able remove tags that the tool does not fully understand.

Another easy way to remove all metadata from an image it to open it in MS Paint, copy it, and paste it into another copy of paint. The Windows clipboard only copies the raw pixels and leaves the metadata behind.

Bit Torrents/P2P apps

Over Tor

A few different Bit Torrent programs are beginning to support routing tracker (and data) traffic over Tor. This HOWTO describes doing so using Azureus. However, *PLEASE DO NOT ROUTE DATA TRAFFIC OVER TOR*. The Tor network is still small, and cannot support the additional strain. At some point in the future, Tor may implement some form of load balancing to support bulk traffic, but this has not happened yet. So please be polite and only send tracker traffic over the Tor net. FIXME: That HOWTO is pretty bad. Find/write a better one.

This means that you should only really follow the instructions in Section 4.1. The instructions are a little confusing, but basically you want to edit your preferences to tell Azureus that your tracker server's external IP is your .onion address from Tor. So long as the port here matches the public port in your tor hidden service config, you should then be able to give people the .onion address. If they set up Azureus to use tor to proxy tracker data (*NOT TORRENT DATA*), they should be able to connect to your torrent.

Be aware, however, that it is still possible for the MPAA to connect to your torrent through tor, and then watch the IP addresses of

where data is coming from. However, it remains to be seen if they will actually put the effort forth to do this for every torrent everywhere.

Over I2P

Unlike Tor, the I2P network is designed to handle client bittorrent traffic running over it, and thus providing maximum anonymity (at the expense of roughly 1/3 the bandwidth efficiency).

Once you're connected to I2P, you can use search.i2p and orion.i2p to track down torrents. Be sure to contribute and create your own torrents when possible.

Incidentally, an anonymous I2P hacker has altered the Java-based gnutella filesharing program Phex in order to make it run entirely over I2P. I2PHex can be found on the I2P forum.

FIXME: At some point make a new section dedicated to comparing/contrasting WASTE, MUTE, I2PHex, DC++/Tor, GNUNet (which has the unbelievably idiotic property of case-sensitive searches) and other anonymous filesharing nets. Everything but I2PHex is broken right now though, so perhaps this is all that needs to be written. I2PHex actually works pretty well.

Guerrilla Data Exchange

If P2P doesn't provide a targeted enough distribution for you yet you do not wish to set up a full scale website, it is possible to exchange large files via data exchange services.

So far, the best services I've found are badongo (1GB limit), verzend.be (1GB limit), and megashares (1.5GB limit). None of those require either login or javascript. Oxyshare (700M limit) also will provide an ftp account with free registration.

Of course, all these services should only be accessed through Tor, and you should not trust them to keep your data confidential. If you wish to control distribution of your particular item, encrypting it symmetrically with GPG is your best bet. gpg -c will do symmetric password-based encryption from the command line, and GUI versions are also available. I would not rely on weaker encryption such as zipfile encryption, since it has been repeatedly broken in the past. However, a new, open zip format called 7-zip supports AES-256 encryption, and is probably more widely installed than GPG.

For video content, both Google Video and YouTube are options. But again, use tor. The legal climate of the US is such that neither of

these services are whistleblower-safe. Again, in the case of Google Video, you need to be especially careful about cookies (and subsequent correlation of search engine usage with your Video account).

The Vector of Information

When publishing information anonymously, the biggest threat you need to watch for is the vector of information. Every piece of information has a source and thus a path. The more people know about the source, the more likely your physical identity to be discovered as being a part of this path.

For example, if you are one of a few people who should have had access to a given piece of data, then attempting to post it anonymously may be dangerous. Depending on the nature of the data, you may wish to either hold on to it while things 'cool down' (and other people might have a chance to come across it and also be implicated), or you may wish you disseminate it as rapidly as possible. If it is material of a nature where the source will seek legal action (or fire you), you may be better served by being patient to allow solid and circumstantial evidence to disintegrate, and/or covert enough to make proving anything impossible. However, if it is material that the source may be willing to kill for in order to keep secret, you should aim for as rapid and wide a distribution as possible, so as to take the heat off of you as soon as possible.

