ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет безопасности информационных технологий

Дисциплина:

«Операционные системы»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

Выполнили:

Нгуен Хонг Хань N3249

_____

(подпись)

Проверил:

Савков Сергей Витальевич

_____

(подпись)

Санкт-Петербург

2022г.

Задание

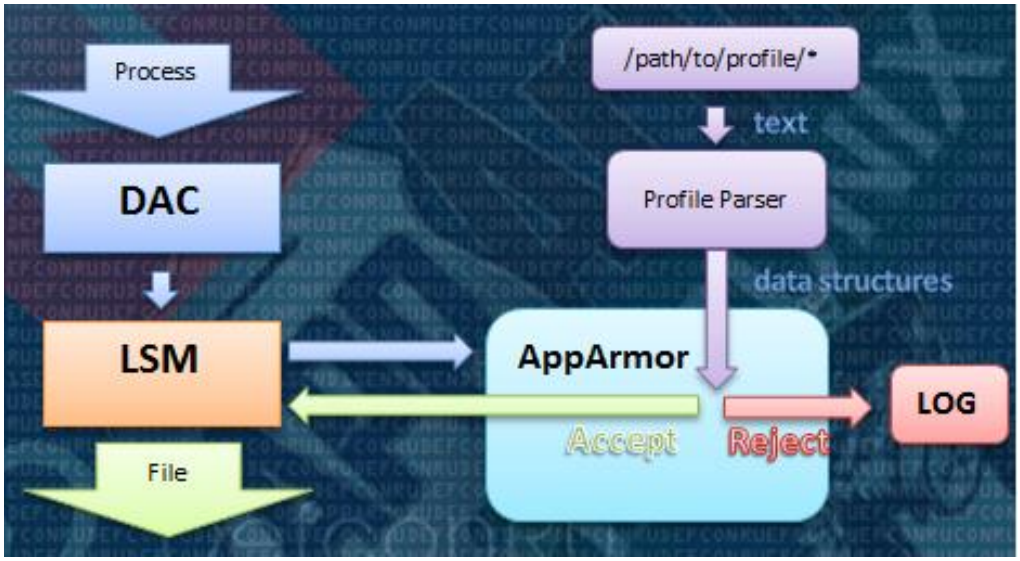
1. Настроить Apparmor для мониторинга сложного приложения и продемонстрировать его работу при ограниченных правах (оконное приложение или веб-сервер)
2. Настроить selinux в режиме мандатного доступа (CentOS и др.) и продемонстрировать работу в двухуровневой модели.

Усиленный вариант (или)

1. Придумать и написать свой LSM-модуль (сложная авторизация действий)
2. Придумать и написать свой PAM-модуль (сложная авторизация действий)

1. **Настроить Apparmor для мониторинга сложного приложения и продемонстрировать его работу при ограниченных правах (оконное приложение или веб-сервер).**

а. Apparmor



AppArmor - это реализация Модуля безопасности линукс по управлению доступом на основе имен. AppArmor ограничивает отдельные программы набором перечисленных файлов и возможностями в соответствии с правилами Posix 1003.1e.

- В режиме Enforce ядро гарантирует соблюдение правил, записанных в файле профиля. Нарушения не допускаются и соответствующая запись попадает в логи.
- В режиме Complain AppArmor лишь регистрирует нарушения, не блокируя при этом сами действия.

