|grepLinux

Exploring Linux, security, and privacy

Account Password Security: Basic Edition

Jan 25th, 2014

The Short Version

Sharing credentials (username and passwords) between the numerous online accounts we have is a difficult dangerous habit to break. I propose the following steps as a manageable way to fix the problem:

- Select password database software like KeePassX or LastPass and if necessary complementary mobile apps
- Track down all of the online accounts you're aware of and scour your email account(s) for accounts you've forgotten; for each account:
  - If you no longer care about the account, delete with prejudice (`pkill -9 $account`) if possible
  - If two-factor authentication is available, set it up
  - Remove any non-critical personal information, especially from legacy accounts
  - Generate a unique random username (if you can change it) and password (with maximum length and largest dictionary) and store it in the password database
  - If a security question is required, create an entry in the password database for a random answer and make note of the site and question in the database entry
- Lock down the security of your password database; use a unique, memorable, and strong password and see the Advanced Edition (coming soon) for more details
- Enjoy the ability click a few buttons to log into your accounts!

Read on for full details.

Motivation and Recent Password Leaks

Passwords for most people are just an annoyance. A common scenario is that an individual has:

- up to just a few basic easy-to-remember short passwords they use as much possible
- a simple way of modifying a password only when a site *forces* them to do so by its unique

requirements
- the same security questions/answers used as often as possible, sometimes unknowingly with publicly-identifiable or otherwise easy-to-find information

From an information security perspective this is disastrous although you can hardly blame these individuals on the state of things; passwords are antiquated and increasingly ineffective given the number of accounts most people have lying around. Indeed, Google is among those trying to replace password-based security with a Universal 2nd Factor, or U2F physical key among other approaches.

That is all for the future though; what we have is a potentially dangerous situation where any *single* site's security being compromised could lead to *several other* of a user's accounts also being compromised due to credential reuse. We live in this unfortunate reality where corporations and other entities have suboptimal or outright poor security practices that allow data breaches to lead to actual username/password combinations and personal information in the hands of criminals.

LinkedIn was one such corporation with poor security practices that resulted in no less than 3.5 million plaintext passwords being dumped publicly, and the original attackers would have both email and password combinations. Ok, so you're a tech geek and you caught wind of this story and changed your LinkedIn password? Well, that's a start, but if you have *any* accounts also using that password then they're at risk as well; Facebook, being aware of this, responded to the recent Adobe security breach by warning its users and prompting security questions and a new password.

For those interested in a more comprehensive list of recent password breaches I advise consulting my long list of password breaches.

A Light at the End of the Tunnel

Even if your password security situation isn't the worst case described above I invite you to ask how close your situation is to the current optimal:

- Every single account has a unique random username (where possible) and password with as large a dictionary as possible
- Every security question has a randomly generated answer
- All credentials are as long as an individual website will allow
- When possible, use two-factor authentication
- A minimum of personal information is stored with each account

Sounds... feasible, right? Ok maybe it sounds absurd and unreasonable, but these days it's completely possible to accomplish this in a highly secure manner that is also easy to use (at least easier than remembering which of the twenty variants of a weak password goes with which site).

Password Management Software

The biggest piece of the puzzle is the software that manages passwords for you. I'm going to highlight two popular solutions, one of which is online-only and closed source and the other offline-only and open source:

- KeePassX (cross-platform)
  - KeePassDroid (Android)
  - MiniKeePass (iOS)
- LastPass (cross-platform)

Note that whatever your choice is, use a **strong password** to access the database! This is going to be one of the last passwords you ever need to memorize and will also be one of the larger security bottlenecks in this scheme, so make it count.

KeePassX

I've written a KeePassX tutorial if you're not familiar with this sort of software — it's fairly straightforward but there are some details to get the most out of its security offerings.

Mobile

To access your password database on your mobile device you'll need an appropriate app; for Android I prefer KeePassDroid and for iOS the highest-rated app I'm aware of is MiniKeePass, though I can't vouch for it.