Oftentimes many pieces of data or many publications can combine to form a more detailed profile of you than they could on their own. Consider the subset of people that would be able to publish information about corruption at your work, information about the networks at your former school, and also info on how to hack a regional wireless provider that has coffee shops near your home. Taken alone, any one of these topics could provide a comfortable cushion to keep you relatively safe from direct suspicion and scrutiny, but taken together, it is easy to see that even any two of them could point the finger directly at you. For this reason, if you are publishing a mix of topics, you may wish to serve each as a separate Tor hidden service or Usenet identity.

Note that it is also possible to use machine learning and artificial intelligence techniques to determine if the same author has written two different documents. This means that *if* your life depends upon your anonymity, *and* it is likely that big government and/or very wealthy corporations will do anything to try to track you down over the long run, you are advised to attempt to alter your vocabulary and sentence structure (and possibly spelling) if you publish both anonymously and publicly. However, for most situations this level of paranoia is completely uncalled for. For example, it almost certainly won't hold up in court unless supporting evidence is provided.

Of course, all of this paranoia about AI tracking you down is completely irrelevant if you do something stupid like post your Tor service URL to a mailinglist under your physical name, post an article on the public web or mailinglist and then also post it on your

"anonymous" web site, or post a tarball full of sourcecode where all file ownership and CVS info contain your username. Use your head.

One particularly easy mistake to make that I have come close to making once or twice myself is to be discussing a topic on IRC, Usenet, etc, and then turn around and discuss it as another identity on another forum or mailinglist, to a degree of detail that it is clear to someone who is also on both forums that the two identities are the same. Think twice, post once. ;)

The Social Network

The Social Network is The Man's favorite method of tracking you down. Everyone likes to impress their friends, but if you do so at the cost of revealing your pseudonym to them, you put yourself at risk. The Man (and more dangerously his paid informants) like to hang out on IRC networks and be regaled with tales of danger and intrigue. Be wary of people who seem over eager to hear about your exploits, and do NOT fall into the trap of feeling you need to "prove" yourself to anyone who challenges your credibility or skill. While it's fucking badass you hacked Paris Hilton's cellphone and the very Agents who were chasing you, you're gonna have to learn to be an unknown hero and not brag about it. It sucks, but it's better than jail. Learn to practice Zen and/or dose on some Ego destroying drugs, but keep your mouth shut. Try to take silent comfort in the fact that The Man had it coming. If you really must brag, create a brand-new nym and post to Usenet or something. Maybe drop some "accidental" clues that might lead an investigation down a blind alley while you're at it. But do be careful.

The social network can also be used in more subtle ways, such as the fact that friends/relatives will tend to visit web content you publish without a referrer URL, since they most likely will receive the link directly from you. If they do not use Tor, their IP addresses can be harvested in this way, creating a trail that begins to single you out.

Note that your pseudonym can also be revealed via Vector of Information. For example, if all you do is rant about conspiracy theory with your friends and coworkers, if you then go and publish information about these conspiracy theories anonymously on the web, if any of your coworkers happen to stumble upon your anonymous site, they are likely to be able to determine it is you. More seriously, any other information you publish on that site will also be attributable to you. For this reason, you may wish to use a variety of online personalities and publishing points. Doing this may be overkill, however, especially if you trust those who would recognize your material not to divulge your true identity to others. If jail time is involved (and sadly, it almost always is these days), trust no one.

However, at the same time, note that a properly controlled social network can be an enormous benefit to your ability to conduct certain types of transactions. Especially if the members of the network never learn who you actually are or where you reside. Having a publicly

accessible social network is unbelievably stupid, however. The key is you must develop some way to control access. This control is what keeps organized crime in existence, and it has the notable disadvantage of existing in the physical world.

Physical Interaction

Anonymous interaction with the physical world is the holy grail. If you can fully interact with the real world through the Internet anonymously, you practically cease to exist as far as the Matrix is concerned. Unfortunately, doing this effectively typically requires capital on the order of at least $1000 USD. Not out of reach of business owners, but your clients may have some difficulty justifying the expense. However, some low cost alternatives do exist and will be provided. FIXME: I do not have the resources to investigate many of these options, particularly the expensive ones. If you do, please don't hesitate to [mailto:aceevader-a-t-mailvault.com contact me] with results.

As a word of caution, any of these techniques that require the use of a local brick and mortar store should *not* be carried out near where you live, lest someone recognize you. Go to an adjacent town/suburb and work from there. Yahoo Yellow Pages is your friend (of course, its cookies are NOT your friend).

