Laboratório - Criptografando e descriptografando dados usando OpenSSL

Objetivos

Parte 1: Criptografando mensagens com OpenSSL

Parte 2: Descriptografando mensagens com OpenSSL

Histórico/Cenário

O OpenSSL é um projeto de código aberto que fornece um kit de ferramentas robusto, de nível comercial e completo para os protocolos TLS (Transport Layer Security) e Secure Sockets Layer (SSL). É também uma biblioteca de criptografia de uso geral. Neste laboratório, você usará o OpenSSL para criptografar e descriptografar mensagens de texto.

Observação: Embora o OpenSSL seja a biblioteca de criptografia de fato hoje, o uso apresentado neste laboratório NÃO é recomendado para proteção robusta. Abaixo estão dois problemas de segurança com este laboratório:

1) O método descrito neste laboratório usa uma função de derivação de chave fraca. A ÚNICA segurança é introduzida por uma senha muito forte.
2) O método descrito neste laboratório não garante a integridade do arquivo de texto.

Este laboratório deve ser usado apenas para fins de instrução. Os métodos aqui apresentados NÃO devem ser usados para proteger dados verdadeiramente sensíveis.

Recursos necessários

- Máquina virtual CyberOps Workstation

Instruções

Parte 1: Criptografando mensagens com OpenSSL

OpenSSL pode ser usado como uma ferramenta autônoma para criptografia. Embora muitos algoritmos de criptografia possam ser usados, esse laboratório se concentra no AES. Para usar o AES para criptografar um arquivo de texto diretamente da linha de comando usando o OpenSSL, siga as etapas abaixo:

Etapa 1: Criptografando um arquivo de texto

a. Log into CyberOPS Workstation VM.

b. Open a terminal window.

c. Como o arquivo de texto a ser criptografado está no diretório /home/analyst/lab.support.files/, mude para esse diretório:

```
[analyst@secOps ~]$ cd ./lab.support.files/
[analyst@secOps lab.support.files]$
```

d. Digite o comando abaixo para listar o conteúdo do arquivo de texto criptografado **letter_to_grandma.txt** na tela:

```
[analyst@secOps lab.support.files]$ cat letter_to_grandma.txt
Oi vovó,
```

```
Estou escrevendo esta carta para agradecer pelos biscoitos de chocolate que
você me enviou. Comprei-os esta manhã e já comi metade da caixa! Eles são
absolutamente deliciosos!

Desejo-lhe tudo de bom. Amor,
Seu neto comedor de biscoitos.
[analyst@secOps lab.support.files]$
```

e. Na mesma janela de terminal, execute o comando abaixo para criptografar o arquivo de texto. O comando usará AES-256 para criptografar o arquivo de texto e salvar a versão criptografada como **message.enc**. O OpenSSL pedirá uma senha e confirmação de senha. Forneça a senha conforme solicitado e lembre-se da senha.

```
[analyst @secOps lab.support.files] $ openssl aes-256-cbc -in
letter_to_grandma.txt -out message.enc
digite a senha de criptografia aes-256-cbc:
Verificando - digite a senha de criptografia aes-256-cbc:
[analyst@secOps lab.support.files]$
```

Documente a senha.

f. Quando o processo for concluído, use o comando **cat** novamente para exibir o conteúdo do arquivo **message.enc**.

```
[analyst @secOps lab.support.files] $ cat message.enc
```

O conteúdo do arquivo **message.enc** foi exibido corretamente? O que é que se parece? Explique.

g. Para tornar o arquivo legível, execute o comando OpenSSL novamente, mas desta vez adicione a opção **-a**. A opção **-a** diz ao OpenSSL para codificar a mensagem criptografada usando um método de codificação diferente do Base64 antes de armazenar os resultados em um arquivo.

Nota: Base64 é um grupo de esquemas de codificação binária a texto semelhantes usados para representar dados binários em um formato de string ASCII.

```
[analyst @secOps lab.support.files] $ openssl aes-256-cbc -a -in
letter_to_grandma.txt -out message.enc
digite a senha de criptografia aes-256-cbc:
Verificando - digite a senha de criptografia aes-256-cbc:
```

h. Mais uma vez, use o comando **cat** para exibir o conteúdo do arquivo **message.enc**, agora regerado:

Nota: O conteúdo de **message.enc** irá variar.

```
[analyst@secOps lab.support.files]$ cat message.enc
U2FsdGVkX19ApWyrn8RD5zNp0RPCuMGZ98wDc26u/vmj1zyDXobGQhm/dDRZasG7
rfnth5Q8NHValEw8vipKGM66dNFyyr9/hJUzCoqhFpRHgNn+Xs5+TOtz/QCPN1bi
08LGTSzOpfkg76XDCk8uPy1hl/+Ng92sM5rgMzLXfEXtaYe5UgwOD42U/U6q73pj
a1ksQrTWsv5mtN7y6mh02Wobo3A1ooHrM7niOwK1a3YKrSp+ZhYzVTrtksWDl6Ci
XMufkv+FOGn+SoEEuh7l4fk0LIPEfGsExVFB4TGdTiZQApRw74rTAZaE/dopaJn0
```

```
sJmR3+3C+dmgzZIKEHWsJ2pgLvj2Sme79J/XxwQVNpw=
[analyst@secOps lab.support.files]$
```

Message.enc é exibido corretamente agora? Explique.

Você pode pensar em um benefício de ter **message.enc** codificado Base64?

Parte 2: Descriptografando mensagens com OpenSSL

Com um comando OpenSSL semelhante, é possível descriptografar **message.enc**.

a. Use o comando abaixo para descriptografar message.enc:

```
[analyst @secOps lab.support.files] $ openssl aes-256-cbc —a -d -in
message.enc -out decrypted_letter.txt
```

b. O OpenSSL pedirá a senha usada para criptografar o arquivo. Enter the same password again.

c. Quando o OpenSSL terminar de descriptografar o arquivo **message.enc**, ele salva a mensagem descriptografada em um arquivo de texto chamado **decrypted_letter.txt**. Use o **gato** exibir o conteúdo de **decrypted_letter.txt**:

```
[analyst @secOps lab.support.files] $ cat decrypted_letter.txt
```

A carta foi descriptografada corretamente?

O comando usado para descriptografar também contém uma opção. Você pode explicar?