б. Ход работы

- Запустим контейнер и проверим может ли выполнить команду sh (всё работает)

```
hanh@ubuntu:~$ sudo docker run -ti ubuntu /bin/bash
```

```
root@4daf208bec57:/# sh
#
```

- **Сохраним профиль на диск в файле /etc/apparmor.d/containers/docker-nginx.**

```
deny /bin/dash mrwklx,
deny /bin/sh mrwklx,
deny /usr/bin/top mrwklx,
```

- **Загружаем профили AppArmor в ядро**

```
hanh@ubuntu:~$ sudo apparmor_parser -r -W /etc/apparmor.d/containers/docker-nginx
```

- **Запустим контейнер с профилем.**

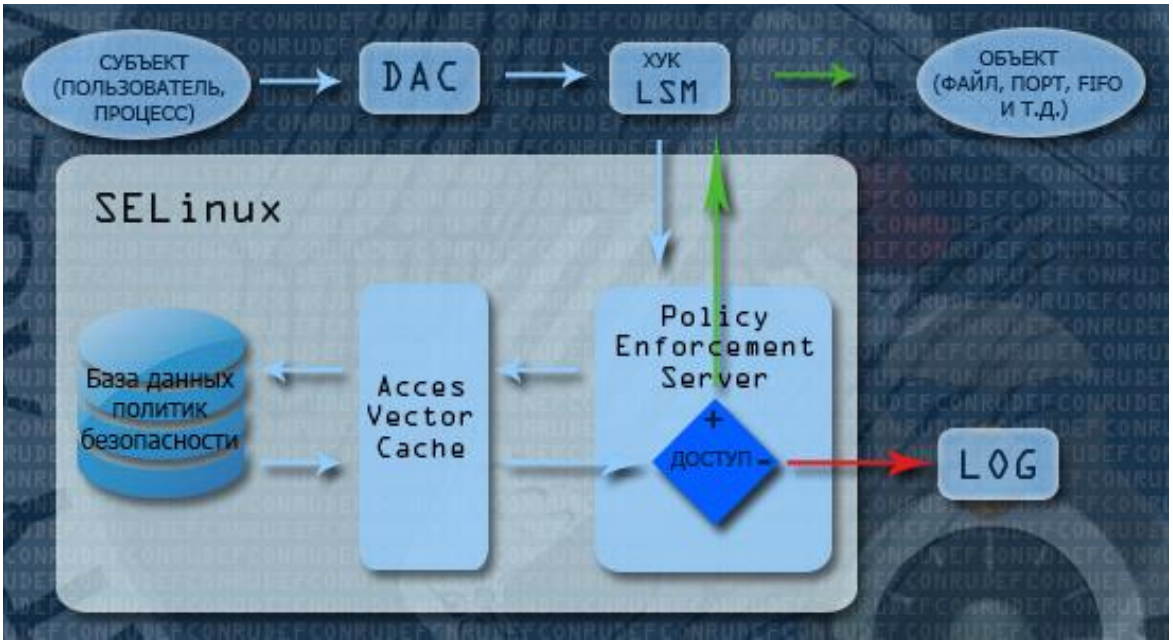
```
hanh@ubuntu:~$ sudo docker run --security-opt "apparmor=docker-nginx"      -p 80:80 -d
--name apparmor-nginx nginx
dafc48fded4190d7709849542ab52e1f22468cbdc17d981187da287df04953a1
hanh@ubuntu:~$ sudo docker container exec -it apparmor-nginx bash
root@dafc48fded41:/# sh
bash: /bin/sh: Permission denied
```

Приложению не удалось выполнить команду sh

2. **Настроить selinux в режиме мандатного доступа (CentOS и др.) и продемонстрировать работу в двухуровневой модели.**

a. selinux

- Как SELinux работает?



- Режимы работы SELinux

  - Enforcing: Режим по-умолчанию. При выборе этого режима все действия, которые каким-то образом нарушают текущую политику безопасности, будут блокироваться, а попытка нарушения будет зафиксирована в журнале.

  - Permissive: В случае использования этого режима, информация о всех действиях, которые нарушают текущую политику безопасности, будут зафиксированы в журнале, но сами действия не будут заблокированы.

  - Disabled: Полное отключение системы принудительного контроля доступа.

- Мандатная модель управления доступом — способ разграничения доступа с фиксированным набором полномочий.