Using Anonymous Money

In the physical world, anonymous cash is a redundant term. Cash is anonymous. Unfortunately on the Internet, money is typically tied to an identity. However, some services do exist to allow you to bend or break this rule.

Money Orders

Money orders are available from the post office or Western Union, and do not typically require any form of ID for amounts less than $3000.00 USD. Some online merchants and most offshore banks will accept money orders. Unfortunately, money orders are typically the most frequent choice of scammers. Many western unions will cash money orders without ID.

Pre-Paid Debit Cards

Pre-paid debit cards are available over the Internet from all of the major credit card companies, marketing to individuals with poor credit. Note that all of them seem to require some proof of identity and valid mailing address (and will not ship to PO boxes). What can be done about this however is opening a private mail box and getting the secondary card sent to the name that this was opened in. Instead of using POB, PMB, or #222 etc. instead use "Number 222" by spelling out the word it does not get picked up by the filters on most prepaid cards at this time.

However, in some areas of the USA these cards are actually available for sale in convenience stores, grocery stores, malls, Walgreen's, Radio Shacks, etc. You pay cash, and you get what is essentially a debit card. Non-reloadable cards are usually available over the counter. Reloadable ones typically must be mailed to you. Neither type requires ID. Major vendors include the major credit card companies themselves, along with GreenDot and Simon Malls. The latter two have store/merchant locators to help you find a store that carries their products. Additionally, most Western Union locations offer named pre-paid debit cards. They do require ID and a mailing address, so you will need to use one of the techniques described below. In Europe, 3V Vouchers are also becoming available. ID requirements are unknown. It seems impossible to get them online without some form of identification chain, unfortunately. Perhaps walk-in to stores is different?

If you cannot get anything reasonable locally, various independent providers will offer you prepaid "virtual" credit cards (sometimes called "Gift Cards") at various rates. Unfortunately, PayPal (and possibly some other online merchants?) can be a bit picky about accepting these cards. The best place to find information about which card providers are trustworthy and widely accepted are various online forums, and of course searching for the company name and "fraud", "scam", "sting", or "paypal". In particular, I've seen good things about Money Around The World, whose cards supposedly will work with Paypal IF you ask them for that feature ahead of time. They do not require ID. Similarly, SloGold advertises debit cards that can be used with paypal and even can be the recipient of wire transfers and direct deposits. XLCard.com has similar features, but their Paypal status is unknown. I've also heard reports that prepaid "Gift Cards" are being sold over the counter at stores such as Safeway, Sunoco, Walgreen's, and Rite Aid in amounts up to $500.

Using the anonymous email address you created in an earlier section, you can then bind these card to a Paypal or StormPay account, and conduct small purchases on ebay or anywhere else anonymously (modulo shipping). Be careful to differentiate between no-name debit cards an anonymous credit cards. Paypal and online merchants may not accept cards with no name on them. No-name debit cards are typically only good at ATMs.

I'm offering anonymous virtual visa's that work for all online purchases and phone purchases in any country. No SSN or national ID required to use these cards. I only accept bitcoin. It is 10BTC for a $100 card and 25BTC for a $250 card. Additionally, at the beginning, it may take up to two weeks for me to deliver. Contact Ploni at chat.freenode.irc channel #bitcoin-otc or

plonialmoni@riseup.net

Offshore Banking (Theoretical: Feedback Requested)

Several companies on the web allow you to create anonymous credit cards and bank accounts funded via wire transfer, gold, or money order. Obviously a fully functional offshore bank account would be more useful than just a debit card. Unfortunately, many of these require a shipping address, a photocopy of some form of ID (for their records only, or so they claim), and/or are prohibitively expensive. A few that looked most promising were E-Fidex, Unitrust Capital, Offshore, Etc, and Cheung & Siu. Unitrust and C&S offer several services including "virtual office space" (with mailing address) as well as the ability to incorporate overseas. They do not mention an ID requirement, though they do mention additional fees if they file the ID documentation for you. Supposedly the Patriot Act has somehow made it an international requirement to produce some form of ID to open a bank account. It's not clear exactly how these institutions skirt this requirement, if they do at all.

