GnuPG

31 de outubro de 20154 de novembro de 2015 / gjuniioor

Índice

Introdução

Gnu Privacy Guard 0 (ou GnuPG, ou mais reduzido ainda, GPG) é uma implementação do protocolo de cifragem de mensagens OpenPGP.

Em uma vídeo aula 1, disponibilizada no canal Ciência Hacker, foi mostrado o uso do GPG. Aqui vai um artigo para introduzir o conceito e utilização dele ao leitor.

OpenPGP

OpenPGP é um protocolo muito utilizado, previamente, para criptografia de e-mails. Baseado no PGP 2, surgiu dando ao mundo um método forte e “simples” de criptografia. Utiliza chaves públicas (assimétricas – recomendo a leitura desse artigo 3 para melhor entender o contexto) como engenharia de sua encriptação e define formatos para gerar mensagens encriptadas, assinaturas em arquivos, certificados e demais…

Aqui 4 pode ser encontrado o site do OpenPGP e há também um episódio do podcast Segurança Legal que fala sobre ele, pode ser acessado por aqui 5.

Instalação

Sendo breve nessa parte. GnuPG vem por padrão na maioria (senão todos) os sistemas *nix. Mas, caso precise, vai alguns comandos que podem ser úteis:

- Arch: `sudo pacman -S gnupg --noconfirm`
- CentOS: `sudo yum install gnupg -y`
- Debian: `sudo apt-get install gnupg -y`

Ou, caso precise, pode acessar a página de download no site do projeto 6.

Uso

O uso do GPG se abrange bastante! Pode ser utilizado desde clientes de e-mail à troca de arquivos. Ele trabalha não só, mas também com o uso do conceito de chaves públicas e privadas. Ou seja, podemos gerar um par de chaves em que a privada fica sob nossa proteção e a pública nós disponibilizamos para o pessoal. Quando quiserem falar comigo, utilizam da chave pública para encriptar os dados e portanto, somente eu poderei ler. Quando eu quiser assinar algo com minha chave, poderão ver pelo fingerprint, por exemplo, que realmente fui eu quem o fiz.

É até comum ver pessoas passando o código de fingerprint de sua chave em páginas de contato, por

exemplo, para que possam ter um nível de confiança maior.

Mas como dito, ele não trabalha apenas com o uso dessa chaves. Você pode criar uma frase de segurança com algum amigo e utilizá-la para criptografar dados com ela. Vamos ver isso em nosso primeiro caso. Posteriormente, abordaremos a geração do par de chaves e seu uso.

Encriptação com Frase

Para isso, basta ter o programa instalado e rodar o comando:

```
$ gpg -c file
```

```
1 $ gpg -c file
2
```

Em que o `file` é o arquivo que deseja encriptar.

Vai ser pedida a frase de segurança. Não preciso nem comentar sobre a utilização de caracteres especiais e senhas realmente fortes para isso, certo?! Após o processo, será gerado um arquivo com extenção `.gpg`, que é o já protegido com a frase.

A opção `-c` abordada apenas indica ao GPG para que use apenas de criptografia simétrica no processo, sem a necessidade das chaves.

Para ter o arquivo limpo para ser lido, basta usar:

```
$ gpg file.gpg
```

```
1 $ gpg file.gpg
2
```

Caso deseje, pode fazer utilizando a opção `-d`, mas essa opção apenas irá exibir o conteúdo do arquivo na tela e não criar um novo arquivo livre para ser lido. No caso de uma mídia (imagem, vídeo...) ou um binário em si, talvez não seja tão interessante.

Bem, no mais, isso já ajuda bastante. Em trocas de arquivos, almeje ao menos por utilizar esse processo.

Gerando Chaves

Vamos começar a melhorar o uso do GPG. Como foi dito, ele utiliza a questão de chaves pública/privada para vir à tona sua criptografia assimétrica (caso não saiba do que se trata, recomendo novamente que leia o artigo indicado 3). Então, vamos lá!

O GPG cria um chaveiro que é onde você poderá gerenciar as chaves que você precisa. Ou seja, basta inserir a chave pública de alguém em seu chaveiro, e quando for precisar para encriptar/decriptar algum dado, poderá indicar de quem é e voilà!

Vamos, num primeiro momento apenas criar nosso par de chaves. Para isso basta utilizar o comando:

```
$ gpg --gen-key
```

```
1 $ gpg --gen-key
2
```

Primeiramente será perguntado sobre o tipo de chave que queira usar. Já é informado no próprio menu que DSA e RSA é usado apenas para assinatura e há outras duas opções: RSA (e RSA) e DSA (e Elgamal). Por padrão, a primeira opção é marcada. Manteremos ela.

Logo após vem a pergunta sobre o tamanho da chaves RSA, que pode ser de 1024 até 4096 bits. 1024 já é um tamanho legal, mas comparado às opções, a mais fraca. 4096 por outro lado seria a mais segura, porém mais demorada. O programa indica utilizar 2048, vamos aceitar também a sugestão.

