Pontifícia Universidade Católica
Campinas

Engenharia de Computação

Sistemas Operácionais B

Relatório Experimento 1

Author:
Bruno Camilo Silvério ..... RA: 16080293
Guilherme Soderi Pernicone ..... RA: 16085037
João Pedro Porta ..... RA: 16039778
Marcelo Dib Coutinho ..... RA: 16023673
Pedro Garcia Pierina ..... RA: 16136293

Supervisor:
Ms. Prof. Edmar Roberto Santana de Resende

October 22, 2018

Abstract

Experimento com o objetivo de aprender e entender as dificuldades de criar e colocar em uso um modulo de Kernel do sitema operácional Ubuntu 16.04.

1 Introdução

Nesse experimento, utilizando Ubuntu 16.04, foi criado um módulo que tem como objetivo utilizar a biblioteca crypto.h para cryptografar, decriptografar uma string dada pelo usuário atravéz de um programa de usuário, escrito em C, o programa de usuário comunica com o módulo atravéz da escrita e leitura no modulo /dev/cryptomodule

2 Fundamentação Teórica

2.1 Sistema Operacional (SO)

Um sistema operacional (SO) funciona como uma interface entre os usuários e o hardware, ele é um programa cuja a função é gerenciar os recursos do sistema, definindo quais programas devem receber a atenção do processador e é responsável por criar um sistema de arquivos a fim de indexar a memória. Além disso o SO também é responsável por fazer a comunicação entre os programas, periféricos e o hardware da máquina.

2.2 Kernel

O kernel ou núcleo é o componente central do sistema operacional. Ele serve como ponte entre os aplicativos e o processamento real dos dados feito a nível de hardware. O kernel é responsável por gerenciar os recursos dos sistemas e oferecendo ajuda para que os aplicativos (softwares) utilizem esses recursos. Para isso ele é encarregado em fazer comunicação entre componentes de hardware e software, oferecendo uma camada de abstração de nível mais baixo para os recursos que os aplicativos devem controlar para executar suas ações.

2.3 Linux

O linux é um sistema operacional de código fonte livre, sendo assim disponível para qualquer um utilizar e modificar, respeitando os termos de contrato. O linux foi criado baseando-se no sistema operacional Unix, pelo programador finlandês Linus Torvalds. Os sistemas operacionais linux são caracterizados por possuírem o núcleo kernel linux em suas maquinas.

2.4 Driver

É um desafio fornecer uma definição única e precisa para o termo driver. No sentido mais simples, um driver é um software que permite que o sistema operacional e um dispositivo se comuniquem um com o outro. Portanto, um driver é um software que traduz o que diz um hardware ou um dispositivo para que o computador possa entender. Sem um software de driver, o hardware conectado (por exemplo, uma placa de vídeo ou impressora) não funcionará corretamente.

2.5 Módulos

Em computação, um módulo carregável do núcleo, é um arquivo objeto que contém código para estender o núcleo em execução, ou o chamado núcleo base, de um sistema operacional. Os módulos são normalmente usados para adicionar suporte para novos hardwares (como controladores de dispositivos) e/ou sistemas de arquivos, ou para adicionar chamadas de sistema. Quando a funcionalidade fornecida por um módulo não for mais necessária, ela pode ser descarregada com o objetivo de liberar memória e outros recursos.

2.6 Electronic codebook (ECB)

O modo mais simples de criptografia é o electronic codebook (ECB). A mensagem é dividida em blocos e cada bloco é criptografado separadamente. A desvantagem deste método é que blocos idênticos de texto plano são criptografados em blocos de texto cifrado idênticos; assim, ele também não oculta padrões de dados. No geral, não oferece uma perfeita confidencialidade de mensagem, e não é recomendado para uso em protocolos criptográficos em geral. Eis aqui um bom exemplo do nível no qual o ECB pode transformar os padrões de dados de texto simples em texto cifrado.

2.7 Advanced Encryption Standard (AES)

AES é uma criptografia de blocos com chave simétrica (cifra de bloco) no caso o AES-128 foi utilizado, dessa forma trabalha com o sistema de blocos de 16 Bytes. É possível utilizar valores de entrada menores sem problemas, mas maiores será necessário dividir em blocos de 16B.

2.8 Hash

Funções hash criptográficas são operações matemáticas tendo um algoritmo de uma via, ou seja, é irreversível. É muito usado com senhas da seguinte forma: Primeiramente é gerado um Hash da senha e este, será apenas comparado ao Hash armazenado no destino. Caso os Hash's sejam iguais, logo a senha é igual. Por exemplo: MD5, SHA-1, SHA-2.

2.9 SHA-256

SHA-256 e SHA-512 são funções Hash inovadoras computadas com palavras de 32 bytes e 64 bytes, respectivamente. Eles usam quantidades de deslocamento e constantes aditivas diferentes, mas as suas estruturas são praticamente idênticas, diferindo apenas no número de rodadas.