E-Gold (Theoretical: Feedback Requested)

There are a couple companies that will keep track of gold electronically, and transfer it to and from certain parties. The most popular (and presumably the most trusted to have actual gold on hand) is E-Gold.com.

E-Gold is purchased from one of their escrow agents who actually buy and sell the gold from E-gold's holdings. In particular, Goldage.net will accept money orders for e-gold and also provides anonymous credit cards. Alternatively, you may face less regulation purchasing e-gold in one of the ad-hoc e-currency exchange forums or one of their corresponding topsites such as MoneyDuck or PaysGold. Your chances of being scammed do go up when you do this, however, so be careful.

Many of the offshore banking institutions also accept transfer to and from E-gold. E-gold has been used (presumably successfully) by the Source Code Club to conduct sales of corporate source to those wishing to evaluate it. E-Gold issued a statement that it will do its best to track down these guys, but so far the Source Code Club seems to remain in operation.

Anonymous Snail Mail

Many people who accept E-gold and many of the Offshore banking companies suggest mailing items to a local shipping agency, post office, or mail box provider with instructions for "Hold for Pickup". This way, it is possible for a package to be delivered to their location in the name of a fictitious company for some holding fee. You can tell them a salesman traveling through town will be by to pick up the

package. A variation on this technique is to use General Delivery in combination with a made up business card and legitimate ID to pick up mail at the Post Office itself. Since the only record of delivery will be to the business name (and not the ID shown), it is supposedly OK to use your real ID.

A far less cumbersome option is to rent a mailbox at a privately owned mailbox rental company (Commercial Mail Receiving Agency - CMRA). Unfortunately, most of these are bound by postal fiat that requires them to enforce ID requirements that may be verified at the post office. Since it has been reported that the Post Office sells consumer's addresses to marketing agencies, this is not very comforting.

The form you have to fill out is Form 1583 and is universal among all CMRAs. It requires two forms of ID, one of them photo. The Privacy Statement is riddled with exceptions to allow the agent to provide information to "contractors", "financial entities", USPS auditors (who appear to be under no privacy obligation themselves), and for purposes of "identifying addresses... used to deliver mail to other persons". Valid ID includes state ID, armed forces, government, corporate, or university identification cards, passport, alien registration card or certificate of naturalization, current lease, mortgage or Deed of Trust, voter or vehicle registration card, home or vehicle insurance policy. *According to this contract, it is *not* mandatory that a photocopy of this identification be taken, but it must be written down on lines 8a and 8b by the clerk who accepts this application.* If you are providing state ID with personal information on it, you would do well to insist that a photocopy not be made to avoid identity theft.

It is possible to avoid the regulatory hassle involved with CMRAs by instead leasing a "virtual office" from an Office Business Center (OBC). A "virtual office" typically consists of a mailing address, some amount of office time per month, a phone line and answering service, and access to conference rooms. Providers who offer this service are not subject to registration with the post office. Numerous virtual office providers can be found in any major metropolitan area, and rates are usually around $50-150/mo for basic service. I personally find it amusing that so long as you have sufficient money to pay for better service, you don't have to be stamped, branded, and tracked by the USPS, but people who cannot afford these extra services have to be watched with utmost scrutiny.

Your last method for anonymous snail mail is to usurp a "dead" mailbox. This is a mailbox that still has a postal address, but is not being used. Examples include vacant lots, empty office rooms, etc. Empty office rooms and janitorial closets typically will require permission of the building manager, of course. Vacant lots and unused street mailboxes can probably be easily "borrowed". In some cases, setting up a whole new mailbox with a "1/2" or "A" address out in the country is a very nice option as well (but may be noticed by neighbors). A completely new address may be noted less, but the flip side to that is the postal carrier may take issue with this.

Along these lines, at least one book reports success in searching/posting on online bulletin boards/classified ad servers for already registered mailboxes, either postal, UPS or unused office space. There may be many people who purchase mailboxes then simply

move out of the area. The same book also mentions that it may be possible to receive mail at a Salvation Army or YMCA for a donation.

Ideally, the physical location that you ultimately have to go to to pick up your mail should change every 12 months. If your budget and need for anonymity was high enough, one way to increase the length of this window is to attempt an SSH Hopping-like technique by chaining virtual office forwarding systems together to attempt to obfuscate your location by crossing many international boundaries. That is, until a tor-like mixed network for mailing packages arises. I have not tried this out yet, but it would seem like Unitrust Capital has a decent offering, as does ABCN. Another option is to open a New Mexico LLC and then sign up for a Ghost Address. You can also try browsing this directory or the DMOZ/Google Directory entry for more options. Let me know how things work out for you if you decide to go this route.