Depois ele questiona sobre o tempo de validade da chave. O número 0 significa que a chave não deve expirar. Qualquer número representa os dias de validade. O número seguido de w, m ou y indica

semanas, meses ou anos, respectivamente. Como podemos mudar a “frase senha” (mais para frente), podemos numa boa utilizar 0.

Após, basta fornecer algumas informações, como nome, endereço de e-mail e um comentário. Depois desse procedimento é pedido uma “frase senha”. Bom investir em segurança nela, usando os conceitos de uma boa senha.

Feito isso, sua chave já está sendo gerada. Ele precisa de alguns dados “rândomicos” para fazer isso de forma mais elegante. Como sugestão, o programa pede para que façamos alguma atividade como digitar no teclado, mexer o mouse ou utilizar o disco. Você pode fazer como quiser qualquer um desses, apenas forneça o que lhe é pedido.

Adicionando Mais Informações às Chaves

Caso você tenha mais de um endereço de e-mail que utilize, é uma boa prática colocá-lo também como identificador à sua chave. Para isso, basta inserirmos esse novo identificador.

Para editar a chave, você precisará do comando:

```
$ gpg --edit-key id
```

```
1 $ gpg --edit-key id
2
```

Esse ID representa algo que identifique a chave que quer alterar. Pode pegar algumas informações da saída do comando: `gpg --list-keys`.

Após entrar no prompt do GPG pode utilizar de comandos como `?` ou `help` para exibir um menu de ajuda e não ficar tão perdido num primeiro contato.

Utilizando o comando `adduid` você poderá inserir o novo endereço de e-mail, apelido ou o que quer que seja. Terminado esse passo, use o comando `save` para salvar as alterações e sair do prompt.

Se você já tiver enviado sua chave para um servidor sem essa nova informação basta reenviar e será atualizado. Esse passo de mandar para um servidor será visto mais adiante.

Gerenciando o Chaveiro

O chaveiro, como já foi dito, é algo criado pelo GPG para armazenar as chaves que você adiciona. Vamos ver agora um pouco do uso do chaveiro.

Você poderá listar as chaves que tem com o seguinte comando:

```
$ gpg --list-keys
```

```
1 $ gpg --list-keys
2
```

Caso queira ver as chaves privadas, basta utilizar:

```
$ gpg --list-secret-
key
```

```
1 $ gpg --list-secret-key
2
```

Normalmente aqui só é mostrada uma chave privada, a sua, ou então de grupos que utilize assim.

Exportando/Importando Chaves

Caso queira exportar para encaminhar para algum contato, utilizar num cliente de e-mail, de chat, ou enviar para um servidor (caso precise ser dessa forma), vamos ver como ocorre.

Exportando

Tenha em mente que para qualquer processo nesse passo precisaremos de um identificador da chave. Para pegar essa informação, liste as chaves que tem e pegue uma informação única à elas.

O comando básico para exportar uma chave é o `--export`. Ele irá mostrar a chave como ela realmente é, em formato binário e tal. Caso queira ver em formato ASCII, utilize a opção `-a`. Para salvar num arquivo, pode utilizar redirecionadores ou então a opção `--output`. Vamos ver alguns exemplos do mostrado até então:

```
$ gpg --export -a id &gt; chave.pub.asc
```

```
1 $ gpg --export -a id &gt; chave.pub.asc
2 $ gpg --output chave.gpg -a --export &lt;id&gt;
3
```

Tenha em mente que a saída em modo ASCII é o mais utilizado por aí. Portanto, uma boa opção ter exportado nesse formato.

Para exportar a chave privada, a única coisa que muda é que no lugar de `--export-key` será utilizado `--export-secret-key`.

Exportando sua chave primária, é bom guardá-la em local seguro! Pois sua chave pública você terá em keyservers (caso utilize esse artigo até o final), já a privada não.

Importando

Para importar uma chave é simples, basta utilizar:

```
$ gpg --import key.gpg
```

```
1 $ gpg --import key.gpg
2
```

> OBS: O programa não faz distinção de chave pública ou privada na importação. Ou melhor, não exige que você especifique.

Após importar, deve-se assinar para mostrar a autenticidade da chave. Utilize o comando:

```
$ gpg --sign-key id
```

```
1 $ gpg --sign-key id
2
```

Em que `id` equivale ao código da chave mostrado. Por exemplo BC71CF75.

Esse código você pode confirmar, por exemplo, telefonando para o dono da chave, conversando pessoalmente, vendo se ele realmente afirma que essa chave é dele em seu site, anyway.

Encriptando Arquivos com Chaves

Tendo em mente (ou no clipboard haha) alguma identificação para a chave que deseja utilizar, basta utilizar:

```
$ gpg --encrypt arquivo
```

```
1 $ gpg --encrypt arquivo
2
```

Pode utiilizar ou não o `--output`, é à moda do freguês.

Após esse comando será pedido alguma informação que identifique a chave que deseja utilizar. Coloque então o e-mail, nome, ID ou algo dessa natureza para selecionar a chave correta.