LastPass

LastPass is a web-based analog to KeePassX; it follows the same concept except the encrypted password database is stored on their servers. In my opinion this is a good reason to go with a locally-stored solution like KeePassX over LastPass; while KeePassX is open source (meaning you have the ability to audit the source code yourself) and local (meaning your security is still entirely in your hands), LastPass requires you to trust the security of LastPass' closed-source implementation which also requires securing a web app against the entire black hat population. Remember that part of the motivation for randomizing our login credentials like this boils down to being unable to trust online services to use the best (or even better-than-awful) security practices. That being said there have been no attacks against LastPass to date that have verifiably leaked any customer information, and as their user base would presumably diminish severely if such a breach occurred they have a strong financial interest to keep their security optimal.

At the time of writing I haven't used LastPass myself, but it is well-reputed and follows the same principles as KeePassX so it serves the purposes of this guide. I'll leave the documentation in the hands of their website; note that it seems that their mobile app requires purchase.

Two-Factor Authentication

This is a method of bolstering the security of password-based authentication; services like battle.net, Google, GitHub, Twitter, Facebook, etc all offer it in some form. The idea is you'll either receive a specific physical device, install a mobile app, or receive an SMS message to provide the "second factor" of authentication: usually a constantly regenerating single-use code. In other words, not only do you have to know the password but you also have to be in possession of a physical device in order to log into an account. This isn't completely failsafe, and targeted malware can still do things like perform man-in-the-middle attacks to steal user credentials. Still, it's going to prevent the shared-credential vulnerability.

Almost every two-factor authentication system I've encountered is compatible with Google Authenticator (Android and iOS), and setting it up can usually be done just by scanning a QR code with your phone. Once set up you'll just pull up the app when logging in and after providing your username and password you'll also provide the single-use code you receive.

Securing Your Online Presence

We're almost there! Let's recap our goals:

- Every single account has a unique random username (where possible) and password with as large a dictionary as possible
- Every security question has a randomly generated answer
- All credentials are as long as an individual website will allow
- When possible, use two-factor authentication
- A minimum of personal information is stored with each account

Finding the Accounts

Once you've acquainted yourself with your choice of password management software, the path to accomplishing the above goals is visible. Unfortunately now comes the most tedious part; depending on how much of an online footprint you've accumulated this could take the better part of a day. You need to track down as many accounts as possible; this might go back well over a decade. Think about it: the weak link could be a random forum from your youth that had no sense of security and is still up and running, and if you shared those credentials with other sites then the potential problem is clear.

You can remember most of the recent websites you've dealt with but your strongest resource for tracking down any accounts you've forgotten about will be your email account(s). Try searching your email for terms like "username," "verify," "verification," "noreply," "account," "welcome," and so on. If you had older email accounts from years ago try to track those down too and do the same for them. At the time of writing I have records for well over 150 accounts.

Securing Credentials

As you find these accounts:

1. Figure out how to log in to them; use password recovery options if needed
2. If you no longer care about them and are able, delete them
3. Generate optimal passwords and store them in your database
4. Clear out any unnecessary personal information, particularly from unused sites
5. If a security question is required, create an entry in your password database with a random answer; make a note there of the site and security question
6. If the site offers two-factor authentication, set it up

Conclusion

This is basically everything that needs to be done. At the end you'll have an encrypted database that contains records for your entire online presence. While this means there's a single point of failure, it's far easier to lock that down (especially if you chose a local password database like KeePassX) rather than potentially hundreds of points of failure.

For those interested in learning how to both lock down this password database and securely sync it between all of your devices using easy to use software, see the advanced edition.

Posted by Isaac Velando Jan 25th, 2014 encryption, infosec, passwords

« Howdy! Linux Fragmentation and New Users »

Comments

Join the discussion…

RolfRen • 5 months ago

I would mention also a great password manager I use - Sticky Password - http:/www.stickypassword.com

Reply • Share ›

Recent Posts

- What's Different About Linux: Programs
- Account Password Security: Advanced Edition
- Not Just the NSA: Privacy Breaches Closer to Home
- Password Security Failure: When Websites Don't Get It
- The @N Hack: Why Absolute Security is a Myth

GitHub Repos

- twitter-blog-broadcaster

  A Twitter bot written in Python meant to broadcast a site's blog posts or similar resources through nicely hashtagged posts (not designed to be followed)

- dotfiles-general

  A collection of assorted dotfiles not directly tied to the desktop environment or window manager

- btrfs-backup-scripts

  A collection of scripts intended for both periodic and manual use for creating backups with a btrfs filesystem

@iwvelando on GitHub