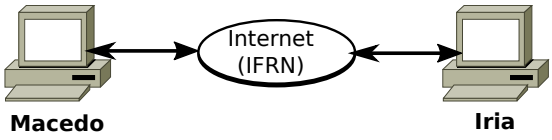
INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Professor: Macêdo Firmino
Disciplina: Segurança de Rede
Prática 13: Criptografia (Ccrypt e GPG) e Função Hash (SHASum).

Olá, meus alunos!! Como é que vocês vão? Na aula de hoje iremos aprender sobre criptografia e função hash na prática. Conheceremos a sua finalidade, principais métodos e ferramentas que implementam algoritmos de criptografia e hash no Linux. Vamos lá!!! Preparados???

Configurando o Ambiente

Para estudarmos estes conceitos e ferramenta iremos utilizar duas máquinas virtuais. Uma para transmitir os segredos (Iria) e outra para receber os segredos (Macedo). Podemos utilizar qualquer distribuição Linux. Nos nossos testes utilizei Kali Linux (máquina de Iria) e uma máquina Ubuntu (máquina Macedo), ambas configuradas em Rede Nat.



Criptografia

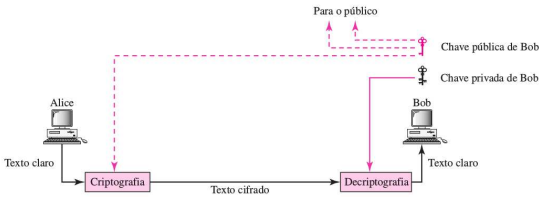
A criptografia é uma técnica utilizada há milênios e que atualmente faz parte do nosso cotidiano oferecendo soluções eficazes no que diz respeito à segurança da informação. Ela é uma ferramenta de segurança amplamente utilizada nos meios de comunicação e consiste basicamente na transformação de determinado informação a fim de ocultar seu real significado.

Existem atualmente duas abordagens para criptografia, as simétricas e as assimétricas. A criptografia simétrica foi o primeiro tipo de criptografia criado. Os algoritmos que a utilizam têm como característica principal o uso de uma mesma chave criptográfica para criptografar ou descriptografar uma informação. Sem a chave, não é possível decifrar a informação recebida.

A criptografia assimétrica, também denominada como criptografia de chave pública, possui como característica básica o uso de duas chaves ao invés de uma, sendo elas:

- Chave pública: chave que pode ser distribuída para outros usuários.
- Chave privada: deve ser mantida em segredo.

Neste modelo de criptografia, o texto cifrado pela chave pública somente poderá ser decifrado com a chave privada.



Na sequência iremos utilizar algumas ferramentas que implementam algoritmos de criptografia para vermos o funcionamento da criptografia na prática.

Chave Simétrica com Ccrypt

O ccrypt é um programa utilitário de criptografia de chave simétrica no Linux. Ele é baseado no AES (cifra de Rijndael) e pode ser utilizado para manter seus arquivos seguros confidenciais, podendo codificar arquivos e/ou pastas.

Vamos utiliza-lo para criptografar um arquivo para mandar pela internet para o destinário realizar a decriptografia e realizar a leitura. Para isso, instale o programa ccrypt usando o comando:

```
sudo apt update

sudo apt install ccrypt
```

Criptografando o Segredo (Iria)

Crie um arquivo de texto para ser codificado. Você poderá utilizar o seu editor preferido. Abaixo está um exemplo de criação usando o comando echo.

```
echo Invasão amanhã as 8h > secreto.txt
```

Utilize o programa ccrypt parar criptografar o arquivo:

```
ccrypt secreto.txt
```

Será solicitado que você informe uma senha e depois que você confirme a senha. Podemos utilizar a senha "ifrn". Na sequência será criado o arquivo secreto.txt.cpt e apagado o arquivo original secreto.txt. Com um visualizador abra o arquivo.

```
cat secreto.txt.cpt
```

Envie o arquivo criptografado para um amigo da turma. Como kali Linux já vem com o servidor Apache configurado e rodando. Iremos utiliza-lo para enviar o arquivo para o usuário alvo. Para isso, copie o arquivo criptografado para a pasta /var/www através do comando:

```
cp secreto.txt.cpt /var/www/html/
```

Em seguida, certifique-se de que o servidor web seja iniciado por meio do comando:

```
service apache2 start
```

Decriptografando o Segredo (Macedo)

Agora no destino (amigo da turma) abra um navegador e digite a URL `http://IP_KALI/secreto.txt.cpt`, que corresponde ao endereço IP da sua máquina Kali e o executável. Faça o *download* do arquivo.

