Sistemas Operacionais B

Projeto #2

1. Introdução

Este projeto deverá permitir ao aluno familiarizar-se com os detalhes de implementação de chamadas de sistema (*system calls*) em um kernel Linux. Espera-se que ao final do projeto, cada aluno seja capaz de implementar, compilar, instalar e testar um kernel Linux modificado contendo duas novas chamadas de sistema que permitam a programas de usuário armazenar e ler arquivos de forma cifrada em um sistema Linux através do uso da API criptográfica do kernel.

2. Descrição do projeto

O projeto consiste em implementar em um kernel Linux duas novas chamadas de sistema: **write_crypt** e **read_crypt**.

A chamada de sistema **write_crypt** deve permitir que programas em espaço de usuário possam armazenar arquivos de forma cifrada, utilizando o algoritmo AES em modo ECB para cifrar os dados.

A chave simétrica usada para cifrar os dados deverá ser definida no código-fonte da chamada de sistema **write_cypt**, só podendo ser modificada através de uma recompilação do kernel.

O formato da chamada de sistema **write_crypt** é mostrado a seguir:

```
ssize_t write_crypt(int fd, const void *buf, size_t nbytes);
```

onde:

Argumento	Descrição
fd	Descritor de arquivos obtido através da chamada de sistema **open**. É um valor inteiro.
buf	Ponteiro para um vetor de caracteres, cujo conteúdo deve ser escrito no arquivo referenciado pelo descritor de arquivos **fd**.
nbytes	Número de bytes do vetor de caracteres apontado por **buf** que devem ser escritos no arquivo referenciado pelo descritor de arquivos **fd**.

O valor de retorno e os códigos de erro retornados pela chamada **write_crypt** devem seguir o mesmo modelo da chamada de sistema **write**, já existente em sistemas Linux.

A chamada de sistema **read_crypt** deve permitir que programas em espaço de usuário possam ler arquivos cifrados com a chamada **write_crypt**, utilizando o algoritmo AES em modo ECB para decifrar os dados lidos.

A chave simétrica usada para decifrar os dados deverá ser definida no código-fonte da chamada de sistema **read_cypt**, só podendo ser modificada através de uma recompilação do kernel.

O formato da chamada de sistema **read_crypt** é mostrado a seguir:

```
ssize_t read_crypt(int fd, void *buf, size_t nbytes);
```

onde:

Argumento	Descrição
fd	Descritor de arquivos obtido através da chamada de sistema **open**. É um valor inteiro.
buf	Ponteiro para um vetor de caracteres, no qual devem ser armazenados os bytes lidos do arquivo referenciado pelo descritor de arquivos **fd**.
nbytes	Número de bytes que devem ser lidos do arquivo referenciado pelo descritor de arquivos **fd** e armazenados no vetor de caracteres apontado por **buf**.

O valor de retorno e os códigos de erro retornados pela chamada **read_crypt** devem seguir o mesmo modelo da chamada de sistema **read**, já existente em sistemas Linux.

Repare que o processo de cifragem dos arquivos será transparente para os programas em espaço de usuário. Isso significa que ao gravar um arquivo no sistema de arquivos utilizando a chamada de sistema **write_crypt**, o conteúdo do arquivo será armazenado cifrado, mas ao ler o arquivo utilizando a chamada de sistema **read_crypt**, o conteúdo retornado será o conteúdo já decifrado.

Para testar o correto funcionamento das chamadas de sistema implementadas, devem ser implementados programas em espaço de usuário que permitam abrir um arquivo e realizar operações de escrita e leitura utilizando as chamadas de sistema **write_crypt** e **read_crypt**, além de programas em espaço de usuário que permitam abrir um arquivo e realizar operações de escrita e leitura utilizando as chamadas de sistema **write** e **read**, de forma que seja possível demonstrar as corretas cifragem e decifragem dos dados.

3. Material complementar

Adding a Hello World System Call to Linux Kernel:
https://medium.com/anubhav-shrimal/adding-a-hello-world-system-call-to-linux-kernel-dad32875872

Implementing a system call in Linux Kernel 4.7.1:
https://medium.com/@ssreehari/implementing-a-system-call-in-linux-kernel-4-7-1-6f98250a8c38

Anatomy of a system call, part 1:
https://lwn.net/Articles/604287/

Anatomy of a system call, part 2:
https://lwn.net/Articles/604515/

The Linux Kernel Documentation - Adding a New System Call:
https://www.kernel.org/doc/html/v4.14/process/adding-syscalls.html

Kernel command using Linux system calls:
https://developer.ibm.com/tutorials/l-system-calls/

Linux Kernel Crypto API:
https://www.kernel.org/doc/html/v4.12/crypto/index.html

The Linux file system:
http://www.tldp.org/LDP/tlk/fs/filesystem.html

Anatomy of the Linux file system:
https://www.ibm.com/developerworks/library/l-linux-filesystem/index.html

4. Resultado

O projeto deve ser acompanhado de um relatório com as seguintes partes obrigatórias:

- Introdução, indicando o que se pretende com o projeto;
- Detalhes de projeto das chamadas de sistema desenvolvidas, detalhando através de textos e diagramas o funcionamento interno das chamadas de sistema e dos algoritmos criptográficos utilizados;
- Descrição das modificações realizadas no código-fonte do kernel Linux;

- Descrição dos programas em espaço de usuário implementados;
- Descrição dos resultados obtidos, detalhando o processo de compilação, instalação e teste de tudo que foi desenvolvido, demonstrando as funcionalidades implementadas através de imagens e textos descrevendo o que está sendo testado, além dos resultados esperados e dos resultados obtidos;
- Conclusão indicando o que foi aprendido com o experimento.

Entrega

A entrega do projeto deve ser feita em seu escaninho no AVA, em uma pasta com o nome “Projeto2”, de acordo com o cronograma previamente estabelecido.

Em todos os arquivos entregues deve constar **OBRIGATORIAMENTE** o nome e o RA dos integrantes do grupo.

Devem ser entregues os seguintes itens:

i. Relatórios de acompanhamento semanal individuais: cada membro do grupo deve preencher seus próprios relatórios de acompanhamento semanal descrevendo apenas as suas atividades desenvolvidas e as suas contribuições realizadas no código, com o número do commit correspondente;
ii. Códigos-fonte implementados;
iii. Relatório final do trabalho, em formato pdf.

Não serão aceitas entregas após a data definida. A não entrega acarreta em nota zero no experimento.

A interação entre os grupos é estimulada, no entanto qualquer tentativa de plágio será punida com a nota -Nmax para todos os envolvidos. Na dúvida do que é ou não plágio, consulte o docente.