Laboratório de Criptografia – OpenSSL

Sobre o OpenSSL: Secure Sockets Layer é um protocolo da camada de aplicação que foi desenvolvido pela Netscape Corporation com o propósito de transmitir informações sigilosas, como detalhes de Cartão de Crédito via Internet. SSL funciona usando uma chave privada para encriptar os dados transferidos pela conexão com SSL habilitado, frustrando assim a interceptação da informação. O mais popular uso do SSL está na conjunção de visualização web com HTTP, mas muitas aplicações de rede podem se beneficiar do uso do SSL. Convencionalmente, URLs que requerem uma conexão SSL iniciam com https: em vez do http.

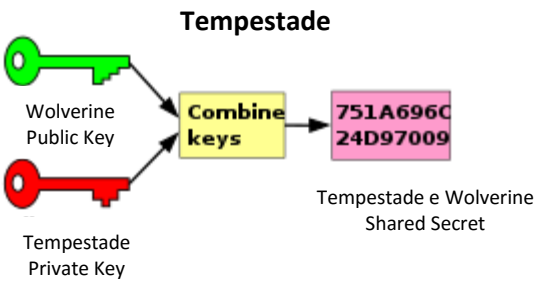
O OpenSSL: é uma robusta ferramenta de implementação do SSL de classificação comercial e relacionada a biblioteca de propósito geral baseada em SSLeay, desenvolvida por Eric A. Young e Tim J. Hudson. OpenSSL está disponível como um equivalente Open Source para implementação comercial do SSL através de estilo de licença Apache.

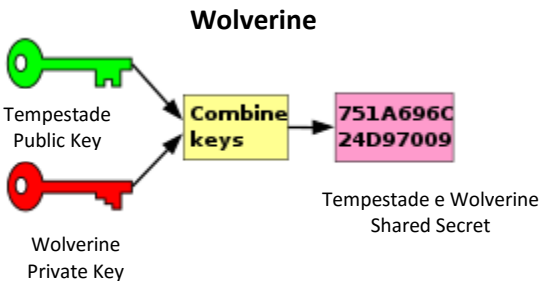
O Objetivo dessa prática é explorar os tipos de criptografia existentes. E não interceptar a mensagem.

Cenário:

Geração de Chaves Tempestade
Pública e Privada

Geração de Chaves Wolverine
Pública e Privada

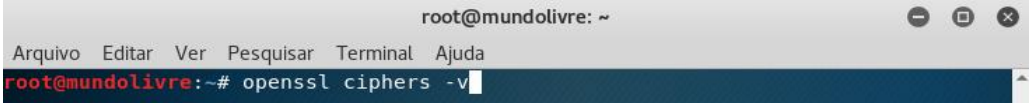




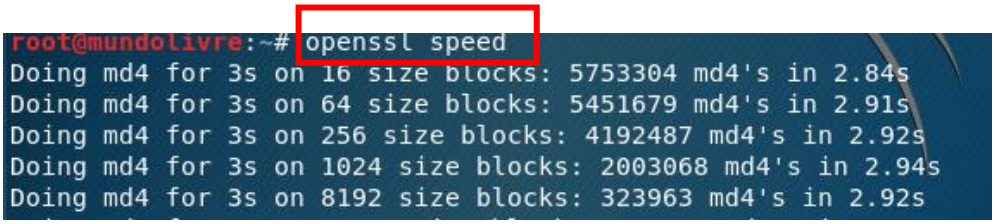
Atividade 01 – Conhecendo o OpenSSL

1. Verificar versão do SSL:

2. Lista todas os algoritmos disponíveis:

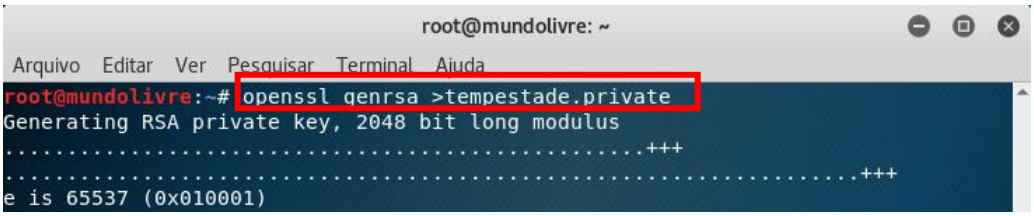


3. Apresenta um benchmark do seu computador em relação à velocidade do processamento de cada algoritmo: (Esse processo fica analisando a capacidade de execução de cada algoritmo.



