

# Reclaiming Territory in Cyberspace

[gabe.rocks](http://gabe.rocks)



Humanity has lost cyberspace to corporations, corruption, and algorithmic control. It is of utmost importance that we increase people's general technological understanding. We can all build a human-centric cyberspace instead of one used to dominate and control us.

## Chapters:

1. Hardware Basics
2. Operating Systems
3. World Wide Web
4. Communications
5. Self-hosting
6. Censorship Avoidance
7. Your Website

# Hardware Basics

We are living in interesting times. These days a vast array of tools are aimed at harvesting as much data as possible to build powerful centralized control systems. These systems are then leveraged against people in all kinds of insidious ways. This is not merely self-interested profit-seeking; it has a purpose and that purpose is to wage war on humanity. The Tools of the Technocracy series was intended to demonstrate the vast array of tools being weaponized against the people.

With this in mind, it is challenging to know how to proceed. People may want to simply eliminate all contact with technology, it's hard to blame them. Others, are simply not prepared to make that decision and can fall into a trap of apathy. This series aims to build a solid foundation to reclaim much of your own digital sovereignty. As you do this, you can leverage many useful systems without becoming a prisoner of the technocracy. Those who are locked into the cloud are not only captives of the system, but are actively providing assistance to those who wish to control everyone, and everything.

But how can we turn the tide? What if it is too late to meaningfully change things? While I can not promise total victory, there are many important ways for people to take action and start to liberate themselves from the cloud. I believe the pursuit of liberty is an asymmetric one in the people's favor. While things may look bleak, it is very clear that every act of defiance, every action taken to bring freedom; costs the tyrants exponentially more effort to reverse.

---

## Strategic overview

In cyberspace there 3 major resources: **Storage, Bandwidth, and Computing Power**. Our goal must be to maximize our use of these resources without having them being used against us.

### 1. Storage

For content and information to exist it has to be stored somewhere. Storage varies in capacity, speed, and form-factor.

- Capacity:
  - Hard drives (HDD) offer significantly more storage per \$ than solid-state drives (SSD).
- Speed:
  - Solid-state drives are much faster than hard drives.
  - Optical media is much slower.
- Form-factor:
  - Flash drives are smaller portable storage devices that can be carried around with you.
  - SD Cards are very tiny and have very high storage-to-space ratios.
  - Hard drives and most solid-state drives are significantly bulkier.

These trade-offs represent opportunities in some areas and troubles in others. Depending on your needs you may only need a small amount of storage, in other cases you may need significantly more. In my opinion, here are the optimal roles for each storage device:

## Hard Drives



High capacity

## Solid-State Drives



Fast

## Flash Drives



Portable

## Optical Media



Permanent

### **Solid-state drives:**

Primary storage for Desktop/Laptop devices.

Their faster speed offers a great deal of advantages:

- Your system will boot faster
- Loading files and programs will be faster
- They are significantly quieter than hard drives
- They're also smaller than hard drives

### **Hard drives:**

Additional storage for media, such as videos, music, documents, and images. Due to their cost effectiveness these are fantastic for larger backups.

### **Flash drives & SD Cards:**

Transferring files over the “sneaker-net” by personally delivering them, or through the mail. Additionally, these are very great for “quick-and-dirty” back-ups of small amounts of files.

### **Optical media:**

Aside from being able to be played directly with their respective reader, optical media is fantastic for archiving. Anything you want stored in an untouched format.

---

At minimum you'll want to preserve data you care about. Backups are very important to prevent data loss. At least having some storage helps prevent you from keeping all your data in the cloud. Portable storage is a great way to have access to the same files on multiple devices without having to constantly sync those files over the network. You can turn HDDs and SSDs into portable storage by using their respective enclosure.

## **2. Connectivity**

The “information superhighway” is fueled by the ability for large amounts of data to be transferred large distances. Connectivity varies in **latency, bandwidth, and range**.

- **Latency** is how fast it takes information to arrive.
- **Bandwidth** is how much information can be carried in a given time.
- **Range** is how far you can send information.

**Wired networks** are king in terms of speed, very low latency and very high bandwidth. Attackers need to have access to the wires to intercept your network information.

**Wireless networks** have significantly more range, but have significant speed and security trade-offs.

**LoRa** (Long Range) has incredible range, and is simple to deploy yourself. Unfortunately, it is not suited for significant amounts of data transmission. Bluetooth is infamously insecure, but has a shorter range and conveniently integrates with many devices.

**Mobile Networks** (LTE/5G) are run by large corporations, but they are convenient. Outside of specific circumstances connectivity is generally associated with your identity. **WiFi** is most commonly providing internet

access to devices in specific buildings. It is generally recommended to use your own access point instead of relying on the router provided by your ISP.

While you may have a fantastic connection, others may not. Ensuring that content you produce is accessible to low data connections puts you at a significant advantage. This means you may want to have as much raw text as possible, and smaller sized images & video. If your website is under 512KB consider joining the [512KB club](#).

---

### **3. Computing power**

More computing power exponentially increases the possibilities available with given resources and information. Computing power has to balance heat, energy efficiency, and complexity. Powerful machines use more power and generate more heat. Clusters of powerful machines generate even more heat and require incredible amounts of energy. It is important not to underestimate the computing power available to the technocrats, many feats impossible to consumer grade hardware are still on the table for larger entities. Likewise, it is foolish to dismiss the capabilities of even relatively small amounts of computing power, even the most simple devices can still relay data to more sophisticated ones.

AI is a force multiplier for computing power. It requires a massive amount of energy, computing power and energy to train, but once the model is trained it requires exponentially less to use the model. This is incredible because AI models can be shared, potentially saving people large amounts of computing power, which in turn saves lots of energy. Unfortunately, this also means that those with access to privately trained AI behind close doors have disproportionate advantages in specific areas.

Smart phones provide a significant amount of computing power on the go. Their operating systems do a great deal to restrict people from installing software outside their store. Many of these devices are sold with relatively little storage to incentivize people to move onto the cloud. The biggest challenge with having a much more powerful device is ensuring it isn't being leveraged against you.

Do you necessarily need to hoard the most computing power you possibly can? I don't think this is necessary. In fact I believe the best strategy is to ensure that the computing power is well distributed between people to even the playing-field. One way of achieving this is sharing a used system with someone instead of recycling it. Not everyone stays up to date with the latest hardware and your "obsolete" device may be an upgrade for someone else. Where possible, it's important to liberate hardware from the cloud, instead of simply purchasing devices and creating more e-waste.

