Nesta atividade, você vai aprender a usar o openssl, uma ferramenta de criptografia de código aberto, para proteger seus arquivos de texto no terminal linux.

Você pode instalar a biblioteca openssl no ubuntu utilizando o comando

```
sudo apt install openssl
```

Para quem não tem linux, sugiro utilizar WSL.

Caso não seja possível, tente instalar o openssh no PowerShell do Windows utilizando o winget.

```
winget install --id=FireDaemon.OpenSSL  -e
```

OU baixando o instalador neste site:

https://www.firedaemon.com/download-firedaemon-openssl

Ao final é preciso adicionar o openssl ao path do windows. Pesquise como adicionar a pasta de instalação “C:\Program Files\FireDaemon OpenSSL 3\bin” do windows. Verifique se esta é a mesma parte do seu computador.

Será um exercício interessante para se acostumar com configurações de computadores.

Vamos começar!

Passo 1: Criar um arquivo de texto

Primeiro, você precisa criar um arquivo de texto que vai ser criptografado. Você pode usar o comando `echo` para gerar um arquivo com conteúdo automaticamente. Por exemplo, digite:

```
echo "Este é um arquivo de texto simples." >
mensagem.txt
```

Você pode trocar a frase entre aspas pelo conteúdo que você quiser.

Para verificar se o arquivo foi gerado corretamente, você pode digitar o comando a seguir para ver o conteúdo.

```
cat mensagem.txt
```

Passo 2: Criptografar o arquivo de texto

Agora que você tem um arquivo de texto, você pode criptografá-lo usando o openssl. O openssl suporta vários algoritmos de criptografia, como AES, DES, RSA, etc. Você pode ver a lista completa de algoritmos disponíveis digitando:

```
openssl list -cipher-algorithms
```

Para este tutorial, vamos usar o algoritmo AES-256-CBC, que é um dos mais seguros e populares. Para criptografar o arquivo de texto usando esse algoritmo, digite:

```
openssl enc -aes-256-cbc -in mensagem.txt -out mensagem.enc
```

Esse comando vai pedir que você digite uma senha, que será usada para gerar uma chave de criptografia. Escolha uma senha forte e memorize-a, pois você vai precisar dela para descriptografar o arquivo depois. O comando vai gerar um novo arquivo chamado mensagem.enc, que é o arquivo de texto criptografado. Você pode verificar que o arquivo está criptografado usando o comando `cat`:

```
cat mensagem.enc
```

Você vai ver que o conteúdo do arquivo é ilegível, pois está codificado em uma forma binária.

Passo 3: Descriptografar o arquivo de texto

Para descriptografar o arquivo de texto, você precisa usar o mesmo algoritmo e a mesma senha que usou para criptografá-lo. Você pode usar o openssl novamente, mas desta vez com a opção `-d`, que significa descriptografar. Digite:

```
openssl enc -aes-256-cbc -d -in mensagem.enc -out mensagem.dec
```

Esse comando vai pedir que você digite a senha que usou para criptografar o arquivo. Se você digitar a senha correta, o comando vai gerar um novo arquivo chamado mensagem.dec, que é o arquivo de texto original. Você pode verificar que o arquivo está descriptografado usando o comando `cat`:

```
cat mensagem.dec
```

Você vai ver que o conteúdo do arquivo é o mesmo que você digitou no passo 1.

Exercícios para reflexão

Experimente com pelo menos mais dois outros algoritmos. Pesquise as adaptações que podem ser necessárias para utilizar outros algoritmos. Use o comando time (linux) ou Measure-Command (powershell) para verificar o tempo de execução de criptografia de diferentes algoritmos.

Reflita:

1. Qual é a chave utilizada para criptografia?
2. Use algoritmos com tamanhos de chave diferentes. O tempo de processamento aumenta ou diminui?
3. Pesquise como alterar o tamanho do bloco de criptografia do algoritmo. Realize experimentos alterando o tamanho do bloco e observe: o tempo de criptografia aumentou ou diminuiu?

4. Refaça os exercícios anteriores com arquivos grandes, de muitos MB ou até mesmo GB. Observe o tempo de criptografia.

Conclusão

Parabéns, você aprendeu a usar o openssl para criptografar e descriptografar arquivos de texto no terminal linux. Você pode usar essa ferramenta para proteger seus dados sensíveis ou confidenciais de olhares indiscretos. Lembre-se de escolher um algoritmo seguro e uma senha forte, e de não compartilhar sua senha com ninguém. Espero que você tenha gostado desta atividade. Até a próxima! 👋