Depois entre em contato com o destinatário e forneça a senha.

O usuário ifrn deverá decriptografar o arquivo usando o comando:

```
ccrypt -d secreto.txt.cpt
```

O arquivo secreto.txt irá surgir, agora basta ler o conteúdo do arquivo.

```
cat secreto.txt
```

Chave Pública com GPG

A ferramenta GPG no Linux fornece criptografia e assinatura digital usando o padrão OpenPGP, desenvolvido em 1991 por Philip Zimmermann. Ele permite encriptar os dados, assim somente o destinatário terá acesso aos dados, adicionalmente poderá verificar se a origem dos dados é confiável (através da assinatura digital).

O GPG se baseia no conceito de chave pública e privada dos algoritmos RSA, El Gamal e DSA. O algoritmos RSA é o padrão. As chaves públicas e privadas são armazenadas no diretório ∼/.gnupg. Iremos criar as chaves públicas e privadas, exportar e enviar a chave pública para o transmissor. O transmissor, por sua vez, irá criar um arquivo de texto com uma mensagem secreta, criptografar o arquivo com a chave pública do receptor e enviá-la. O receptor irá receber o texto criptografado e realizar a decodificação para ter acesso a mensagem original.

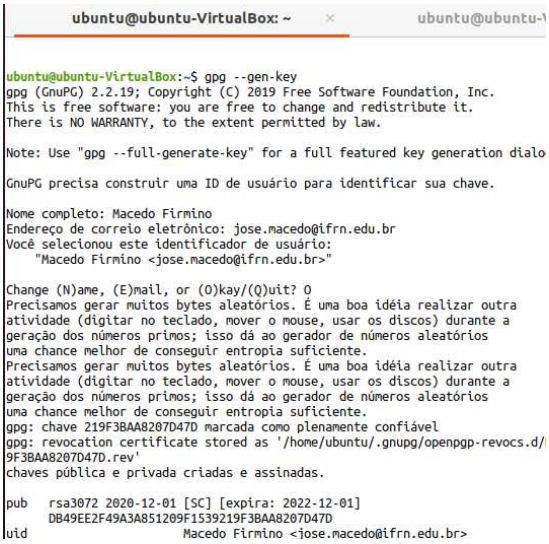
O Gnupg já vem instalado na maioria das distribuições Linux. Mas caso, a sua distribuição não possua realize a instalação utilizando os comandos:

```
apt install gnupg
```

Gerando as Chaves no Receptor (Macedo)

Para criar um par de chaves (pública e privada) com o algoritmos RSA utilize o comando:

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



Será solicitado que você informe o seu nome completo e o seu e-mail que será utilizado como identificador. Depois será solicitado que você insira esta senha para ter acesso a chave privada. Usaremos a senha **ifrn**. Será gerada uma chave de 3072 bits. Esta chave terá validade de 2 anos.

As informações das chaves geradas podem ser visualizadas pelo comando:

```
sudo gpg --list-key
```

Agora iremos exportar a chave pública de Macedo para enviá-las para as pessoas que desejem mandar mensagens codificadas para ele. Para isso, utilize o comando:

```
gpg -o chave_publica.txt
 --export jose.macedo@ifrn.edu.br
```

Será gerado o arquivo chave_publica.txt com a chave pública de Macedo. Depois, o transmissor deverá enviar este arquivo para aqueles que desejam enviar dados criptografados para ele.

Utilizaremos novamenteo apache para enviar o arquivo para o usuário alvo. Para isso, copie o arquivo com a chave para a pasta /var/www através do comando:

```
cp chave_publica.txt /var/www/html/
```

O Transmissor Importando a chave Pública (Iria)

Agora no transmissor (amigo da turma) abra um navegador e digite a URL `http://IP_KALI/chave_publica.txt`, que corresponde ao endereço IP da sua máquina Kali e o executável. Faça o *download* do arquivo.