  В Security Enhanced Linux механизм мандатного управления доступом реализован в виде двух форм, если можно так выразиться:

  - MLS (Multi-Level Security, многоуровневая система безопасности) / MCS (Multi Categories Security, мультикатегорийная безопасность) Субъект может читать данные только на его уровне доступа и ниже, а записывать только в пределах своего уровня доступа.
  - TE (Type Enforcement, принудительная типизация доступа)

6. Ход работы

*** Принудительная типизация доступа**

- Использовать будем CentOS7
- Установим необходимые пакеты:
  yum install policycoreutils-python
- В выводе утилиты мы видим, что SELinux включен, текущий режим – enforcing.

```
[root@localhost hanhnguyen26]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
```

- Используемые файлы для запуски сервера

```
[root@localhost hanhnguyen26]# ls /home/hanhnguyen26/Downloads/lab8
index.html  pics  scr.js  style.css
[root@localhost hanhnguyen26]# ls -Z /home/hanhnguyen26/Downloads/lab8
-rw-r--r--. hanhnguyen26 hanhnguyen26 unconfined_u:object_r:user_home_t:s0 index.html
drwxr-xr-x. hanhnguyen26 hanhnguyen26 unconfined_u:object_r:user_home_t:s0 pics
-rw-r--r--. hanhnguyen26 hanhnguyen26 unconfined_u:object_r:user_home_t:s0 scr.js
-rw-r--r--. hanhnguyen26 hanhnguyen26 unconfined_u:object_r:user_home_t:s0 style.css
```

- Скопируем их в папку /var/www/html

```
[root@localhost hanhnguyen26]# cp -a /home/hanhnguyen26/Downloads/lab8/pics /var/www/html
[root@localhost hanhnguyen26]# cp -a /home/hanhnguyen26/Downloads/lab8/index.html /var/www/html
[root@localhost hanhnguyen26]# cp -a /home/hanhnguyen26/Downloads/lab8/style.css /var/www/html
```

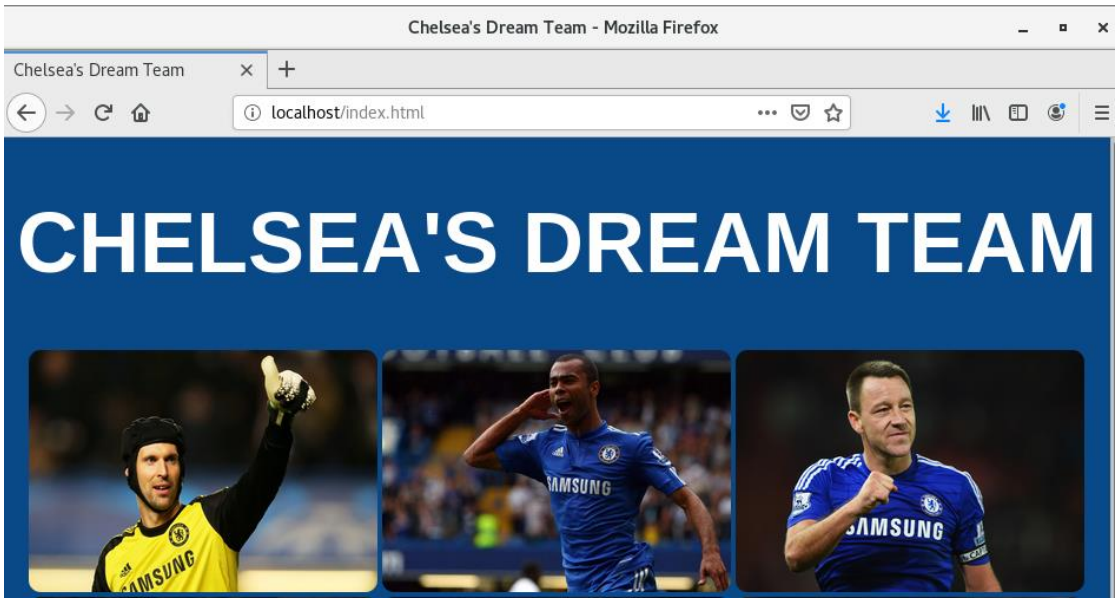
- Посмотрим на контекст безопасности файлов: тип – user_home_t

```
[root@localhost hanhnguyen26]# ls -Z /var/www/html
-rw-r--r--. hanhnguyen26 hanhnguyen26 unconfined_u:object_r:user_home_t:s0 index.html
drwxr-xr-x. hanhnguyen26 hanhnguyen26 unconfined_u:object_r:user_home_t:s0 pics
-rw-r--r--. hanhnguyen26 hanhnguyen26 unconfined_u:object_r:user_home_t:s0 scr.js
-rw-r--r--. hanhnguyen26 hanhnguyen26 unconfined_u:object_r:user_home_t:s0 style.css
```