3 O que foi desenvolvido

3.1 Progama de usuário em C

O programa em C possue um menu em que o usuário tem a opção de, encriptar uma string, encriptar escrevendo os bytes em hexadecimal, e decriptando utilizando tambem os valores dos bytes em hexadecimal, o programa então faz as escrita no módulo /dev/cryptomodule, inserindo no primeiro byte da string uma letra equivalente a operação a ser realizada, para então poder ler do mesmo módulo. O programa, também, se selecionado para encriptar um string comum, faz a conversão de uma string para um string com os valores originais representados em hexadecimal.

3.2 Módulo

O módulo, acionado após uma escrita em /dev/cryptomodule, recebe a string que o usuário proporcionou, e dependendo da primeira letra da string, faz a operação requisitada pelo o usuário, que pode ser, encriptação, desencriptação ou hash da string, e o modulo faz essas operações utilizando a biblioteca crypto.h.

4 Desafios Encontrados

Durante o processo de desenvolvimento foram encontrados varios desafios:

1. Comunicação entre programa de usuário e módulo
2. Trasformação de uma string com caracteres hexadecimais para sua sua representação original
3. Utilização da biblioteca crypto
4. Scatterlist e como utiliza-las

4.1 Comunicação entre programa de usuário e módulo

Essa dificuldade foi facilmente superada após estudo de códigos prontos disponibilizados pelo professor. Utilizamos a função write e read em C, aprendemos sobre essas funções durante a execução do experimento, pois até o momento todos integrantes só havim utilizados as funções fwrite e fread, o conceito é parecido, entretanto fread e fwrite só são utilizados para arquivos.

4.2 Trasformação de uma string com caracteres hexadecimais para sua sua representação original

Para esse problema utilizamos manipulação de bits com a operação shifleft $<<$, e com com a subtração do caracter para o valor que ele representa utilizando os valores da *ASCII Table*.

Ex.:
4A ⇒ 00110100 01000001 : 2 caractéres
4Ah ⇒ 00000100 00001010 : Caractéres subtraídos para o valor original em hexadecimal
400Ah ⇒ 01000000 00001010 : O primero byte é shiftado 4 vezes
J / 4Ah ⇒ 00101010 : Os dois bytes são somados, gerando o valor original que os dois caractéres representavam

4.3 Utilização da biblioteca crypto

Esse problema não foi específico a uma função ou a uma lógica, mas sim a diferentes mecânismos necessários para a utilização do modulo que o tornou complicado de utilizar, foi possível sua utilização atravez de estudos dos códigos prontos e vários testes.

4.4 Scatterlist e como utiliza-las

Scatterlist é uma struct em que são armazenadas *pagenumber*, *offset* e *size* que são todas as informções para encontrar uma variável na memória, esse método de armazenar os dados de uma variável é muito utilizada em módulos de kernel.
Para resolver o problema de encontrar essa variável foi encontrado a função *sg_virt* que calcula o endereço da variável dando sua respectiva scatterlist, assim possibilitando a utilização dessa variável

5 Conclusão

Resultados dos experimentos

5.1 Programa de Usuário em C

No programa de usuário foi criado um menu com cinco opções:

1. Criptografar uma string.
2. Criptografar uma string em hexadecimal
3. Decriptografar uma string em hexadecimal
4. Gerar hash de uma string em hexadecimal
5. Sair

Criptografar uma string: Essa opção faz com que o programa converta a string em ASCII para sua conversão em hexadecimal. E, então, escreve a string convertida, com o primeiro caracter 'c', no módulo */dev/cryptomodule*.

Criptografar uma string em hexadecimal: Aqui o programa apenas escreve a string dada pelo usuário o módulo, também com o primeiro caracter 'c'.

Decriptografar uma string em hexadecimal: Nessa opção também é feita a escrita, entretanto com 'd' como primeiro caracter.

Gerar hash de uma string em hexadecimal: Aqui a escrita é feita direta, com um 'h' na primeira posição da string.

Sair: Sair do programa sem realizar nenhuma operação.

Em todas as opçõe, menos *Sair*, o resultado é impresso no final da execução.

5.2 Módulo cryptomodule

No módulo, utilizando várias funções específicas de Kernel, a primeira coisa que o módulo faz é reverter a string de entrada, escrita pelo programa de

usuário, então a string é tratada adicionando *Padding* quando necessário. Depois do tratamento, utilizando o primeiro caracter para designar a função, direcionamos a string tratada à função certa equivalente a opção escolhida pelo usuário.

O primerio problema que não foi possível contornar é quando a string dada pelo o usuário tem um tamanho real maior que 16 bytes. Já que esse é o tamanho do nosso bloco de encriptação, portanto não foi resolvido a necessidade de multiplas conversões.

Após à execução das funções necessárias o resultado delas é escrito novamente no módulo, para que o programa de usuário possa ler os resultado das operações.