Como Iria irá enviar mensagens criptografadas para Macedo, a mesma precisará importar a chave pública de Macedo. Para isso, utilizará o comando:

```
gpg --import chave_publica.txt
```

Gerando as Chaves do Transmissor (Iria)

No computador do transmissor, crie um par de chaves (pública e privada) com o algoritmos RSA utilize o comando:

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

Informe o seu nome completo do transmissor e o e-mail que será utilizado como identificador das chaves. Depois que insira a senha (por exemplo, ifrn) para ter acesso posteriormente a chave privada.

Criptografando no Transmissor (Iria)

Agora o transmissor irá criar um arquivo de texto contendo uma mensagem secreta para o receptor (Macedo). Utilize o editor da sua preferência. Abaixo um exemplo de geração do arquivo com o comando echo.

```
echo Invasão amanhã as 8h > secreto.txt
```

Na sequência Iria irá criptografar o arquivo secreto.txt com o comando:

```
gpg -r iria@ifrn.edu.br
 -u jose.macedo@ifrn.edu.br
 -e secreto.txt
```

Onde iria@ifrn.edu.br e jose.macedo@ifrn.edu.br são os identificadores dos usuário transmissor e destinatário, respectivamente. Será gerado um arquivo criptografado secreto.txt.gpg. Este arquivo poderá ser enviado para o destinatário pela Internet (por exemplo, utilizando o netcat). Se quiser pode abrir o arquivo e verificar o seu conteúdo usando o comando:

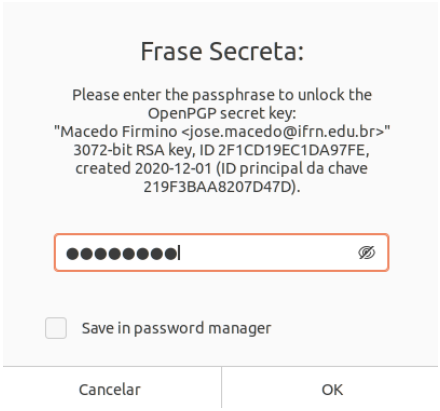
```
cat secreto.txt.gpg
```

Decriptografando no Receptor (Macedo)

Após o receptor receber o arquivo criptografado de Iria, ele poderá utilizar a sua chave privada para decriptografar a mensagem. Para isso, ele pode utilizar o comando:

```
gpg -d secreto.txt.gpg
```

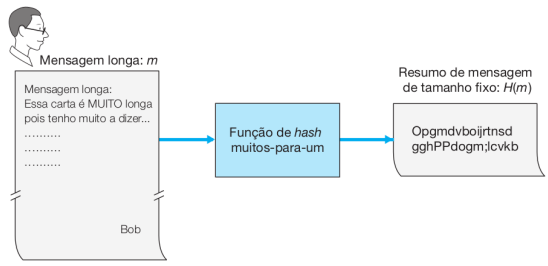
Será solicitado que forneça a senha para ter acesso a chave privada. Esta senha é a mesma utilizada no processo de criação das chaves.



Ao final a mensagem original em texto limpo será mostrada no terminal.

Função Hash

A função Hash (Resumo) é uma classe de algoritmo que mapeie dados grandes e de tamanho variável para pequenos dados de tamanho fixo. Por esse motivo, as funções Hash são conhecidas por resumirem o dado. Estas funções são unidirecionais, ou seja, não é possível recuperar o dado original a partir do resumo gerado. Dessa forma, as funções Hash são largamente utilizadas para buscar elementos em bases de dados, verificar a integridade de arquivos baixados ou armazenar e armazenar senhas de usuários.



Na utilização em bases de dados, a busca pela informação é feita pelo resumo e não pelo dado. Em geral, calcula-se o resumo do dado e, com o resumo, sabe-se alocalização da informação na base de dados. Na verificação de integridade, aplica-se a função Hash diretamente sobre o dado e salva-se o resumo gerado. Após o dado ser transmitido para o receptor, este calcula o resumo sobre o dado recebido e obtém um novo resumo. Se os resumos forem iguais, assume-se que o dado é igual. Se forem diferentes, recomenda-se um novo download do arquivo.

Para o armazenamento da senha de forma segura, usa-se um procedimento semelhante. No servidor, quando o usuário insere a senha, calcula-se a função Hash da senha e o mesmo é armazenado. Quando o usuário fornecer novamente a senha, o servidor calculará a função Hash novamente e irá comparar com o resumo armazenado. Se os resumos forem iguais, o usuário é autenticado.