> **Note**
>
> One last important thing to note about the mail (and physical interaction in general) is to be extremely careful with things you handle, especially if a fingerprint is on file with the local DMV, or if you purchased your printer with a credit card. The EFF maintains an excellent page about printers that encode identifying information in printouts, and how to detect if your unlisted printer is also bugged in this fashion. I have been told that printing to transparency film works even better than the techniques the EFF suggest, as the transparency will make the layered dots visible to the naked eye without the use of a blacklight or microscope.

Anonymous Telephony

Anonymous telephony is a tricky feat to accomplish: seemingly easy to do, but also easy to make mistakes that ruin your anonymity. Basically the goal is to obtain a cell phone that is untraceable to your physical identity. This in and of itself has recently become possible, but there is a steady stream of subtle information leakage from any phone that will eventually point to its owner.

Obtaining an Anonymous Cell Phone

In the US, anonymous cell phones have recently come available in truck stops, discount retail stores (Wal-mart, RadioShack, etc), and at cell carrier outlet stores. The main carriers that offer anonymous pre-paid service are T-Mobile, Cingular, Net10, and the ominously named TracFone. Note that some retail stores will ask you for your name and address, so you should have one ready.

For some reason, pre-paid cell phones are subject to a very bizarre price structure. The same phones offered on the web by the

carriers are typically $100 more when you visit your carrier's local store. While it may be tempting to order these phones directly from the web using an anonymous debit card because of this, you probably are better served by going to a retail store and purchasing with cash, just to keep a distance between your debit card and your phone line (though sometimes this binding is required anyway for other reasons). Walmart, Costco, Radio Shack, etc typically have the phones for web prices or cheaper.

Another detail you should be aware of is that cell phones typically come "locked" to a given carrier, preventing you from switching carriers in the future. When selecting a phone, you probably want to try to obtain a model that is easy to unlock, so that if you need to switch cell phone carriers, you can. Nokia phones are usually easiest to unlock, typically by entering in a "secret key". The Nokia 6010 offered by T-Mobile in particular is readily unlockable, and is available at Walmart. To unlock it, use the DCT4 form, Network: T-Mobile, Gen: v2, Model: 6100 and use the first code. If the first code fails, try the 7th.

Information Leakage

There are a couple of things you need to be aware of when using an anonymous cell phone. If you are not careful, your anonymity can be reduced to zero in a hurry, and you can easily reveal your identity and location with a couple simple mistakes. In particular, here are a few things you should be aware of:

IMEI Numbers

Each phone has a unique, semi-permanent serial number called the IMEI number. These numbers are actively tracked in databases that are becoming international in scope. Note that this number is a property of the phone itself, and does NOT change if you pop out your SIM card to change carriers. As such, changing carriers with the same phone buys you no extra anonymity, and placing a SIM that is easily traceable to you into your anonymous pre-paid phone kills any anonymity you had, potentially even retroactively.

Note that the converse is also true. If you have an old phone previously registered under your name and decide to try to use it with a pre-paid carrier, you have no anonymity.

E911 Service

E911 is a standard set forth by the US FCC that essentially specifies how accurate cell phone carriers have to be when tracking their users under various conditions. Cell phone providers can meet the accuracy requirements however they see fit, and the major carriers have adopted a couple of different technologies.

While there have been some frightening uses of this technology by spyware installed on phones, what is most frightening about E911 is that there is no law that governs location-data privacy. This means *nothing* stops The Man from watching the location movements of any and every cell phone user he feels like. E911 location information is transmitted at all time while the phone is on, and no warrants are needed to obtain this information.

The FCC has mandated that E911 be present on every cell phone sold after Dec 31, 2005. However, several models of phones do allow you to disable the E911 location information. For other models, your only option is to keep the phone turned off with the batteries out.

For phones that are not turned off or have a nonremovable battery, a Faraday cage can be used to block all incoming and outgoing signals. One example is the experimental Offpocket. Alternatively, tinfoil can be used though such an implementation can easily garner unwanted attention and subjectively looks bad.