Decriptando Arquivos com Chaves

Para decriptar é bem simples. Pode-se utilizar o `--output` ou não, fique a vontade:

```
$ gpg --decrypt
arquivo
```

```
1 $ gpg --decrypt arquivo
2
```

Note que não é necessário especificar a chave pois a ideia é que você possua em seu chaveiro. Caso não seja encontrada, um aviso é mostrado na tela, e poderá utilizar o ID da chave para baixar dos servidores ou buscar em outro lugar.

Utilizando Servidores

Key server utilizam do protocolo HKP (HTTP Keyserver Protocol), que seria uma adaptação do HTTP para uso com o transporte de chaves.

Há vários servidores espalhados por ai, basta você escolher o que quer fazer uso. Por exemplo, o servidor do próprio projeto PGP 7 ou GnuPG 8. Muitos desses servidores, após receber uma nova chave, replica para parceiros, assim, mantendo sempre a possibilidade de ter todas as chaves online. Ou seja, muito provável que após enviar para o servidor do GnuPG, em instantes esteja no do PGP também, ou do MIT e assim por diante.

Para utilizar um servidor em específico basta utilizar o parâmetro `--keyserver` e passar o servidor como argumento. Por exemplo:

```
$ gpg --keyserver
hkp://keys.gnupg.n
```

```
1 $ gpg --keyserver hkp://keys.gnupg.net:11371 --search-keys gjuniioor
2
```

A porta 11371 é padrão nesse tipo de servidor, mas nada impede o administrador de mudar caso necessite.

> OBS: O parâmetro `--search-keys` será melhor abordado mais para frente.

Enviando Chaves

Então, para poder passar de forma fácil sua chave para amigos e contatos é bacana que você envie para um servidor. É bem simples o processo:

```
$ gpg --send-key id
```

```
1 $ gpg --send-key id
2
```

> O ID é a identificação da chave, aquele código, como BC71CF75 que já foi abordado aqui.

Note que nem foi preciso indicar um servidor, a aplicação utiliza por padrão o servidor do GnuPG.

Há alguns servidores que dispõem de uma interface web para fazer buscas ou enviar as chaves, como o do PGP por exemplo. Basta utilizar o protocolo http e não hkp em si, na mesma porta, que ele responde.

Baixando Chaves

Agora, para importar a chave de alguém que esteja em determinado servidor é bem simples. Basta saber alguma informação que seu contato tenha usado no cadastro da chave, por exemplo, nome, e-mail, comentário, apelido ou o próprio ID da chave (como disse, é comum que disponham-as em websites pessoais). Após isso, basta utilizar o comando:

```
$ gpg --keyserver
hkp://subkeys.pgp.
```

```
1 $ gpg --keyserver hkp://subkeys.pgp.net:11371 --search-keys gjuniioor
2
```

Isso vai procurar minha chave (pois bem, já pode criptografar as coisas caso queira falar comigo, nem que seja para teste, para ver se entendeu bem o conceito e tudo mais).

Será mostrada uma lista com as chaves que foram encontradas e uma numeração à esquerda representando, nesse momento, a identificação da chave. Note que há um prompt do GPG esperando pelo seu comando. Se colocar o número da chave, irá adicionar ao seu chaveiro. Se inserir `Q` vai sair do GPG.

Conclusão

Vimos nesse artigo a utilização do cliente por linha de comando. Mas há vários softwares GUI feitos com o objetivo de simplificar o uso total do GnuPG, basta uma simples busca e irá encontrar algumas opções. Não vou indicar nenhum aqui pois nunca os utilizei, sempre preferi CLI mesmo.

Após tudo que foi aprendido nesse artigo, é bom saber o fluxo comum de uso do GPG caso não tenha ficado claro:

1. Envia sua chave pública para as pessoas. Elas utilizarão as delas para criptografar os dados para apenas você poder ler após utilizar sua chave privada.
2. Pega a chave pública das pessoas e criptografa a informação com ela, para apenas essa pessoa ter acesso com sua chave privada.

Pode ser utilizado também no caso de grupos, em que um integrante criptografa algo com a chave pública desse grupo e todos (e apenas) que tem a chave privada poderão ter acesso.

É extremamente importante manter suas informações seguras! O risco físico (que pode ocorrer caso perca o celular ou tenha o notebook roubado) é pequeno quando comparado aos prejuízos abstratos, como leitura de e-mails ou divulgação de seus nudes.

Não fique pensando que o que está aqui é tudo! Há muito mais a ser abordado. Abaixo há mais links para te dar uma trilha de estudos a seguir… E no mais, rodar um `man gpg` funciona muito bem! Se divirta! ;)

Referências

0 - https://gnupg.org/

1 0 - https://gnupg.org/
2 1 - https://www.youtube.com/watch?v=XjkEXpv37Nw
3 2 - http://www.pgpi.org/
4 3 - http://eofcommunity.com/forum/viewtopic.php?f=33&t=1265
5 4 - http://www.openpgp.org/
6 5 - http://www.segurancalegal.com/2015/03/episodio-71-openpgp.html
7 6 - https://www.gnupg.org/download/
8 7 - hkp://subkeys.pgp.net:11371
9 8 - hkp://keys.gnupg.net:11371
10