## **Assessment**

The cloud as a whole is an atrocious waste in the first place. Crimes against ordinary people aren't prevented by all this passive surveillance, children still go missing in incredible numbers despite all this incredible technology, and none of it has been used to make people happier, empowered, healthier human beings. At best, the cloud is a tax-write-off for enterprises that do not wish to maintain their own systems. This has significant costs to us as we lose any and all privacy from the system.

Even as a group, humanity is at a staggering disadvantage in terms of these resources. Governments of the world, big tech, and other large institutions will have the majority of the storage, bandwidth, and computing power.

While that makes it sound like it's "game over" it's really not. Manufacturing, configuring, and maintaining all of these systems is insanely expensive. Much of the work done by these systems is solely focused on collecting data on people, analyzing it, and then manipulating people...if the people let them. This is why it is very important for these systems to make it as easy as possible for people to keep feeding them data; every single conscious choice to send these systems less data and reduce it's influence turns this very expensive enterprise into a massive waste.

This asymmetry is the root of our goal. We don't need to build a competing digital-financial-complex to win, we simply have to regain enough privacy in our own lives to undermine this system. The goal of these systems isn't to simply hijack our devices and turn them against us; where possible, they want to turn us against ourselves.

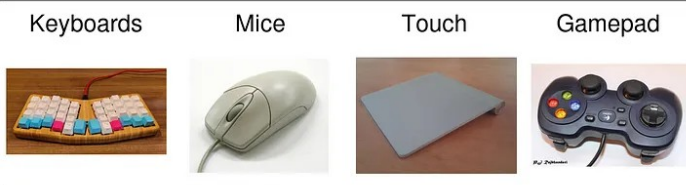
## Devices

The internet as a whole is a giant battlefield in the war for cyberspace. In this respect, every single connected device is a strategic location that may be already entirely compromised by malevolent forces. These are not limited to the technocracy, ordinary cyber-criminals would be very happy to make use of the resources your device has access to. When an attacker takes control of your device, it will inevitably cause you problems. The more capabilities the device has, the more of a danger it is to you. Machines can lie, a compromised device can and will act in subtle, unpredictable ways. Many systems ultimately trust that the device is doing what the device reports it is doing.

What kind of capabilities can devices have?

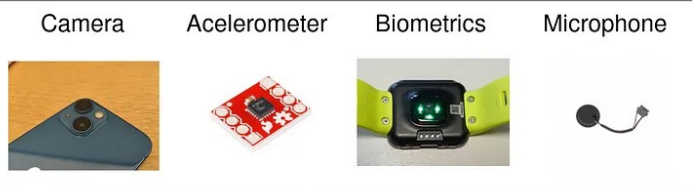
### Input

Devices and components that you more-or-less chose to provide actions for the device to interpret



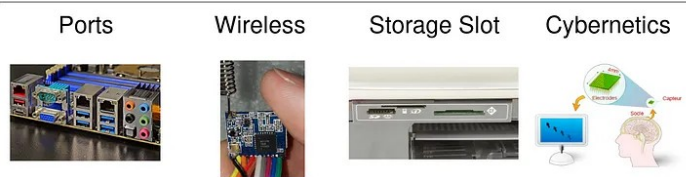
### Sensors

Devices and components that can passively collect information without interaction



### Connectivity

Devices and components that expand the capabilities of the device



### Output

Devices and components that can send information back to you



The main computing elements in machines are the CPU (Central Processing Unit) and the GPU (Graphics Processing Unit) also known as the video card. The CPU is responsible for running the majority of the tasks the system runs, with the GPU being responsible for drawing the display(s).

For CPUs there are specific instruction sets used by programs for that architecture. This is important because not all software will be available to each architecture.

## Aarch64



Single-board computers  
& Mobile phones

## X86\_64



PCs / Laptops

When the machine turns on the boot-loader looks for any Operating Systems that are stored on the system, which are then loaded. Once the OS is loaded, the OS prepares any systems and services that are needed to run. Unfortunately these devices can and do have back-doors in them, most infamously there is the [IntelME](#) which is essentially a whole new system running underneath your system. There are mitigation (discussed below) but it's important to have the long-view when understanding what we are up against.

## The immense cost of anti-human devices

### Planned obsolescence

A major factor in the increasing amount of e-waste is manufacturers dropping support for devices. When a device no longer receives updates it is not only a significant security risk; it also can arbitrarily limit what software the device can run. This robs people of the time and resources used to acquire and maintain the device in the first place.

### Defective by design

In addition to planned obsolescence, many devices are made difficult to repair. Often by Manufacturers not disclosing vital information required to maintain and fix them. By not being able to repair devices, you often can't update or even modify them to your needs.

This increasingly applies to more than just computers, but appliances, vehicles and tools.

# Mitigation measures

## Dedicated devices

When it comes to hardware “less is more” in so many ways. The only way for someone to access photos on a non-networked digital camera is for them to have the device. On the other hand, “smart” devices, with many features and little transparency will be passively sending all kinds of data without the people nearby being aware. An “air-gapped” PC (a computer without network access) is a lot harder to hack into, usually requiring physical access.

- A webcam you can unplug is one you can easily prevent from recording you
- A hardware authenticator can keep encryption keys safe
- Having your backup on storage that isn’t plugged-in can protect you from ransomware
- Using a power bank to charge your mobile device can prevent them from being hacked by charging ports

## Faraday bags

Not every device can be single-purpose unfortunately. For laptops, smartphones, and tablets you may want to consider using a faraday bag that can block out any network signal.

## Hardware switches

Some devices provide hardware switches to turn off specific features, some laptops have it for WiFi. Note that this may be more of a hotkey or “suggestion” to the operating system rather than actually cutting off the power. Make sure you know that the switch actually cuts power to the device you intend to turn off.

## Right to Repair

[Right to repair](#) is an important movement in restoring our rights over our devices. Keeping details about these devices secret does little to keep things secure and is simply a subsidy to platforms with the worst practices. Total ownership over your devices is one of the most important steps in ensuring the creators don’t end up owning you.

## Free and Open Source Hardware (FOSH)

Instead of fighting existing manufacturers, it can often be easier to support the creation of new devices with open schematics and great documentation. This is another area where a little effort goes a long way. As the tools to build devices become more accessible more and more useful projects will come online.

[https://en.wikipedia.org/wiki/Open-source\\_hardware](https://en.wikipedia.org/wiki/Open-source_hardware)

## Libreboot

Libreboot is a fantastic project for creating Free and Open Source (FOSS) firmware for computers. This is very important in removing the Intel ME from intel devices.