Os principal algoritmo de SHA é o *Secure Hash Function* (SHA). Ele foi projetada pela Agência de Segurança Nacional dos Estados Unidos e é a mais utilizado. As versão mais comuns são: SHA, SHA-1 e SHA-2. O SHA-1 é um hash de 160 bits. SHA-2 é na verdade uma "família" de hashes (SHA224, SHA256, SHA384 e SHA512) e vem em uma variedade de comprimentos, sendo o mais popular de 256 bits.

SHASum

O SHASum é uma família de programa que calcula e verifica as funções hashes SHA-1 e SHA-2. Ele é instalado por padrão na maioria das distribuições Linux. Os comandos são: shasum, sha224sum, sha256sum, sha384sum e sha512sum.

O comando shaSum calcula o resumo da mensagem de um arquivo. Isso permite que ele seja comparado ao resumo original da mensagem para verificarmos se o arquivo não foi modificado.

Iremos criar um arquivo de texto, calcular o seu hash, realizar alterações no arquivo e calcular novamente. Depois iremos enviar o arquivo e o hash para um amigo da turma, por exemplo por e-mail. O seu amigo deverá receber o arquivo, calcular o hash e verificar se os dados não foram alterados na transmissão.

Inicialmente iremos criar o arquivo. Você pode utilizar o editor de sua preferência, abaixo será mostrado um exemplo.

```
echo Ganhou um ponto na nota > msg.txt
```

O sha1sum produz um valor de hash de 160 bits (20 bytes) do arquivo, através do comando:

```
sha1sum msg.txt
```

Como resultado obtivemos:

```
6eea6ea93404069efb3dbeaa23e2ec230234c1f6
```

Agora iremos renomear o arquivo e calcularmos novamente a função Hash.

```
mv msg.txt msg2.txt

sha1sum msg.txt
```

Como resultado obtivemos:

```
6eea6ea93404069efb3dbeaa23e2ec230234c1f6
```

Observe que alterando os atributos do arquivo, por exemplo, nome, permissão e localização o resultado do hash não se altera.

Agora iremos editar e alterar as informações contidas no arquivo e recalcular o hash.

```
echo do 2º bimestre >> msg2.txt
sha1sum msg2.txt
```

Como resultado obtivemos um hash diferente visto que a mensagem é diferente.

```
601fde733e801e93dd365a742ffc7fc9745a1c7e
```

Temos ainda os comandos SHA256sum (calcula o Hash SHA com 256 bits de saída) e sha512sum (calcula o Hash SHA com 512 bits de saída). Quanto maior a saída mais difícil da realização de ataques na função hash.

Transmitindo o Arquivo e o Hash (kali)

Quando formos enviar um arquivo que desejam garantir a sua integridade, devemos disponibilizar também o seu Hash. Desta forma, na máquina Kali iremos gerar uma mensagem e seu respectivo hash. Depois iremos enviar para o destinatário. De modo que, ele possa obter a mensagem original, calcular o hash e determinar se a mensagem está correta.

Para isso, utilize os comandos:

```
echo Aula de Redes 2023 > mensagem.txt

sha1sum mensagem.txt > hash.sha1
```

Utilizaremos novamenteo apache para enviar o arquivo para o usuário alvo. Para isso, copie o arquivo com a chave para a pasta /var/www através do comando:

```
cp mensagem.txt /var/www/html/

cp hash.sha1 /var/www/html/
```

Recebendo e Verificando a Integridade do Arquivo

Agora no destino (maquina Ubuntu) abra um navegador e digite a URL `http://IP_KALI/mensagem.txt` e depois `http://IP_KALI/hash.sha1`, que corresponde ao endereço IP da sua máquina Kali e os arquivos. Faça o *download* dos arquivos.

Agora iremos verificar a integridade do arquivo recebido. Para isso coloque os arquivos (mensagem.txt e hash.sha1) na mesma pasta e digite o comando:

```
sha1sum -c mensagem.txt.sha1
```

Se o arquivo estiver integro irá aparecer o nome do arquivo e informar o resultado, da seguinte forma:

```
mensagem.txt: SUCESSO
```

Atividades

1. Forme duplas e troquem mensagens criptografadas e assinadas utilizando as ferramentas Ccrypt, GPG e shasum.