LUKS, VeraCrypt, CryFS, EncFS, gocryptfs, etc?

Uma discussão prática sobre criptografia

Esse artigo traz uma discussão resumida sobre os tipos de criptografia disponíveis em nível do ‘dia a dia’ e sobre as implicações do seu uso, tanto localmente como em nuvem. Não foram abordados aqui procedimentos técnicos avançados a respeito do funcionamento teórico dos tipos citados.

Bem, a primeira coisa que geralmente pensamos quando vamos criptografar algo, seja um arquivo ou pasta, é saber qual senha digitada será, ou estou errado? Enquanto somente isso parece dar uma impressão de ser uma informação valiosa, a forma como você irá produzir e gerir esse arquivo criptografado vai direcionar muito mais o ciclo de vida do seu arquivo, como por onde você pode transportá-lo ou como ele vai crescer ao longo do tempo.

Basicamente, existem dois tipos básicos de criptografia, **volume cheio** e **volume adaptativo**. As duas são descritas em muitos softwares como volu*mes ou containers. Essa* denominação foi criada por mim, nesse texto, como método pedagógico para ajudar na explicação a seguir.

Mas qual a diferença entre elas?

A criptografia de volume cheio funciona através da criação de um volume com um tamanho pré-definido, como se fosse um mini pendrive. Por exemplo, com ela é possível o usuário criar um arquivo de tamanho fixo em que ele possa abrí-lo no sistema operacional e gravar os dados dentro. Esse processo é percebido, por exemplo, quando uma partição do disco rígido é criptografada no ato da instalação do SO, onde antes de ele carregar, irá chamar uma janela para usuário digitar a senha de descriptografia, para após isso, o volume ser montado como sistema de arquivo e poder ser utilizado pelo usuário. De forma análoga, você pode ter vários volumes em algum diretório do seu sistema operacional, então montá-los e desmontá-los a qualquer tempo.

Os tipos mais comuns de criptografia de volume cheio são: LUKS, Veracrypt, Truecrypt e PLAIN dm-crypt.

Já a criptografia de volume adaptativo, assim como o nome autoexplicativo, funciona gravando cada pedacinho de forma separada dentro do volume criado, e cresce de forma contínua. A diferença entre a primeira é que não é necessário o usuário definir o tamanho do volume na hora da criação, mas ele crescerá com o tempo, de forma adaptativa.

Os tipos mais comuns de criptografia de volume adaptativo são: CryFS, gocryptfs e EncFS.

Bem, até aqui você já deve ter feito um breve resumo sobre o que escrevi e provavelmente o resultado trás um conceito prévio: o espaço em disco.

O espaço em disco é um ponto crucial que deve ser levado em consideração na criação dos volumes criptografados. Para deixar um pouco mais claro, vou trabalhar essa questão com exemplos do dia a dia.

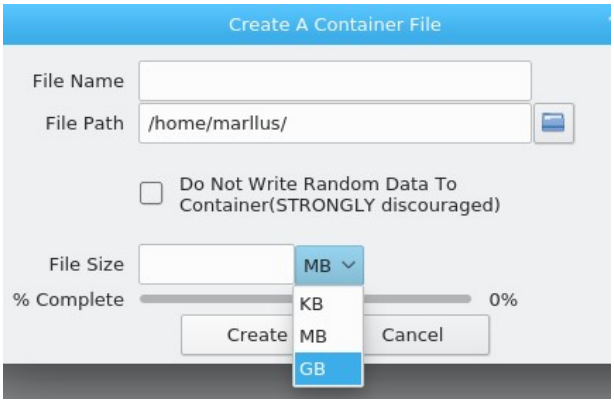
Imagine que você tem um repositório de 10GB na nuvem, disponíveis no serviço Dropbox. Então, sabe que pode sincronizar isso tudo para lá. Então, se você escolher criar um arquivo criptografado do tipo volume cheio no seu sistema operacional, terá que especificar o tamanho dele. Vamos supor que você coloque 5GB. Então, após a criação e primeira sincronização do volume para o servidor do dropbox, você gastou 5GB de banda e provavelmente causou uma demora e lentidão no upload da sua rede. Agora, imagine logo após isso você montar o volume no sistema operacional e criar um arquivo de texto de 10KB dentro e então fechá-lo. O serviço de sincronia do dropbox considerará aquele volume como um bloco único alterado e como ele está criptografado, o dropbox não tem como saber o que é isso e nem fazer controle de versão. O que ele vai fazer? Sincronizar os mesmos 5GB novamente. Muito ruim heim? Quase impraticável.