<https://libreboot.org/>



# Operating System Fundamentals

After you've taken careful consideration of what devices you want in your life, it's important to take control of the systems you use. Just as not all devices are equal, different systems have massive differences. Fundamentally the question ends up being one of trust. Do you trust your data on a Windows machine designed by Microsoft? Do you trust your data on smartphones with Google or Apple? One of the most powerful tools these systems have is to convince you to put your data onto the cloud by making it easier and more convenient. They don't need to proactively steal information from you if they can simply encourage you to do it yourself.

Our primary advantage against the technocracy is that we can almost always do more with less. By taking control of your systems, you can make significant steps to enhancing your digital privacy, security, and freedom. Every unit of data you deprive from the cloud incrementally makes a waste out of the immense resources spent on running it. This means that it is critical to prioritize keeping your data and data processing on your own systems. Thankfully this doesn't mean doing it entirely alone. There are many ways to collaborate and pool resources without sacrificing ownership, autonomy, or privacy.

First, we need to understand our systems better. By understanding how our systems work, we can truly reclaim our own processing and bring our data home.

## Abstraction

Every time you launch a program or open an app your device has to use the Operating System (OS) to make important preparations. Application files are loaded from storage, and all kinds of system resources and devices are accessed. The important feature of the OS is abstraction; simplifying complex systems into easy-to-use interfaces for applications. Because of this, application developers don't need worry about what kind of device your system has, they only need to know what features it has. For example, your browser can easily use whatever camera & microphone you have, rather than only supporting specific ones.

## Computing power abstraction

The heart of your computing power is the CPU, it is the engine that provides the raw power for computations. Almost all modern CPUs have multiple cores which are units that allow more processing to be done synchronously. The OS translates physical cores (and extra functionality) into virtual threads which run tasks. The important part to understand is that when processing large amounts of data there are often ways to do it faster with programs that make use of more threads.

Memory is similar to storage, but it's the warehouse the CPU uses to organize data. As programs bring in larger and larger units of memory it is up to the OS to eventually free that memory when it is no longer in use when the program stops using it. Operations that use memory are significantly faster than ones working on storage. These trade-offs are things software developers should keep in mind when writing software, it can very often be the difference between a powerful or unusable tool.

Simply being aware of the limits of your system (such as how much memory it has, and how many cores/threads the CPU has) goes a long way in helping understand how to optimize certain tasks. When processing large amounts of data, you may prefer to use a program or tool that makes use of more threads. You can avoid crashes by not attempting things that use up all of your memory.

The critical part of taking control of your own computing is ensuring that you're doing your processing on your machine. Running your own programs rather than using software-as-a-service (SaaS). By doing this you are not only protecting your data, but also your processes. With total control of your environment you are less vulnerable to outside failures.

## Storage abstraction

The most important aspect of having control of your own data is making sure it is only on storage you control, and ideally have physical possession of. For each storage device you have the ability to divide it up into partitions. These are useful in situations where you need different filesystems on the same device. In many cases, the bootloader will be installed to a small system partition and the rest will use the filesystem your OS prefers. For example, your Windows system likely already has a system partition with the bootloader installed, the main partition with all your files, and possibly a recovery partition.

Devices need to have a partition table which tells the system how the partitions are set. The two partition tables are GPT & MBR. On older devices you'll more often see MBR partition tables, and newer devices will often use GPT. Once the partition table is written you can now write the filesystems to the partitions.

Filesystems					
	Windows Compatible	MacOS Compatible	Linux Compatible	Bootloader	Large Files
FAT32	✓	✓	✓	✓	✗
NTFS	✓	✗	i	✗	✓
EXT4	i	✗	✓	✗	✓

**i** Not always available by default

Filesystems allow you to save data on your device. There are many other useful ones like sshfs, which allows you to remotely manage files on different systems. When installing a new OS to a device, you may have to create a separate partition for the bootloader first. Many installers make partitioning substantially easier, but understanding how they work helps you better suit systems to your needs.

Storage is the very first thing the cloud aims to take away from you. In exchange for access to your files from everywhere, they put your data in their control. Putting your files on your own devices takes a bit more effort, but it's one of the most important ways to secure your data. The easiest way to keep information safe is to not distribute it when you don't need to.

## Network abstraction

*There's no place like "127.0.0.1".*

There are many important systems used to simplify various aspects of connecting to the internet. DNS (Domain name system) allows you you access sites and services without knowing their IP address. Your own system has a [hosts file](#) which is another way to assign names to IPs. For almost every device 127.0.0.1 maps to localhost, which means the device itself. Any service running on your own device will then be accessible from there.

## DNS

DNS (Domain Name System) is what allows your system to communicate with other devices. It works like this: 1. You open your browser and enter <https://fsf.org> 2. Your device asks the DNS server “Where can I find fsf.org?” 3. The DNS server will then ask other nameservers for an up-to-date record of where the site is located 4. The DNS server will then reply to your device with an IP (such as 209.51.188.174) 5. Your browser then loads the page from <https://fsf.org>

This is generally done on the system level in your network settings. A huge privacy concern is that by default most DNS queries are not encrypted which means it isn’t difficult for bad actors on your network to:

- Know what sites you visit
- What online services you use
- How often you use them

This doesn’t necessarily relay any information about the content of the connection however. If you’re using encrypted connections you can be reasonably sure that the content remains safe. This isn’t a huge comfort because for many actors metadata is far more valuable than the content. From a privacy perspective, DNS seems to be one of the most under-appreciated aspects of protecting oneself. In many ways it is because most good VPNs will handle DNS as well, but there are other considerations.

DNS can also be set at the application level. Browsers, for example, will have their own DNS settings. Most major browsers now support [DOH](#), which is a very convenient way to encrypt your browser’s DNS queries. For encrypting all of your system’s DNS queries you may want to consider using [DNSCrypt](#).

## Challenges

### Hardware restrictions

Hardware will significantly limit what options you have to install on your device. While some options may simply be more difficult (Android on the desktop? Say it isn’t so!) others simply won’t be possible at all. This is a significant challenge when “de-googling” android phones. Custom ROMS won’t always be available for less-popular devices. Specific software may not be accessible on all platforms, software you use on Windows may not be available on MacOS or Linux. Even when it’s compatible, both Android and iOS restrict “side-loading” (installing) software outside of the store by default.

### Spying

From phones to fridges, “smart” devices continuously collect data on their users and how the device is used. Most spying takes place on the network or application level but the OS itself may prevent you from removing unwanted applications or services. In almost every case, this telemetry doesn’t benefit you at all. They will often say that the data is collected to improve their services, but in most cases the data collection is the end itself.