CALEA and Relevant Surveillance Law

The CALEA is the US law that governs obtaining warrants for wiretap on electronic communications. Much like E911, it merely specifies requirements that industry must follow in granting the federal government access to communications. The problem with this system is twofold. First off, obtaining a wiretap warrant is pretty much a rubber-stamp process with little real oversight; and second the fact that the mass-surveillance infrastructure built to support CALEA is easily subverted to criminal and even rogue-state ends.

FIXME: At this point in time, it is unclear as to whether recent expansions in wiretap law make it easier to obtain a warrant for arbitrary pre-paid customers before their identity has been revealed through other means. It seems as though The Man has to at least have a vague idea that the phone number in question is being used for Unapproved activity, but as the warrant statistics indicate, even this may be at best a symbolic gesture. As such, it is recommended that even after obtaining a prepaid cell phone, you not put full faith in its anonymous and private nature.

The Social Network

Once again the Social Network rears its ugly head. If there is one thing you can be sure of, it's that *EVERY PHONE NUMBER YOU CALL OR THAT CALLS YOU IS LOGGED*, even if you are not currently under surveillance. The call logs are indexed by IMEI, so switching phone numbers and carriers does you no good. This means that it is possible to automatically determine that your anonymous phone and your nonymous phone share many of the same numbers and thus are operated by the same person, or at least two people that know each other. Avoid calling the same people on your anonymous phone as you do on phones that can be traced

back to you, and instruct them not to call you either. The more numbers are shared (either outgoing or incoming), the greater your risk of being uncovered. When a phone starts to be contaminated in this way, toss it and get a new one. People have been caught this way.

Assuming an Identity

Unfortunately for most interactions with the physical world, you typically need at least some form of ID. You basically have five options:

Employ a Homeless Person or Post to Online Classifieds

If you live in an urban area, you might be able to find a reasonably coherent homeless person (or someone willing answer a classified ad posted on a community bulletin board or website) to assist you for a small fee. It turns out that the international nature of craigslist can make it possible to operate in geographical contexts far distinct from your physical location (though Craigslist seems to have decided to block Tor, so you may need to put a special line in your privoxy config to access them anonymously).

This can get sticky, and probably requires a good judge of character to pull off. You should definitely make sure the money you give them for the institution is in the form of a money order written out to the intended recipient, to minimize their chances of running off. You should only pay them for the job after they complete it.

Make sure that it is not possible for them to obtain access to the account or mailbox after they create it. Obviously keep any keys/cards to yourself, and make sure that it is difficult for them to get any replacements immediately. Possibly use two different people for mailbox creation and account creation. Ideally, you should use a service where replacement cards are mailed to a mailbox you control, and not to them. You may wish to bring a friend along, to make it clear that if there's trouble, "more than one person" will be looking for them.

Even after all of this, it is still possible they might flake out, or worse, attempt to blackmail you by threatening to call the authorities. Give them a decent cover story, such as you are trying to hide from an obsessive ex-lover, or have a job where people might seek revenge on you personally (meter maid, tow truck operator, judge, lawyer, etc). Have a story ready about how some friend of yours or someone on the news was harassed because of their job. Even if you believe your reasons for seeking privacy are legally safe, you should limit what you tell your courier about your exact circumstances, since this can weaken your privacy (it's a small world).

Ideally, you should be using them for one-shot deals, like courier service or to set up an overseas account, or to open an account whose card and number will only be given via mail (ie to you, not them). The less information they have about what they are doing, and the less they see of the end result, the better off you are. Don't work local to your home (or theirs). Ideally, you should never see this person again.

Even with all the hassle, unfortunately this is the safest method to use with respect to US law. If you are doing ANYTHING that might attract the attention of or otherwise annoy an FBI agent (which in these troubled times is just about anything), acting by proxy is the only way to go.

Manufacture ID for Yourself

Unfortunately, making a fake governmental ID can bring a lot harsher penalties than is worthwhile to risk, depending on your threat model. US Title 18, 1028 criminalizes any interstate production/use of government issued identification with penalties of up to 15 years in jail. Simply using fake state ID is considered a misdemeanor and is punishable by a maximum of 3 years, though first-offense misdemeanors almost never receive jail time. This means that it is usually simply not worth making state ID for most people, since you will likely have to destroy (or sell) most of your equipment if you don't want to spend time in federal prison for being caught using it.

