Password Lists - a Discussion

John C Nash

August 7, 2017, revised June 24, 2022, March 26, 2024

Motivation

Most people have a number of bank, utility, email, server, and other accounts for which passwords and usernames may be required, along with other security information. Keeping this all up to date is a daunting task. Moreover, we should consider how our survivors will be able to access such information should we die so executors can manage legacy assets and family memorabilia that may be stored in digital repositories. There are a variety of tools available for managing passwords and extra information. This document reviews some ideas concerning such tools, but each of these imposes its own challenges to learning and to the potential for change over time that we may not be able to manage easily.

Key requirements

Any password "vault" needs to be accessible in multiple ways so that events like fire, flood, machine failure, or theft do not mean loss of availability, nor exposure of the private information to unauthorized persons or machines. However, this requirement can mean there are multiple copies of key files stored in different places. Clearly these will need to be encrypted in as secure a way as we can reasonably manage. However, there are other desiderata we should consider. Let us itemize them, considering that the entire "vault" will be considered as a single collection which we will call PWLIST

- We want to have PWLIST stored where it is accessible whenever and whereever it is needed
- It should be possible to check to ensure we have the correct version, and that we can can update local copies appropriately.
- Local copies must be available when there is no network connection.
- We want secure encryption of PWLIST. Clearly we will need to remember some form of master password or pass-phrase or key, or else have a secure token or device in our possession.
- We want to have relatively simple workflow. Access should be easy and straightforward.
- Our processes should allow for recovery in case of adverse events.
- We should plan for use by executor(s) or attorney(s) in case of death or incapacity.

Encryption

We need to be able to encrypt, decrypt and (if available) view the contents of an encrypted file without leaving readable traces in memory. These processes will be labelled **ecrypt**, **dcrypt** and **vcrypt** respectively. The subject of encryption is very broad, and for the purposes here we will consider only the encryption of a single file, and we will discuss that topic in the companion document **Encrypting a single file** (file FileEncryption.pdf).

Outside services

There are many password services available. It is worth noting

- services using open source software (even if there is a fee for the service) versus proprietary software
- the platforms on which access to passwords is available. In particular, some will NOT be available for some operating systems or for mobile devices, or will be difficult to use on some platforms.

- Fees for service
- Special conditions
- Reputation for reliabliity. Note that hacking of a password service that actually stores passwords may have extremely serious consequences, though not all store data (e.g., Enpass does not). The WIRED article mentioned below notes that LastPass used to be considered "good", but has had security breaches.

There are many such services, with many web-site suggestions, for example, the 2024 list from PC Magazine https://www.pcmag.com/picks/the-best-password-managers, or https://www.wired.com/story/best-password-managers/ It is, unfortunately, not always clear whether such lists are advertising promotions. Nevertheless, they do provide a start to choosing a service.

Some examples (with comments made by colleagues) are:

- Dashlane
- 1Password
- Keeper Password Manager & Digital Vault
- NordPass
- RoboForm
- LogMeOnce
- Proton Pass
- Bitwarden
  Alan McKay: "A couple of years ago at work we went through an analysis of password managers to use at work, and we ended up with Bitwarden as it was the only one that met all of our requirements. Our process for selecting a product like this is pretty extensive. I can't really say much more than that other than that we've been running it in production for about 18 months now and have been extremely happy with it. We chose the on-prem option. It is pretty easy to manage, pretty easy to do regular updates. Pretty easy to have groups of people who share passwords."
  J-F Messier: "I have been using Bitwarden for several years now. My main move to Bitwarden was the fact it is all open source, available to be installed locally on my own servers if I want, and that it was much cheaper than any other solution. I can give some demos, but to give demos of some advanced features, I would need to either have a separate paid account or use my own, but the fact that it contains passwords and keys limits what I could display. I wanted to have my own install of Bitwarden, but never had much time for this. However, I would be willing to have an install on one of my VPS servers in Beauharnois. If there is an interest, I may try to find time to get something up and running. FYI: I used Lastpass previously, but they became very expensive, and numerous incidents were reported about Lastpass, although no passwords were reportedly leaked. As a comparison, Bitwarden costs $10 a year, and their price did not change for multiple years so far."
- Enpass
- KeePassXC
  Aaron Wilcox: "I've been using KeePassX (and now KeePassXC) for a number of years. Cross-platform and open-source (GPL 2 or 3), and it does give you the option of exporting your database to CVS, HTML, or XML. No native cloud storage sync feature, but their FAQ states that this is a deliberate design decision."

Custom services

The basic process for keeping a password or private information file can be set up to use our own resources. That is what my wife and I have done, largely because we needed to handle the survivorship issue, but we caution that is is generally less convenient than using some of the outside services and software.