## Vulnerabilities

It is important to make sure your OS & programs are up to date to avoid well-known vulnerabilities from being used against you. Law enforcement agencies have deployed malware to bypass encryption.

## Breaking Free


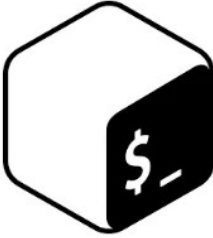
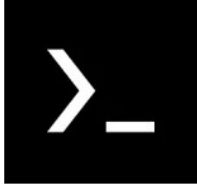
### Removing data from the cloud

Move it all. Move everything off the cloud. The goal of this course is to give you ownership and agency over your own data. You're going to want your own local backups regardless. Have multiple redundant (offline) backups in multiple locations to be safe from corruption, damage, and house-fires.

### Free Software

Free software is software that gives you the user the freedom to share, study and modify it. We call this free software because the user is free. [fsf.org](http://fsf.org)

Using the terminal

	Terminals	
Windows	Linux	Android
		
Powershell	Bash	Termux

While you can do all kinds of things with without interacting with your system's terminal, being able to type out commands can really help you use some very powerful and flexible tools.

There is a fantastic array of software that you can quickly use by typing a few short terms.

- youtube-dl

You can easily download videos from many sites, like twitter, with a simple command. This tool is fantastic for being able to keep local copies of videos that may be censored or otherwise unavailable.

- wget

Downloads files from the web, including archiving entire websites for offline reference.

- ffmpeg

Great for converting videos.

- ssh

ssh is used to remotely manage systems by using their terminal.

For example, imagine you want 10 seconds out of a 40 minute video on Youtube and want to put it on your website. I'm running Linux. Here's one way to accomplish that using the terminal:

```
youtube-dl youtube.com/some-video -o video.mp4
```

(downloads the video to 'video.mp4')

```
ffmpeg -i video.mp4 -ss (start time) -t 10 video-clipped.mp4
```

(uses ffmpeg to clip out the 10 second clip at the chosen start time and save it as 'video-clipped.mp4')

```
scp video-clipped.mp4 my-website:/var/www/website/videos/video-clipped.mp4
```

Using **scp** I then copied the clipped video on to the website's files making it available instantly.

Anyone would then be able to access the video from <https://my-website/videos/video-clipped.mp4>. This does require a webserver and a few other things setup, which you will learn how to set up in future parts. I use this example to demonstrate how much time you can save with a few quick keystrokes. Especially once you start to automate things you do on a regular basis.

[Linux terminal commands](#)

[Basic command prompt in Windows](#)

## Virtual Servers

If you are interested in running your own servers and services you'll likely want to get familiar with Linux and Open Source software. Instead of rushing to reinstall your system, you may want to try it out without changing your current system. Here are some ways to get started:

### VPSs

One of the easiest ways to get started is to spin up a Virtual Private Server (VPS) with a cloud host. This will allow you to try out all kinds of different configurations without using virtual machines yourself. Sign up to [Vultr](#) with this link and get \$100 in free credit.

### Virtual Machines

With [virtualbox](#) you can create systems on top of your system. This gives you a fantastic simulation of what would be an entirely separate device. Virtual machines are handy because you can even pause them to freeze them in time, save the state, and even rollback to that state in the future.

## The World Wide Web

This is one of the most critical posts in the series. The primary battlefield in the information war is the world wide web. A majority of internet users aren't actually users. They are captives of digital kingdoms that harvest their data, control their interactions, and manipulate them. Breaking free requires not only escaping these digital fiefdoms but also helping others do the same.

### Understanding the battlefield

The world wide web is a network of networks. You connect to it through your ISP (internet service provider). Your ISP will then connect to large networks that act as the backbone of the internet. This is what allows almost anyone to connect to all kinds of different sites and services.

### Websites

When you load a website here is what happens:

- Your browser does a DNS query asking for the IP address of the server.
- It then downloads the page, and any data and media associated with that page.
- As it displays the content of that page, any additional scripts running on that page will start automatically.

Websites can have all kinds of content on them. Beyond simple text and images there can be video, audio, as well as interactive content. Extra content isn't always good. Many times your browser can be downloading media and assets from sources that track the people accessing that content. This one of the ways Facebook, Google, and other Big Tech companies are able to track your browsing habits.

Scripts running on pages can do a wide multitude of things, from dynamically updating content on the page, to fingerprinting you to identify you specifically. While you can turn off scripts, many sites will unnecessarily gate content behind scripts and often an account as well. This trains people into constantly handing over more and more personal information in exchange for basic access. This is has fundamentally altered the web from an open information resource to a highly controlled technological terror.

### Services

Servers can send information other than just pages and content, they can process and serve data. This allows many sites to have a more interactive experience. For example there's infinite scrolling, instead of you having to open more and more pages to see results, the page just asks the server for more items and slots them into the page. The opportunities are limitless; and so are the troubles. This functionality allows sites, apps, and even smart devices to monitor and manipulate you.

Services also opened the door for people to be able to establish an online identity. This is what allows people to do all kinds of things like online purchases and comment on posts.

Many services these days have either been naively or malevolently made to collect much more information than would be ideal. Not all services need to operate this way. Where possible it's important to redesign services from the ground up to respect user's privacy & autonomy.

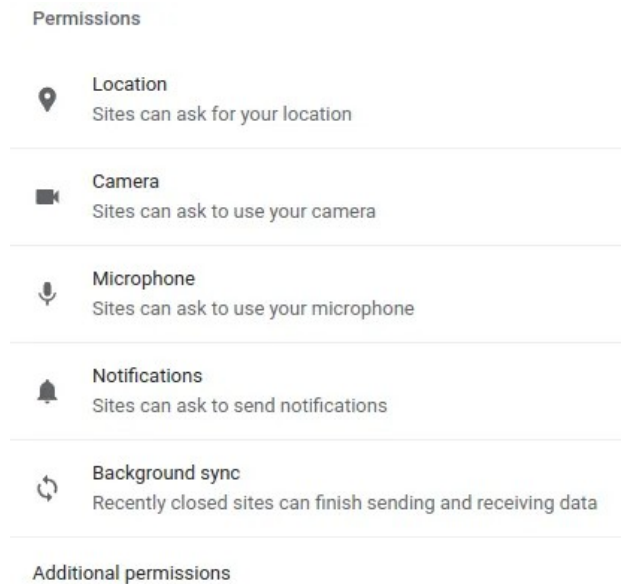
## Your tools: Browsers

Browsers are the program or mobile app you use to access content on the web. [Choosing what browser to use](#) is a complicated topic without a ‘one-size-fits all’ solution.