However, it is possible to obtain a CMRA mailbox with two non-governmental forms of ID (such as an employee ID and a local city/community college ID), so "novelty" ID creation is still an option. As far as I know, presenting "novelty" non-governmental identification is not criminalized. There is slim possibility of charges of mail fraud, but from reading USC 18-63 and the DOJ prosecution policy, it would seem that mail fraud is only applicable if someone has actually been deprived of money/property via the mail. After all, are they going to prosecute every author that publishes under a pseudonym who has ever sent something through the mail? That would be a bit excessive, even for the US government.

If you invest a bit of money (around $200-$500) you should be able to make a variety of ID yourself. There are a couple of text files that describe the process (along with some supplementary material I found on usenet). Alternatively, you can check out this book for a detailed overview of how to create a wide variety of ID.

Also, beware of cheap template collections you might obtain via P2P networks. These almost all suck and are dangerously out of date. The reason for this is that even electronic transfer of ID is criminalized just the same as physical ID in USC 18-1028. It is possible that templates may begin showing up on anonymous P2P networks, but you should focus on cloning local (ideally non-governmental) ID anyways. If you are still dead-set on creating state ID, the 2004 US ID Checking Guide (FIXME: anyone have 2005/2006?) contains information on all the security features present in the IDs of all 50 states, so that if you decide to go the template route, you can verify what you have is current, and you can use it to cross-check to make sure you don't miss anything. Alternatively, local copy shops typically have high quality scanners you can use to save yourself some money. As far as printer, the above book recommends the ALPS MD series, but those are discontinued and prone to breaking (meaning buying a used one is probably a bad idea). You're probably better off using an Epson C82 or 740 (the 840 tends to print too fast and is prone to smudging), which have been reported to

work well on alt.2600.fake-id. Use the Photo-EZ trick mentioned in the above text files for stenciling patterns for UV/metallic inks.

Lastly, it should be noted that some places (especially the offshore banks) require only photocopy/fax of ID, which should be especially easy to spoof. However, some outfits may actually query your driver's license number at the appropriate DMV. If they are operating overseas, they are less likely to have the capability to do this, but in any case, I have not tried it, so attempt at your own risk. My guess is that due to recent events, companies will have less freedom to query these databases, since this just opens the door up for rampant abuse. However, if you do try it out, take proper precautions for ensuring the fax phone line can't be traced back to you (use a local copy shop), and use all the digital precautions we've discussed thus far. This way, at worst you get rejected immediately, and no harm is done.

> **Note**
>
> As mentioned previously, make sure not to purchase your printer with a credit card. The EFF maintains an excellent page about printers that encode identifying information in printouts, and how to detect if your unlisted printer is also bugged in this fashion. I have been told that printing to transparency film works even better than the techniques the EFF suggest for detecting identifying markings, as the transparency will make the layered dots visible to the naked eye without the use of a blacklight or microscope.
>
> If your printer does print identifying dots, you will want to sell it/dispose of it if you created state ID, because the identifying information can be used to prove that you produced your own ID instead of obtaining it elsewhere (the difference between a misdemeanor with typically no jail time and a maximum of 3 years, and a felony with a maximum of 15).

Manufacture ID for Another

The previous technique is not without its weaknesses either. For instance, it is not ideal that a photocopy of your picture and ID is on file with an account. While it is presently not implemented (at least not as far as I know), assume that within the decade it will be possible to use face recognition to quickly match drivers license records to other pictures of individuals. Even this aside, there still is the slim but real chance of human recognition, or recognition after an investigation begins.

For protection against this, you can take the extra step of creating the ID for the homeless person/courier and keeping it to yourself until you need their services. That solves both the recognition risk, and the risk of the them trying to use the ID to obtain access to your account/mailbox. Unfortunately, it does not solve the blackmail problem. For this reason I *strongly recommend against making state ID to give to another*, as this is instant leverage for blackmail, independent of the nature of the use.

However, since "novelty" non-governmental ID is not criminal in and of itself, blackmail is much less possible. There are a myriad of reasons someone might desire anonymous banking and mail delivery without them necessarily being guilty of any crime (again, jealous lover, dangerous job, etc). Invent one and tell it to your courier before they agree to help you out. Preferably not the real reason, since this can be used to determine who you are.