Atividade 02 – Geração de Chave Tempestade

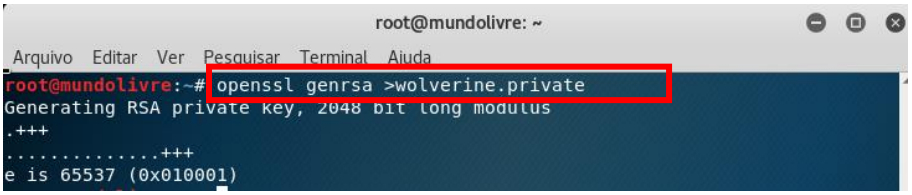
1. Em um novo terminal, crie a Chave privada de Tempestade:



2. Chave pública de Tempestade:

Atividade 03 – Geração de Chave Wolverine

1. Chave privada de Wolverine:



2. Chave pública de Wolverine:

Atividade 04 – Criação da Mensagem / Adicionando Algoritmo de Criptografia

1. Tempestade cria a mensagem:

2. Gera MD5 da mensagem:

```
root@mundolivre:~# openssl dgst -md5 mensagem.txt
MD5(mensagem.txt)= 6d733834b4fa8416aa66bb8a2377142b
```

3. Tempestade encripta a mensagem usando a chave pública de Wolverine (wolverine.public)

```
root@mundolivre:~# openssl rsautl -encrypt -in mensagem.txt -out mensagem.encrypted -pu
bin -inkey wolverine.public
```

4. Wolverine decripta a mensagem de Tempestade usando a chave privada dele:

```
root@mundolivre:~# openssl rsautl -decrypt -in mensagem.encrypted -out mensagem.decrypted -inkey
wolverine.private
```

5. Wolverine envia a resposta para Tempestade:

```
root@mundolivre:~# echo "Quero sim" >mensagem.txt
```

6. Wolverine assina a mensagem com sua chave privada (wolverine.private):

```
root@mundolivre:~# openssl rsautl -sign -in mensagem.txt -out mensagem.signed -inkey wolverine.pr
ivate
```

7. Tempestade verifica a mensagem usando a chave pública de Wolverine (wolverine.public):

```
root@mundolivre:~# openssl rsautl -verify -in mensagem.signed -out mensagem.verified -pubin -inke
y wolverine.public
```

8. Mensagem secreta de Tempestade para Wolverine (Local do Café):

```
root@mundolivre:~# echo "Local do Café" >mensagem.txt
```

9. Tempestade gera uma chave randômica que será usada na encriptação da mensagem:

```
root@mundolivre:~# openssl rand -out aleat.key -base64 32
```

Gera um arquivo chamado aleat.key contendo 32bytes de dados aleatórios codificados em base64 (usando somente caracteres imprimíveis).

10. Tempestade encripta a mensagem com a chave aleatória gerada:

```
root@mundolivre:~# openssl des3 -e -kfile aleat.key -in mensagem.txt -out mensagem.encrypted
```

11. Tempestade cria um digest da mensagem para assinar:

```
root@mundolivre:~# openssl dgst -binary mensagem.txt >mensagem.digest
```

12. Tempestade assina o digest (hash) com sua chave privada:

```
root@mundolivre:~# openssl rsautl -sign -in mensagem.digest -out digest.signed -inkey tempestade.
private
```

13. Tempestade encripta a chave randômica com a chave pública de Wolverine:

```
root@mundolivre:~# openssl rsautl -encrypt -in aleat.key -out key.encrypted -pubin -inkey wolveri
ne.public
```

Atividade 05 – Verificação da Chave

1. Wolverine verifica o Hash assinado por Tempestade usando a chave pública dela:

```
root@mundolivre:~# openssl rsautl -verify -in digest.signed -out mensagem.digest1 -pubin -inkey
 tempestade.public
```

2. Wolverine calcula o hash da mensagem:

```
root@mundolivre:~# openssl dgst -binary mensagem.txt >mensagem.digest2
```

3. Wolverine compara os dois hash:

```
root@mundolivre:~# diff mensagem.digest1 mensagem.digest2
```

Nessa prática você apenas navegará sobre as principais criptografias, RSA, MD5, DES3. Não terá nenhuma transmissão de dados. Apenas uma simulação de como os dados são encripatados de forma automática em uma transmissão de dados criptografada.

Simulando uma conversa entre duas pessoas.