When taking charge of your online interactions, you need to be aware of how browsers work. Then you can make more informed decisions about what browser to use.

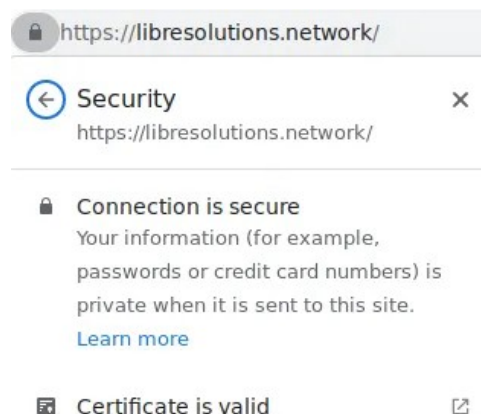
### Permissions

You may trust some sites more than others. Permissions decide what kind of access each site can use. Depending on your settings; sites may be able to autoplay videos, request the camera/microphone, and use your location. As always, the best way to keep information safe is not to distribute it.



## HTTPS

HTTPS is an encrypted connection for the web. This requires the server to have a certificate. The certificate is an encryption key that can ensure the validity of the content. Certificates have to be signed by a certificate authority. Let's Encrypt is a free, efficient way to generate certificates for servers you control.

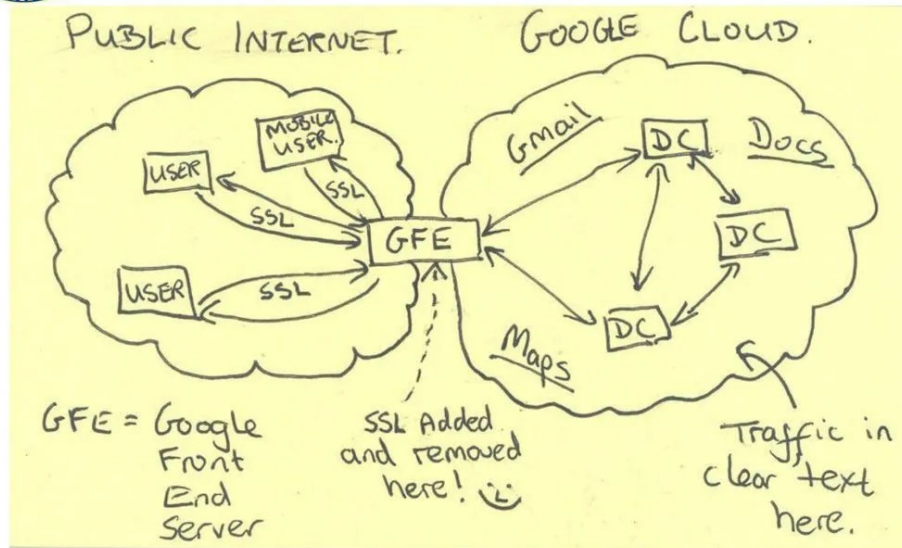


HTTPS is not a perfect guarantee of safety.

TOP SECRET//SI//NOFORN



## Current Efforts - Google



TOP SECRET//SI//NOFORN

*SSL added and removed here :^)*

This diagram from the Snowden leaks means that if an entity can put a server in the middle of the connection, it can remove any and all protection provided by the connection. This is essentially what Cloudflare does in exchange for their DDOS Protection.

### Javascript

Web pages can be enhanced (or corrupted) with javascript. [Interverse](#) is a project designed to help web people discover links between sites. By providing a machine readable index, sites can promote other sites to each other. This can eventually build a fully-decentralized, resilient way to discover sites.

The javascript running interverse does a variety of things:

1. After the page is loaded it downloads a file (`/well-known/interverse`) from the website hosting the client.
2. The browser then directly loads the file from the sites linked to by the origin site, receiving their details and connections.
3. Using a library called Alpine.js the data is then displayed on the page.

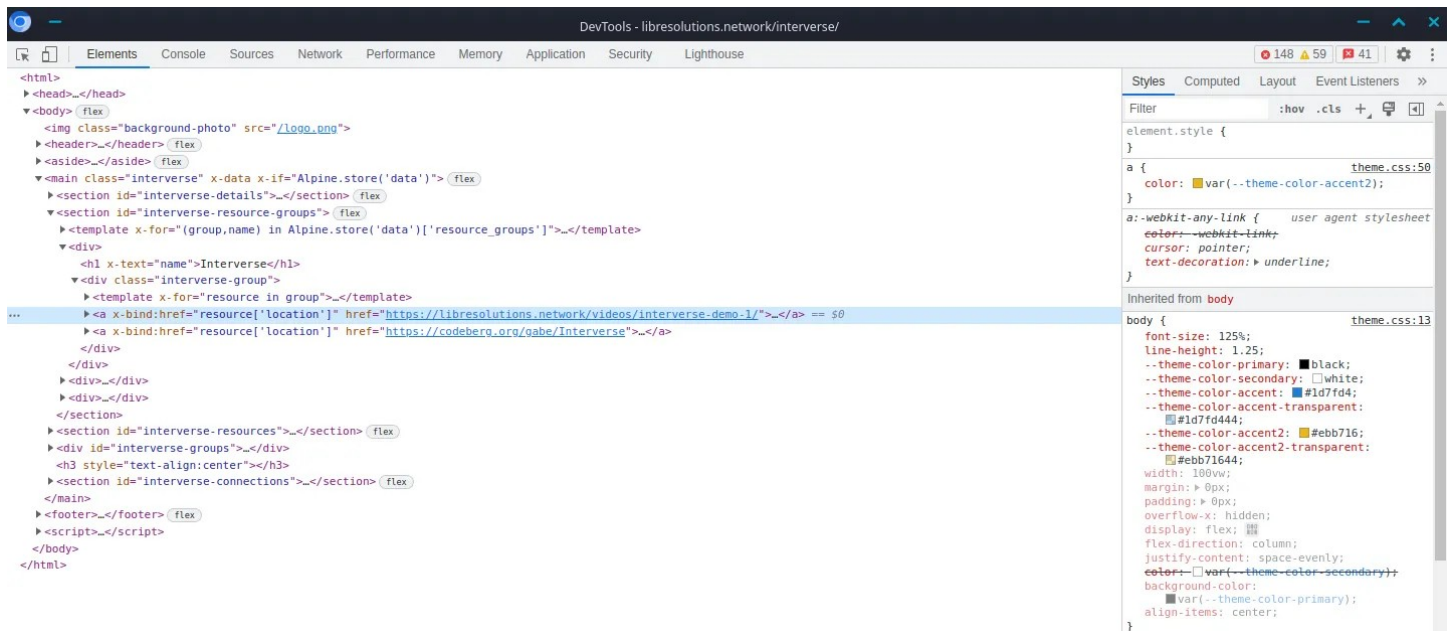
### Dev tools

Both Firefox & Chrome provide very featureful tools for understanding how sites you use work. You can open them with CTRL-SHIFT-I. You will then be greeted with either a new window or a sidebar with multiple tabs including:



## Elements

This tab lays out the entire page's code. You can make changes which will be reflected as long as you keep the page open.