- Так как правильный контекст безопасности для файлов, взаимодействующих с Apache, это httpd_sys_content_t, он не может получить доступ к файлу /var/www/html/index.html

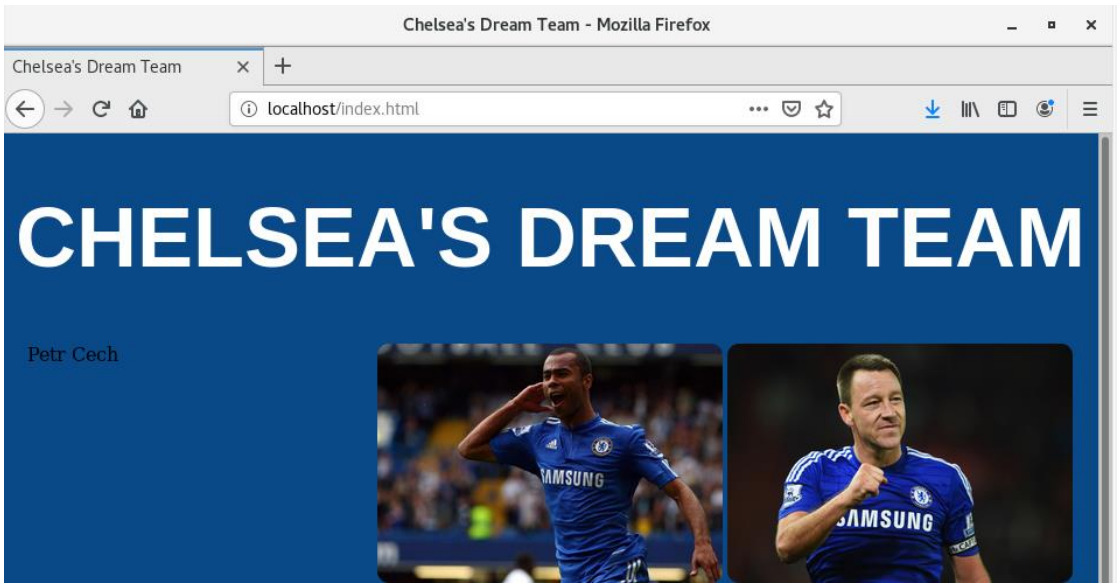
- Изменим контекст и проверим правильно ли все сделано

```
[root@localhost hanhnguyen26]# chcon -Rv --type=httpd_sys_content_t /var/www/html
changing security context of ‘/var/www/html/pics/Drogba.jpg’
changing security context of ‘/var/www/html/pics/Terry.jpg’
changing security context of ‘/var/www/html/pics/azpili.jpg’
changing security context of ‘/var/www/html/pics/cech.jpg’
changing security context of ‘/var/www/html/pics/cole.jpg’
changing security context of ‘/var/www/html/pics/fabregas.jpg’
changing security context of ‘/var/www/html/pics/hazard.jpg’
changing security context of ‘/var/www/html/pics/kante.jpg’
changing security context of ‘/var/www/html/pics/lampard.jpg’
changing security context of ‘/var/www/html/pics/luiz.jpg’
changing security context of ‘/var/www/html/pics/zola.jpg’
changing security context of ‘/var/www/html/pics’
changing security context of ‘/var/www/html/index.html’
changing security context of ‘/var/www/html/style.css’
changing security context of ‘/var/www/html/scr.js’
changing security context of ‘/var/www/html’
```



