Corelan Team

:: Knowledge is not an object, it's a flow ::

Crypto in the box, stone age edition

Corelan Team (corelanc0d3r) · Wednesday, January 6th, 2016

Introduction

First of all, Happy New Year to everyone! I hope 2016 will be a fantastic and healthy year, filled with fun, joy, energy, and lots of pleasant surprises.

I remember when all of my data would fit on a single floppy disk. 10 times. The first laptops looked like (and felt like) mainframes on wheels and the entire internet could probably be stored on a single hard drive.

Things have changed. Technology allows us to be truly mobile. Mobility is no longer a device, but a concept, a way of living. We've gotten used to being able to access anything anywhere, anytime; so "storing a floppy in a locked desk to keep it safe" is no longer a viable option for most of us.
The increased digitalization and portability comes with an increase of exposure and risk. This is 2016, and luckily we found ways to keep our stuff safe, right?

Just "encrypt your stuff", people say. Encrypt your communications, encrypt your data. Encrypt encrypt encrypt. We've been hearing this for years, so technology must be ready for it too, right?

As the amount of data and the importance of data increases, and as a person who travels a lot, I decided to start using "cloud" storage as my main form of backup a few years ago.
To make things clear, I consider "cloud" to be "someone else's computer", and thus – by definition – a location that I don't trust.
Also, the use of cloud storage is not necessarily a result of my frequent travel endeavours, but rather just an easy way to store data (and backups) off-site. It solves many problems, but it also comes with a new challenge: fix the "trust" issue.
Of course I am using full hard drive encryption. Piece of cake. Pretty much all Operating Systems allow you to do this out-of-the box, in certain cases your hard-drive even supports it natively.
Guess what, it doesn't really protect my data when it gets synced to the cloud. On top of that, as soon as my laptop is running, the hard drive encryption doesn't do much more than slowing down my machine (sigh). Overall, it's probably a smart thing to do (layered defense and all), but it doesn't solve anything as soon as you want to use your device.

Requirements

So I figured I needed additional encryption software. I rolled up my sleeves (ok, not really), and fired up Google.
My requirements were plain simple:

- I have various sets of data. I want to use different keys/passwords for each set, and I want to be able to open/close a set when I want to. I feel this is an important feature because I don't want to have all of my data "open" all the time. Additionally, each set of data deserves and requires its own password. (Even if you don't have both customer data & personal data stored on your machine, I'm sure you can think of reasons to not use the same password for everything.)
  As my data gets backed-up/synced to the cloud and back to my second laptop, I want to make sure the sync process works as efficiently as possible. (That is, don't waste bandwidth and sync fast)
- I am using a combination of Windows and Mac (OSX) devices.
- Although I am a tiny little bit tech savvy, I prefer things to be user-friendly and as transparent as possible. (i.e. when I open a "set", I just want it to be integrated with the rest of my system, without requiring me to do black magic, voodoo, perform 2 back-flips or anally insert a light-saber to use the data first).
- Not all of my data needs to be synced to the cloud. In fact, I don't want some data to be synced anywhere at all.
- I don't mind paying a fee for the tool. There is no free lunch.

Options

Looking at the available applications, I discovered there are mainly 3 options or models:

- Single-Container-based: The application creates a single (big) encrypted file, and allows me to open (decrypt) the file. Upon decryption, the big file gets mounted as a virtual drive and I can use it in a transparent way.
- File based: The tool interacts with the existing files on the filesystem and I have to manually encrypt and decrypt each file in place.
- File based container: Perhaps there is a better term for this model, but similar to the container-based model, the application creates a virtual drive which allows me to access my data in a transparent way. The major difference is that each file & folder gets encrypted individually by the application in the backend (as opposed to being stored into one big file), and thus all encrypted files & folders are stored as an individual file on the filesystem.

Which one is the best? well, it depends on how you look at it.
Let's take the encryption part out of the equation for now. Let's assume (just for simplicity) that there are no backdoors and that the privacy & crypto-aspects are properly implemented and trusted... Of course, I'm just making this statement to avoid a discussion that I won't win, can't win and don't even want to win because all of your arguments are probably right. Since the arguments are probably right for all applications, we can make abstraction of them too.
So – Let's focus purely on the functional requirements and the usability of the tool. (After all, those are the criteria that determine if people will use it or not, whether we like it or not).
I decided to try a bunch of tools, tested all 3 models and tried to find the best model that meets my simple requirements.

Single container

The container-based approach (TrueCrypt, VeraCrypt, ...) works well. It allows me to create individual containers, each with their own key, and runs on Windows, OSX, etc. Upon mounting a container, a virtual drive is created, allowing me to interact with my data in a very easy way. Sweet.
Unfortunately, as everything is stored into a single big (huge) file, the "sync-to-the-cloud" backup process was not as nice. Having to sync a 10Gb file to the cloud everytime I made a change to one of the files was not so nice. In addition to that, I discovered (the hard way) that restoring a deleted file using the cloud backup&restore functionality was not so user-friendly either.

File based

Perhaps a file based approach would fix the sync issue. I tried a couple of tools that allow me to encrypt and decrypt individual files when I need them. Works fine, but I got tired of having to manually encrypt/decrypt every single time. Not so user friendly. Additionally, I found that it's not so easy to find a good tool that is cross-platform and is relatively easy to use.