## Console

This will display any errors or logs on the page. You can even run your own javascript on the page.

## Sources

Displays files and media used by the page.

## Network

Lists and profiles any network connections made while loading the page.

## Censorship Avoidance

### RSS

Using an RSS reader allows you to directly download new content from sites directly. This is very useful in reducing your dependence on big tech social media feeds. Here's how you can get the RSS URL from nearly any website:

1. Open the page
2. Search for the word 'RSS' using CTRL-F or "Find on page"
3. If there's no link there, view the source of the page (CTRL-U) and see if there's an RSS link. An easy way to find it may be searching for '/feed'
4. Copy the link
5. Add the link to your preferred RSS reader

## **Bookmarks**

Saving websites you visit by bookmarking sites reduces your dependence on search engines. You can have your browser auto-complete results from your bookmarks so every site is only a few keystrokes away!

## **Privacy front-ends**

Sometimes the content you want to access is on social media sites you no longer have accounts for. Libredirect is a fantastic service for redirecting you to a website that grabs the content for you. The major ones are:

- Invidious for YouTube
- Nitter for Twitter
- Bibliogram for Instagram

## **Communications**

The technocracy is highly motivated to take control of your means of communication. It's no longer simply about who can promote messages to the masses, rather it's increasingly focused on how you communicate with your intimate contacts.

Regardless of how the technical implementation turn out to be in detail: if the plans of the commission should be coming into action, the intrusion into privacy will be fairly deep. Just imagine for each and every message, regardless of a suspicion, to be automatically searched, evaluated and, in terms of a supposed match, reported — not only to the providers but straight to the authorities.

Inevitably, this would include countless perfectly normal, legitimate photos and videos that people send each other. If automatic detection, so far still unreliable, should raise alarm, the content would have to be checked by humans either way. Not only would this violate the right to privacy even more, but as well open another gateway that opens a possibility to be misused.

[Why Chat Control is Dangerous](#)

## **Public vs Private**

There are all kinds of different choices you'll want to make in terms of public vs private communication. For communications meant to be private you're going to want to encrypt them in some way. It's important to be clear with yourself about which communications are intended to be private or which are for the world to see. If you're not careful they'll often get mixed.

## **An N-Way street**

Communications aren't just about sending messages, it's also about receiving them. It doesn't matter if someone has a megaphone to the world if the systems in place have mechanisms to prevent that message from being received by anyone.

## **Means**

Communication ideally involves a connection. It doesn't matter if it's wifi, fiber, or your cellular connection, odds are your communications are being done over one or many networks.

## Resources

In part 1 we defined the primary resources in cyberspace to be:

- Storage
- Computing power
- Bandwidth

Computing power is currently the least important when it comes to communications, in most scenarios either Bandwidth (specifically latency) or Storage become more important. Your communications need to be stored somewhere and the cloud is just someone else's computer. The privacy and security of your devices matters a great deal in keeping your communications safe.

Just as it is important to move your data back onto devices and systems in your control it is ideal to use forms of communication that don't leave your messages on servers outside of your control.

## Encryption

When it comes to communication there can be different stages of encryption:

### 1. Encryption for transport (Network)

Which means that messages can't be read while being sent to their destination

### 2. Encryption at rest (Storage)

Which means that messages are encrypted when stored, and need to be decrypted to be read

The ideal to have both, many services (such as gmail) will say they're protected with encryption, but they only mean (1) encryption over the network, and can be scanned / catalogued.

Another harsh reality when it comes to encrypted communication :

*Not your keys, not your messages.*

Just like when it comes to cryptocurrencies, the data is only as safe as the encryption keys themselves. Some providers will additionally encrypt your messages at rest (2) but they will manage the keys for you. In every circumstance the ideal is to control the keys yourself.

## Mitigation

### End-to-end encryption

End-to-end encryption is a sophisticated method that uses keys from both participants to encrypt messages directly for those participants. This feature is a must-have for private communications on platforms you don't control.

### Peer-to-peer connections

One of the best ways to have direct communication would be to not have to route messages through a server. When trying to do this over the internet (instead of locally over WiFi/Bluetooth ect) your devices need a method to "find each other". WebRTC is a solution that requires a TURN server. This means that the server knows that two IPs are communicating, but has no insight into what is being communicated.

## Keep sensitive information away from these:

- SMS / MMS
- Phone calls
- E-mail
- Hosted apps such as:
  - Facebook
  - Twitter
  - Whatsapp
  - Google Hangouts / Gmail
  - Telegram

This doesn't necessarily mean that you must entirely avoid these, even if in some cases it would do you a lot of good. Where possible you want to at least move any important or private communications to alternative methods.

## Useful Solutions

There are some excellent privacy resources available:

- [DigDeeper](#)
- [Privacy Watchdog](#)
- [Privacy Raccoon](#)

## Delta Chat

[Delta chat](#) is incredibly unique. Instead of building a protocol and new network from scratch, it's built on top of e-mail. This means you can seamlessly interact with your existing contacts with added privacy and security once they start using it.

## Matrix & XMPP

<https://videos.lukesmith.xyz/videos/embed/8fbf5fd7-3d52-43b4-bf0f-baf0a6355519>

## Embrace the Analog

Because [nothing is really safe](#) from state-level actors, in-person and non-digital means of communication are the ideal methods for really sensitive information.

The only way to truly protect data is to never record it in the first place

## Self-hosting

What is self-hosting? It's running your own software and services.

At minimum it means using offline, local software (like LibreOffice) instead of cloud-based solutions. For many it also means running online services from your own systems.

Often when you leave big tech platforms like Microsoft and Google you often want to enjoy many of the useful features like having data automatically sync between devices. The nice thing is that with the power of your own computing you have total freedom in dictating how exactly you want this done.