- Если изменим тип файл изображения на samba_share_t, он не может показан.

```
[root@localhost hanhnguyen26]# chcon --type=samba_share_t /var/www/html/pics/cech.jpg
```



*** Многоуровневая система безопасности**

Создадим двух пользователей и назначим им уровни доступа

```
[root@localhost home]# semanage login -l

Login Name           SELinux User         MLS/MCS Range        Service

__default__          user_u               s0-s0                *
hanh123              user_u               s1                   *
hanh789              user_u               s0-s2                *
root                 root                 s0-s15:c0.c1023      *
```

Также назначим уровни доступа домашним директориям созданных пользователей

```
[root@localhost hanh123]# chcon -R -l s1 /home/hanh123
```

```
[root@localhost hanh789]# chcon -R -l s2 /home/hanh789
```

```
[root@localhost home]# ls -lZ
total 4
drwx------. 15 hanh    hanh    unconfined_u:object_r:user_home_dir_t:s0 4096 May 26 14:22 hanh
drwx------.  3 hanh123 hanh123 user_u:object_r:user_home_dir_t:s1         94 May 26 16:05 hanh123
drwx------.  3 hanh789 hanh789 user_u:object_r:user_home_dir_t:s2         97 May 26 15:50 hanh789
```

```
[root@localhost hanh123]# ls -lZ
total 0
-rw-rw-r--. 1 hanh123 hanh123 root:object_r:user_home_t:s1 0 May 26 15:19 lab8.txt
```

```
[root@localhost hanh789]# ls -lZ
total 0
-rw-rw-r--. 1 hanh789 hanh789 root:object_r:unlabeled_t:s2 0 May 26 15:40 hanh789.txt
```

```
[hanh789@localhost ~]$ cat /home/hanh123/lab8.txt
cat: /home/hanh123/lab8.txt: Permission denied
[hanh789@localhost ~]$ echo itmo > /home/hanh123/lab8.txt
bash: /home/hanh123/lab8.txt: Permission denied
[hanh789@localhost ~]$
```

```
[hanh123@localhost hanh789]$ echo fbit > /home/hanh789/hanh789.txt
bash: /home/hanh789/hanh789.txt: Permission denied
```

3. Придумать и написать свой PAM-модуль (сложная авторизация действий)

- **Создадим PAM-модуль:** Система выдает пользователю два случайного числа. Пользователь должен выпонить сложение.

```
hanh@ubuntu:/etc/pam.d$ sudo touch pam_test.c
[sudo] password for hanh:
hanh@ubuntu:/etc/pam.d$ ls
chfn                        gdm-fingerprint         polkit-1
chpasswd                    gdm-launch-environment  ppp
chsh                        gdm-password            runuser
common-account              lab8.o                  runuser-l
common-auth                 login                   su
common-password             newusers                sudo
common-session              other                   su-l
common-session-noninteractive  pam_test.c           systemd-user
```