Attempt to Build a Government Recognized Identity

This is an extremely complicated process that is easy to screw up and can land you in jail. The process typically starts with some form of fake ID created as discussed above, and uses that to obtain legit secondary documents from the government, which can then be used to get you into the DMV and other databases with legit ID. If you are seriously considering this, you should have a look at the books in the print media resource section.

Also be aware that the long-term danger of face/biometric recognition technology still applies to this method, especially if your face exists as two different people in the DMV database. Many states already implement this type of technology using electronic fingerprints to verify no two licenses have the same thumbprint. In this case, the clever folks at the Chaos Computer Club have a mechanism through which you can borrow the fingerprints of another. Another hack that I've been told works well is to use a red felt-tip marker to obscure identifying marks on your finger before it is scanned. The red ink messes up the red laser light from optical scanners.

Identity Theft

Identity theft for the purposes of stealing equity, reputation, and credit from other human beings is morally wrong, and since this is the predominant reason that identity theft occurs, I do not wish to discuss it. This HOWTO is about taking your freedom back from the institutionalized oppression that is the Matrix, not stealing from your fellow man. If this isn't enough to discourage you, note that ID theft is more dangerous than creating an identity since the victim may notice additional accounts on his credit report or elsewhere, and may report them at any time. It is likely that the penalties for ID theft will soon skyrocket in the typical "tough on crime" reactionary fashion as well.

However, that being said, it is becoming apparent that at some point in the future, bureaucratic governmental arrogance and momentum will push for mandatory national database verification of identification. When this regrettable time comes (and in some places it has already arrived) identity theft may become a necessary measure in protecting one's privacy. The upshot is that the ubiquity of these data checkpoints will increase the vulnerability surface of the bureaucracy tremendously, making identity theft considerably easier. In that case, both moral and practical considerations dictate that those using identity theft for privacy should strive

to conduct this activity with as minimal an impact on the host identity as possible. This means of course no monetary theft, and typically avoiding any actions that would alter the credit ratings or otherwise appear on the credit reports of the host (such as the creation or use of credit cards and bank accounts).

Protecting Yourself from Fraud

The most risky aspect of interacting with the Anonymous Economy isn't being caught by The Man, it's being ripped off. The best way to protect yourself from fraud is to always scroogle search for the merchant you intend to do business with and add the terms "fraud", "scam" or "sting". If nothing comes up, try to post to a relevant forum and ask. In the unfortunate event that you are ripped off, do complain loudly and vocally on as many forums as possible. Sometimes informing a merchant that you are about to smear their reputation all over the Internet can 'jog their memory' into remembering to ship your purchase after all, so you might want to contact them first.

Be ware that some merchants will gladly honor smaller purchases only to defraud on larger ones, so unfortunately conducting smaller transactions with a merchant might not guarantee that they are safe for larger transactions. Luckily there is a solution. Escrow services can help you conduct larger purchases without fear of the merchant defrauding on delivery. Essentially, the way they work is that you pay the service the amount of the purchase price plus a small fee, and take a shipping tracking number from the merchant, and hold your money until the shipping carrier reports received the product, at which point the escrow agent release the funds to the merchant. Unfortunately, some escrow sites themselves are scams. It is probably best to use one of the services listed on this ebay page.

If you are a merchant, remember that when conducting business your reputation is your bond. All you need to do is mishandle a single customer and your sales will plummet as word spreads like digital lightening that you are a fraud. You want to avoid this like the plague, since ruining a rep typically means you have to rebuild your entire anonymous cover.

You should also allow payment via as many options as possible, particularly escrow services. You want to ensure that any level of anonymous client is capable of transacting with you, and you offer redundancy in payment options. Most likely Paypal is not going to take a particularly fond eye of you (and has other problems, as well), so sometimes a backup plan can be helpful. You should also go through the trouble of setting up your own website anonymously, so you don't have to deal with ebay's rules on what items can be sold (though craigslist is one alternative to ebay with minimum restrictions). It is also easier to build up a widely commented upon and easily verifiable reputation if you set up your own website.

For information on the reputability of various online currencies, you might want to consult The Gold Pages Online Currency Journal and

the Global Digital Currency Association.