As you become more proficient at managing your own systems, more options become available. Learning to become comfortable with GNU/Linux systems and how to interact with them through the command-line makes troubleshooting significantly easier.

## **Data sovereignty**

Having your data stored on your own systems is a pre-requisite to computing on your own terms. It is not only your own digital sovereignty that matters but also of those around you. This is because the very systems that manipulate and control people will use any and all information against anyone and everyone.

## **Your own website**

Instead of relying on a cloud provider you can also host your website from your own systems.

## **Fediverse**

Advanced users can often self-host their own fediverse servers.

## **Fantastic self-hosted projects:**

- [Forgejo](#)
- [Misskey](#)
- [Conduit](#)
- [FreshRSS](#)
- Many many [more](#)

## **Helping others**

The more services you run, the more you're able to share those services with those you trust (and hopefully trust you back). Which is a great way to help those who may not necessarily be ready to put the time and energy into setting up their own. Fortunately, the familiarity of using these services may motivate them to try hosting the services themselves.

## **Costs**

- Hardware

Naturally, you will need to have hardware to run services on. This includes not only the routers, computers, & servers but also the storage if you're storing large amounts of data.

In the fight to re-democratize cyberspace, it's important to support initiatives that make hardware more accessible and more free (as in freedom).

- Skill

In many cases there is somewhat of a barrier-to-entry that this course tries to diminish as much as possible. Spending time to learn and practice the skills required is definitely worth the time and effort.

- Start-up time

Some services are simpler to setup than others. It may take some time to configure a service. The more familiar you are with these systems the more you can make use of time-saving methods.

- Maintenance

In most cases you'll want to run your services on a simpler device that uses less energy so that you can keep it running as much as possible. Service interruptions are something you may not always be able to prevent.

- Growing complexity

Depending on how they're organized, the more services you run the more complicated administration of it can be. Unless it's your passion, it may be worth considering if you're running the systems or they've somehow grown to manage you.

## **The Process**

You've found some software you'd like to run. Maybe it's hosted on a github repository somewhere or it's in your Linux distribution's packages. Either way, you have acquired the program and just need to know how to run it.

Services are simply programs like any other. They just usually have some extra steps to get them running.

## **Dependencies**

Software is usually built on other software. Instead of packaging everything together the program may simply rely on the program being installed on the same machine. Dependencies are other programs that the service requires to run.

For example a service that's written in the python programming language will need python installed on the system. This can also include individual python libraries or modules.

Different software ecosystems have their own different ways of managing dependencies. Python uses pip and rust uses cargo to install their libraries.

## **Installing**

In many cases, the installation guide for the service you want to run may tell you what dependencies are required or install them as part of the process. When your Linux system's package manager installs software it usually places it in a specific place on that systems filesystem.

When the service's files are in a desirable location (depending on your system, or personal preference) your system can now attempt to run the software. If it's not properly configured you can usually expect an error, some software may simply act in unexpected ways.

## **Configuration**

Sometimes software will have specific attributes or data that will need to be specific to that installation. This information can be:

- your machine's domain name

- user accounts & passwords
- choosing where persistent files are stored
- any additional 'settings' for the program

Usually you will change these settings in a text file in the correct name and location. Other programs make use of environment variables to access this information.

## **Post-installation**

For remote services, you may also need to do additional steps to make them accessible. For other devices to easily find yours you may need a domain name and an SSL certificate. With these you can securely accept connections for your service.

## **Getting it done - the easiest way**

Today, to start you're going to use a Virtual Private Server (VPS) provider called Vultr. With my [referral code](#) you can sign up and get \$100 in credit to test with.

In many cases you'll want to use your own hardware, but if you're new to GNU/Linux it can be much easier to simply learn with a disposable VPS. This way, you can learn specifically what you need to and start from scratch for the next project.

## **Simplicity without giving up control**

### **Yunohost**

Keep calm and host yourself

<https://yunohost.org/>

### **Libre Server**

Running a personal server for you and your friends or family can be one way to regain agency within an otherwise dehumanizing system. Such servers can federate together to create community oases within the information desert. Growing and connecting in a decentralized way, rather than building a single monolithic platform for the next Silicon Valley tyrant to monopolize.

<https://libreserver.org/>

### **EmbassyOS**

Take back control

Everything you do online is intermediated - your actions permissioned, your data custodied.

Opt out by running a private server.

Previously, it was only available to the tech saavy and the wealthy. Start9 levels the playing field by making it possible for everyone else.

<https://start9.com/>

# Censorship Avoidance

One of the more troubling trends of our time is the outright suppression of alternative points of view. While power-plays and tribalism are essentially hallmarks of the human experience, our modern digital experience introduces new challenges in this domain. With an essentially limitless deluge of information, spam, and outright assaults on your mind, it can be very difficult to navigate it all alone.

Because it's impossible for a person to understand every topic with the detail and respect it deserves, people inevitably have to rely on others on a variety of topics. This creates other challenges because when someone outsources their thinking, (or has been made to outsource it) is that it puts that person at massive risks of abuse by those doing the thinking.

It is impossible to entirely eliminate "outsourcing thinking", but we can actively choose how we direct it. By consciously taking control of or what we allow to influence us, we can move many unconscious choices back into the conscious realm.

Trust is a fundamentally important aspect of this. Do you really trust those who seek to micro-manage every aspect of everyone's lives? Do you trust those who seek to profit off limiting your access to information?

Censorship is more of a cultural problem than a technological one. People have largely accepted large unaccountable corporations being able to eliminate dissent, even when edicts come from governments. Tragically, this means that people themselves do most of the censorship for these large institutions.

## Behavior

The good news is that because it is still mostly people's choices making information out of sight and mind, there are real concrete steps we can take to mitigate censorship.

When ideas or people can't simply be outright banned, a large driver of present-day information control is amplification of prioritized content and voices. Choosing to re-focus your attention onto the voices you choose really goes a long way.

## Feeds

Wherever possible you should prefer chronologically sorted feeds to algorithmically determined ones. Services will often decide what's more "relevant" for you to see first to take control of your time and attention. Even many big tech social media platforms still provide the possibility to view "latest" or "new" items in your feed, where possible it's best to choose this option.

## Lists

Another level of taking control of your information sources involves using the lists features on various platforms. Instead of solely relying on the main feed to find content. Once you have trustworthy people and organizations you can ensure you're not missing any updates from them.

## Bookmarks

Visit websites directly. Bookmark it so you can revisit the website without relying on a search engine to remind you that it still exists.



## Running your own services

A great way of keeping the free and open web alive is to participate. Run your own website outside the control of big tech platforms.