Файл pam_test.c

```
// Включаем необходимые заголовочные файлы.
#include <security/pam_modules.h>
#include <stdarg.h>
#include <time.h>

//Это определит тип нашего модуля
#define PAM_SM_AUTH
#define MAX_V 30
PAM_EXTERN int pam_sm_authenticate(pam_handle_t * pamh, int flags,int argc, const char
**argv)
{
        unsigned int ctrl;
        int retval;
        const char *name, *p;
        char *right;
        long x1,x2,y;

        //завели несколько случайных величин
        x1=random()%MAX_V;
        x2=random()%MAX_V;

       /* получим имя пользователя */
        retval = pam_get_user(pamh, &name, "login: ");

        /*получим пароль используя диалог*/
        {
            struct pam_conv *conv;
            struct pam_message *pmsg[3],msg[3];
```

```
            struct pam_response *response;


            retval = pam_get_item( pamh, PAM_CONV, (const void **) &conv ) ;

            pmsg[0] = &msg[0];
            msg[0].msg_style = PAM_PROMPT_ECHO_OFF;
            msg[0].msg=malloc(100);
            snprintf(msg[0].msg,60,"Second Password: %d + %d = ?",x1,x2);

            retval = conv->conv(1, ( const struct pam_message ** ) pmsg
                                , &response, conv->appdata_ptr);

            /*просчитаем правильный ответ*/
            y=x1+x2;
            right=malloc(100);
            snprintf(right,20,"%d",y);

            if (!(strcmp(right,response->resp))){
            return PAM_SUCCESS;
            }else{
            return PAM_AUTH_ERR;
            }
          }
            return PAM_SUCCESS;
}

// Инициализация идентификаторов групп
PAM_EXTERN int pam_sm_setcred(pam_handle_t * pamh, int flags
                              ,int argc, const char **argv)
{
        unsigned int ctrl;
        int retval;
        retval = PAM_SUCCESS;
        return retval;
}

// Это определение необходимо для статической линковки модулей PAM в приложениях.
#ifdef PAM_STATIC
struct pam_module _pam_unix_auth_modstruct = {
    "[pam_test",
    pam_sm_authenticate,
    pam_sm_setcred,
    NULL,
    NULL,
    NULL,
    NULL,
};
#endif
```

- **Скомпилируем**

```
hanh@ubuntu:/etc/pam.d$ sudo gcc -fPIC -fno-stack-protector -c pam_test.c
```

- **Положим в папку для модулей**

```
hanh@ubuntu:/etc/pam.d$ sudo ld -x --shared -o /lib/x86_64-linux-gnu/security/pam_test.so pam_te
st.o
hanh@ubuntu:/etc/pam.d$ sudo nano /etc/pam.d/login
```

- **Добовим наш модуль в login**

```
  GNU nano 4.8                                /etc/pam.d/login
# pam_selinux.so changes the SELinux context of the used TTY and configures
# SELinux in order to transition to the user context with the next execve()
# call.
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so open
# When the module is present, "required" would be sufficient (When SELinux
# is disabled, this returns success.)

# This module parses environment configuration file(s)
# and also allows you to use an extended config
# file /etc/security/pam_env.conf.
#
# parsing /etc/environment needs "readenv=1"
session       required   pam_env.so readenv=1
# locale variables are also kept into /etc/default/locale in etch
# reading this file *in addition to /etc/environment* does not hurt
session       required   pam_env.so readenv=1 envfile=/etc/default/locale

# Standard Un*x authentication.
@include common-auth

auth required pam_test.so
# This allows certain extra groups to be granted to a user
# based on things like time of day, tty, service, and user.
# Please edit /etc/security/group.conf to fit your needs
# (Replaces the `CONSOLE_GROUPS' option in login.defs)
auth       optional   pam_group.so
```

- **Результат**

```
hanh@ubuntu:/etc/pam.d$ sudo login
ubuntu login: hanh
Password:
Second Password: 13 + 16 = ?
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.13.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

246 updates can be applied immediately.
149 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun May 22 13:07:55 PDT 2022 on pts/0
```

Выводы: в результате работы мы настроили apparmor для мониторинга приложения, в результате чего часть функций приложения была недоступна. Также мы настроили selinux для работы в режиме мандатного доступа для двухуровневой системы. Результаты работы Selinux при наших настройках полностью соответствуют ожиданиям. Написали свой PAM-модуль.