File based container

After doing some basic Googling, I found a tool called BoxCryptor (now called BoxCryptor Classic). After playing with the tool, I discovered it does the job well, and meets all of my requirements. w00t. Based on the Google results, it also seems to be the only tool that meets the requirements (<surprised face>). Anyway, I was more than happy to pay for a license, to (hopefully) support future development, avoid the tool from dying and allowing the developers to pay their bills.

From a crypto perspective, BoxCryptor Classic is based on encFS and uses the following settings by default:

- Cipher Algorithm: ssl/aes
- Cipher keysize: 256
- Cipher blocksize: 4096
- PBKDF2 iterations: 5000
- Salt size: 20
- Per-block HMAC: NO
- Unique IV: NO
- Chained IV: NO
- External IV: NO

(I'll let the experts judge how secure this is, but not having the last 4 settings enabled, and the number of PBKDF2 iterations set to 5000, and with no option to change these values, it looks like a choice was made to prefer performance over privacy. Again, I am not a crypto expert).

Time passed by, and 'future development' certainly happened. Version2 was released and a decision was made to abandon the BoxCryptor Classic version. I upgraded to v2... and discovered that it doesn't meet the requirements anymore.
I'm not in a position to judge the security of the new architecture and I'm sure they carefully listened to requests of their customers to determine the new feature set; but to me the new (and more expensive) version lacks the "innovative" features that made their product stand out, made the difference, and were part of the reason for me to purchase a license in the first place.
To be more specific, they decided to take away the ability to have individual sets of data, each protected with its own keys/passwords. The new model is a "single set, single password" only model.
Sad news, but at least the BoxCryptor Classic version still worked stable & fine. For the time being.

A few months ago, Apple released 'El Capitan'.
Just like some of you, I have a habit of postponing major updates to my systems until things settle down. Surely enough, people started complaining about a large variety of things, including the use of BoxCryptor Classic.
I found this thread in the BoxCryptor forums: https://forums.boxcryptor.com/topic/boxcryptor-classic-mac-os-x-1011-el-capitan#post-7948
TL;DR: BoxCryptor Classic doesn't work anymore; and is no longer supported on any newer operating system. Yes, there is a dirty workaround, but it will probably die in a future OSX upgrade.
I engaged with their support team, and they confirmed that they will stick to their decision to only support v2 on new systems, and that it is too complicated to put the original features back.

Bummer.

Back to square one. (and no, I didn't see that coming)

So – what's next?
I don't know. I am being told to encrypt encrypt encrypt, and I've been telling people to do the same. But I don't know how to make it really work. I mean, how to make it usable. Perhaps my requirements are too progressive or just plain stupid? I don't know either. I don't claim to be smarter than anyone else, but I can imagine that a lot of people feel the same.

For now, I'm holding off from upgrading to El Capitan, but that feels more like fighting symptoms rather than fixing the issue. I hate doing that.
I've tried to find a workable alternative. After all, a few years have passed since I started using BoxCryptor, so I was convinced that there would be other tools that do the same (both from a crypto and usability perspective).
TL;DR: Nope. The only "similar" alternative would involve building a custom encFS implementation, using potentially unstable/unsupported ports... Not exactly the user-friendliness, reliability that is going to convince anyone to use the solution. It doesn't really matter that "it is open source, so you can improve it".

Is crypto ready for use in a modern environment? Perhaps. (I'm not smart enough to truly answer that question)
Are the tools ready? Based on my experience: Not so much. (or at least, not that many)

Conceptually, crypto works and tools do the job.
But unless I missed something, the current state of combined implementations in a modern/hybrid 'anything anytime anywhere' society, seem to put us back into a stone age reality.
Perhaps one of the reasons is based on the fact that not enough people are using it. Then again, it may be too difficult to use the tools in the first place. Or the tools may lack important features that are required under the current state of technological evolution and behaviour.

Can it be done? But of course. I believe it's "just" a UI and feature-set problem.
All it takes is to bring crypto folks together with UI designers and people that build a feature-rich application, something that people can use, not just something that does the job.

Why doesn't that happen (or doesn't it happen more often)?
I don't know.
The BoxCryptor folks had it right, and I'm not the only person who thinks so.
Big kudos to them. Too bad they don't fully realize that they have abandoned a raw diamond.
I strongly believe (and hope) that someone else will take the ideas and run with it.

The good news is that nothing prevents anyone from making something that can be used in a simple way, and more advanced way if needed.
Whisper Systems is a perfect example of how things are done right. People talk about Signal and people use it because it works so well.

And besides, nothing prevents anyone from asking a small fee for their work. People will buy your stuff if it works. Even if it's open source.

I need your help!

I'd like to finish this story/rant with some questions:

What would it take for someone to build something that makes the difference and closes the gap between technology and something people actually use?
What would be a good incentive ?
What would it take for our community to go from yelling advise to actually making it possible to follow the advise?
Are we truly a community if we can't join forces and convince more people that usability matters?

I look forward to seeing your answers, questions, comments, suggestions. And I also look forward to supporting initiatives that will improve our lives.

Hackers bend rules and seek improvement... Be a hacker.

This entry was posted
on Wednesday, January 6th, 2016 at 12:55 pm and is filed under 001_Security, Crypto
You can follow any responses to this entry through the Comments (RSS) feed. You can leave a response, or trackback from your own site.