Basic password viewing process

We consider that PWLIST is essentially plain text (that is, we could view it and edit it with a text editor). However, some users may wish to have PWLIST as a composite file such as a ZIP or tar.gz archive. We recommend, however, that the unarchived sub-file of passwords be plain text. Plain text has the advantage

over other formats that we avoid the possibility that non-visible formatting characters are inserted into passwords which will cause login failures. We abhor documents in word processing formats for this reason. Moreover, we know of no platform that does NOT allow plain text to be viewed.

SIDEBAR

Be careful when copying and pasting information. We have seen failures when the end of line characters (which are NOT the same on different platforms) get appended to passwords. Also some online systems will accept the "Enter" of a password, while others require direct click on a "Login" or "Submit" button on the web page.

It is assumed that we will store PWLIST in an encrypted form. Call this PWLIST.CCC. There are many tools for encryption, but much of the online discussion of these conflates the encryption of a file, directory, disk (i.e., volume) or stream. In the abstract, these are equivalent, but in detailed practice they require different procedures or instructions. See the separate document **FileEncryption.pdf** for some options and discussion. An important additional feature is to **view** an encrypted file without leaving any information in memory that could be recovered.

To simplify the discussion, we will use the alias $REMOTE/ to be the address and directory of the **remote** location of the PWLIST.CCC file. $LOCAL/ will be the location on the device we are using to access or look at the PWLIST.

Pre-requisite steps:

- Encrypt PWLIST to PWLIST.CCC
- Upload, preferably via a secure method such as SSH, to $REMOTE/

Viewing the list

- If $REMOTE/PWLIST.CCC is more recent than $LOCAL/PWLIST.CCC, download it and replace the local copy.
- **vcrypt** $LOCAL/PWLIST.CCC (else **dcrypt** $LOCAL/PWLIST.CCC then clean up assiduously)

Most of the password managers provide convenience wrappers to enable the password to be placed in an appropriate submission box. If we use a custom tool, we will need to copy and paste (and take care not to leave information available for unauthorized access).

Editing the list

- If $REMOTE/PWLIST.CCC is more recent than $LOCAL/PWLIST.CCC, download it and replace the local copy.
- (optional) for security, make a dated, renamed copy of the "old" file, e.g., 20240326PWLIST.CCC
- **dcrypt** $LOCAL/PWLIST.CCC to $LOCAL/PWLIST
- edit PWLIST
- **ecrypt** PWLIST to PWLIST.CCC being careful to use the correct passphrase. Take particular care with CapsLock keys. (We have personal bad experiences of this issue).
- Upload $LOCAL/PWLIST.CCC to $REMOTE/PWLIST.CCC, preferably with a secure transfer method.

Options for $REMOTE/

One of the awkward decisions for password management is where (and possibly how) to store PWLIST.CCC.

- Some commercial offerings have their own servers.
- Some tools break PWLIST.CCC into a number of pieces that have difficult to recognize names. This increases security against unauthorized access but increases risk of loss if there is data corruption.
- Some tools allow the user to nominate a storage location e.g., Dropbox or Google Docs.

- If a private server is available, it is quite easy to use public/private key access with SSH as the transfer method and the server itself for storage. We have used such a solution, but the setup of the sever (a Virtual Private Server or VPS in our case) requires some expertise and effort for setup and maintenance. This is not for unsophisticated users.
- A relatively straightforward choice is that $REMOTE/ is a storage device that can be plugged into whatever platform has to access PWLIST. This works well if there are scripts for the chosen platform and the storage device can connect in a way that allows scripts to reliably read (and for updates, write) to the storage device. USB flash drives, mobile phones and similar objects can be used. Note that there is the advantage that such a storage unit kept in a known location serves executors in the case of death or incapacity.

Using USB or other plug-in storage units

If it is decided to use USB storage units, then it is advisable to use more than one, in case of loss, theft or failure. Failure of flash memory is quite common, and generally happens suddenly. Rotating disks tend to fail incrementally, and retries for reading are a warning allowing for transfer of data to another device.

The awkward aspects of plug-in storage devices are

- when there are updates, we need to update all of them, or at least know which information is outdated
- we have to have the storage unit with us. Given the small size of flash drives, carrying them is not onerous.
- the combination of the above issues implies we should have a method to remind ourselves to update storage units.

Making the process convenient

All the tools to secure information put obstacles in the way of easy use of that information. There is an intrinsic conflict between security and convenience. However, it can be helpful to have scripts to manage the processes, BUT NOT THE ENTRY OF PASSWORDS. It is also important to NOT save passwords where they could be compromised.

Once scripts are available, it is generally possible to prepare clickable icons, and we have done this for our own scripts, even including them on the taskbar.

Extensions of this document

I welcome suggestions and assistance in extending this document.

In particular, users could benefit from,

- examples of scripts and programs and their use
- case studies, particularly indicating advantages and disadvantages for particular users
- updates, reviews and warnings

I can be contacted via profjcnash@gmail.com.