É nesse caso que os sistemas de criptografia que trabalham com volumes adaptativos caem como uma luva. Primeiro, que ao criar um volume desse tipo não é preciso especificar o tamanho prévio dele, portanto, ele será quase zero de espaço. Então, partindo do mesmo exemplo, mas agora com um volume adaptativo, a primeira sincronia levaria provavelmente alguns segundos. Então, se você criasse um arquivo de texto com os mesmos 10KB dentro do volume, o serviço do dropbox reconheceria os blocos alterados e enviaria somente eles, de forma adaptativa. Isso quer dizer q a segunda sincronização iria te custar bem menos banda de upload. Muito melhor, né?

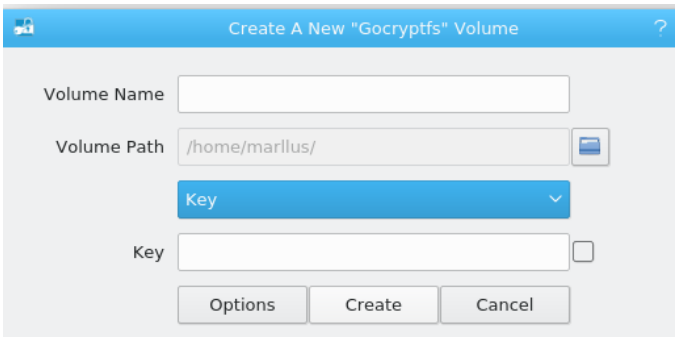
Por fim, um último adendo. Na criptografia de volume cheio, o tamanho do volume definido no ato da sua criação é fixo durante toda sua vida e não pode sera alterado. Já a de volume adaptativo, ele pode crescer continuamente, e o que definirá seu tamanho total, de fato, será a informação escrita dentro dele.

> *Resumindo, o custo de se usar criptografia para sincronia na nuvem vai depender do tipo de criptografia escolhida. Caso escolha errado, terá consequências desastrosas. Portanto, se for trabalhar com volume cheio para sincronizar na nuvem, eu recomendo a criação de volumes pequenos, caso contrário, adoção da criptografia de volume adaptativo.*

Seguem abaixo dois prints. O primeiro é um ato de criação de um volume cheio e o segundo de um volume adaptativo.



Criptografia de volume cheio: você tem que especificar o tamanho prévio



Criptografia de volume adaptativo: o tamanho total será definido com a quantidade de arquivos ao longo do tempo

Você deve estar se perguntando o por quê de não usar somente criptografia de volume adaptativo, já que ela parece ser muito boa, pois pode ser facilmente sincronizada através das mais diversas plataformas de nuvem. Bem, é uma pergunta pertinente e isso mereceria outro artigo com discussões a respeito da utilidade, as restrições, tipos de arquivos e locais de sincronização. Mas, encerro por aqui com a sugestão de duas ferramentas com interfaces de janela bem legais para criação e gestão de volumes criptografados ‘cheios’ e ‘adaptativos’. A primeira é a ZuluCrypt/ZuluMount, que trabalha com LUKS e VeraCrypt (recomendados) e a segunda é a Sirikali, que faz a gestão de volumes adaptativos, como CryFS e gocryptfs (recomendados).

Há de se considerar questões sobre segurança e eficiência dos algoritmos de criptografia mencionados. Aqui no site do projeto CryFS você pode ler um ótimo texto sobre isso, além dos pontos elencados nesse artigo. Para mais informações a respeito da performance, recomendo um comparativo realizado no site do projeto do gocryptfs.

Obrigado!

:)

Referências: https://marllus.com, além das citadas no texto.