To be accessible to the world wide web you really only need three things:

1. A web server There are many different web servers you can run on your machine. If you're not very proficient at making websites [hugo](#) allows you to generate all the files from simple [markdown](#).
2. A Domain You can get a FREE subdomain at [FreeDNS](#), with a free subdomain you can make your content immediately accessible. It's also a great place to practice how to setup more intricate dns options.
3. SSL Certificate Thanks to [Let's Encrypt](#), anyone can generate a valid SSL certificate on their machine for free. Because these certificates are signed by Mozilla.

### [Remembering Peter Eckersley](#)

4. Use a host If you'd really like to skip all that you can make use of hosting services like [Neocities](#) to get started quickly.

## Tor

[The Onion Router](#) is what's often referred to when people say "the darkweb". The Tor Browser uses the Tor network to access either the "clearweb" which is the internet most people are familiar with, and hidden services.

A hidden service is a server that is accessible from within the tor network. Hidden services don't require a domain be provided to them by any central authority. This means that .onion domains can't be seized.

## Hosting your own hidden service

If you already have your website up and running the Tor project has fantastic documentation on getting started. The basic process is:

- Install Tor
- Modify your Tor configuration file to specify what hidden service(s) you want
- restart tor
- open the hostname file to get your .onion address

## VPNs & Tunnels

If you have a small group of people you trust, you may not need to have your secret services accessible to the world. Perhaps you want your communication devices to solely utilize your private network to avoid leaking all kinds of data.

Using something like WireGuard you could setup your own private network and host private services. SSH Tunnels are also really useful tools you can use to essentially create 1-to-1 hidden services between devices.

### [SSH Tunneling Explained](#)

## Hardware

Hardware matters, now more than ever it's important for people to have access to Free (as in Freedom) devices that can be run on our own terms. Unfortunately manufacturers and [retailers themselves](#) are guilty of tilting the scales against people. This is a must-watch conversation.

## Your Website

### Do I need a website?

The more people who control their own web presence, the less vulnerable we are to information suppression from large institutions. In many ways we are very lucky to be alive in a time where it is so simple, and so inexpensive to distribute important information so widely.

A simple website, is really just a folder with different text and media. In most cases it is outright trivial to serve basic websites out of one's home, but many public hosts exist. By keeping it simple, the web can be more free and accessible for all.

### Introducing: The site generator

While learning CSS, Javascript & HTML is worth doing, there are tools that will transform [markdown formatting](#) into a usable website. This is a fantastic way to save effort, and allows you to focus solely on the content.

Two great ones are:

- [Jekyll](#)

You can download themes [here](#)

- [Hugo](#)

Hugo themes are available [here](#)

I use hugo, so this will be written with hugo in mind. You're welcome to take a look at the [Jekyll documentation](#) to see if it suits your needs better.

### Themes

Themes lay out the overall look and layout of the website. Depending on the theme you use they can change things a great deal, or only a little. Thankfully everything about the theme can be overridden.

Here are my themes:

- [Freedom](#)
- [Valor](#)

These themes are being developed partially as a learning exercise, but also to create an easy way for someone to be able to build media-rich online presence without relying on other platforms. Multimedia embeds are a high priority, and so are social media previews. If you're interested in using these themes your feedback is greatly appreciated!

## Using hugo

Once you've installed hugo, the process is very simple:

1. Create your site with `hugo new site mysite` "mysite" is a placeholder name here.
2. Move into your the project root with `cd mysite`
3. Start the temporary webserver with `hugo server`
4. Open the page with the link provided by hugo (usually `http://localhost:1313`)
5. As you stare into the blank page, imagine the possibilities.

## Making Changes

1. First you'll want to create the index page. Open `mysite/content/_index.md` with your favourite text editor and add whatever you'd like.
2. Download a [theme](#) and place it inside `mysite/themes/`
3. Open `mysite/config.toml` and add `theme = "freedom"` (or the name of a different theme you downloaded)

Provided there's no errors, you'll see your changes refresh in real-time. Once you hit save the page will reload immediately. Once you're done making your changes you'll want to run hugo once and copy the contents of the `mysite/public` folder to your webserver.

## Building up your infrastructure

Building the website is really only the start. Ideally you'll want to make sure your website is on hardware you control. For serving content, computing power isn't a huge concern. Storage and Bandwidth become your primary concerns when serving static content like pages, audio, video, and images.

### Storage

Naturally you need enough storage to store all the files you want to serve. You'll likely want it to be on a fast and reliable storage like SSDs.

### Bandwidth

With large data formats such as video and high resolution images you're going to need a very fast upload speed ideally with very little latency. For most basic uses you may not need a great deal of storage or bandwidth. Video content is a challenge because even relatively small amount of views can become very demanding. When serving video content I would highly recommend keeping file sizes as small as possible.

`ffmpeg` makes that easy with the following options `-s 1280x720 -b:v 750K` if high quality video is a priority I would recommend trying to make use of webtorrent. That way large media can at least be shared between concurrent users.

## Prying eyes

When choosing themes, libraries and even hosts, it's important to make sure that those choices aren't impacting your users. Extra "bloat" really doesn't help much in the long run. Try to use refined and streamlined ways to display your content.


## Data

There are many exciting things you can use Hugo for with relatively simple data setups. Hugo has the functionality to load local and remote data to create interesting pages.

[Fetching Local and Remote JSON in Hugo](#)

## Collaboration

One of the fantastic things about static site generators is that it makes it easy for people to work together on vital projects. Spyware Watchdog is an excellent example of this.




### Online Spyware Watchdog

The goal of this website is to classify spyware programs, so that users can be more aware that they are installing spyware.

Many "everyday" applications are actually spyware. **We take an expansive and strict stance on what constitutes spyware.** We define spyware as anything that includes telemetry, phoning home, automatic updates or is listening in without the user's knowledge or consent.

This website contains articles on popular programs with explanations of the spyware features found in them. Included are easy to understand ratings, detailed explanations, and proof, of how features of these programs can spy on the user.

If you want to contribute to this website, you can always [make a pull request](#). All contributions must be licensed under the CC0 license to be accepted.



Anyone is able to use git to contribute. This kind of activity can scale quite well, and be used in almost any domain. There needs to be more projects like this taking advantage of these amazing collaborative tools to create phenomenal resources

## Sneakernet

The fascinating thing is that these tools don't require an internet connection. People can build private offline resources that could be shared through flash drives at gatherings. This allows people to bring together and share information much faster, and without risks of surveillance and censorship.

[Sneakernet: The Fastest Internet](#)

**Share with others**

