ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

старший преподаватель		М. Д. Поляк
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О КУРСОВОЙ РАБОТЕ №1

ДОБАВЛЕНИЕ ЗАЩИТЫ ОТ НЕСАНКЦИОНИРОВАННОГО ЗАПУСКА ОПЕРАЦИОННОЙ СИСТЕМЫ.

по курсу: ОПЕРАЦИОННЫЕ СИСТЕМЫ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4236		А. К. Панин
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2025

Цель работы:

Знакомство с устройством ядра ОС Linux. Получение опыта разработки драйвера устройства.

Индивидуальное задание:

Необходимо внести изменения в процесс загрузки ядра Linux, добавив проверку наличия подключенного через интерфейс USB flash-накопителя с заданным серийным номером. Если в процессе загрузки операционной системы нужный flash-накопитель подключен к одному из портов USB, то операционная система успешно загружается в штатном режиме. Если flash-накопитель с нужным серийным номером отсутствует, отобразить на экране предупреждение и таймер с обратным отсчетом (30 секунд), загрузка операционной системы при этом приостанавливается. По истечении обратного отсчета таймера должно происходить автоматическое выключение компьютера. При подключении к любому из USB-портов нужного flash-накопителя во время обратного отсчета таймера, таймер должен останавливаться, после чего операционная система должна загружаться в штатном режиме.

Сравнение с аналогами:

В данном разделе рассматриваются существующие методы и подходы, аналогичные реализованному решению, а также их преимущества и недостатки. Проблема защиты операционной системы от несанкционированного запуска является актуальной, и несколько подходов предлагают решение этой задачи на разных уровнях системы. Рассмотрим несколько таких методов, включая проверки на уровне BIOS/UEFI, загрузчиков и системных скриптов.

Защита на уровне BIOS/UEFI

Одним из распространенных методов защиты от несанкционированного запуска является использование функций, предоставляемых BIOS или UEFI. Эти системы позволяют ограничить доступ к системе, проверяя, какие устройства подключены к компьютеру, и позволяют задавать требования для загрузки системы.

Преимущества:

- Защита осуществляется на уровне аппаратуры, до загрузки операционной системы, что делает её труднодоступной для взлома.
- Достаточно прост в использовании, не требует изменений в операционной системе.

Недостатки:

- Механизмы защиты на уровне BIOS/UEFI ограничены и не предлагают гибкости для пользовательских сценариев, таких как проверка серийных номеров устройств.
- Не всегда доступна возможность настроить BIOS/UEFI на уровне пользователя (например, в некоторых устройствах могут быть заблокированы настройки).
- Уязвимость к обходу (например, с помощью изменения настроек в BIOS/UEFI или через использование устройств, которые обходят эту защиту).

В отличие от защиты на уровне BIOS, разработанный метод представляет собой гибкое решение, интегрированное непосредственно в ядро Linux, позволяющее задавать конкретные проверки и действия на уровне операционной системы.

Использование загрузчиков GRUB и других альтернатив

Загрузчик GRUB (Grand Unified Bootloader) является важной частью процесса загрузки в операционных системах Linux. Он позволяет настраивать параметры загрузки и выполнять различные действия перед загрузкой операционной системы.

Варианты защиты с использованием GRUB включают:

- Проверка наличия определенных устройств при загрузке.
- Настройка времени задержки, вывода сообщений и выполнения определенных команд в случае, если нужные устройства не найдены.

Преимущества:

- Простота интеграции в существующую систему.
- Возможность динамической проверки устройств, без необходимости внесения изменений в ядро.
- Поддержка широкого спектра конфигураций загрузки.

Недостатки:

- Ограниченность в функционале: GRUB не имеет полноценных средств для сложных проверок, таких как работа с серийными номерами устройств.
- Невозможность выполнения более сложных операций, таких как остановка загрузки системы на определенное время или автоматическое выключение системы.

Метод, использующий только GRUB, ограничивает возможности контроля, в отличие от подхода, который предложен далее, где интеграция с ядром Linux позволяет более гибко контролировать процесс загрузки.

Использование скриптов в initramfs

Другим подходом является использование скриптов в initramfs. Это позволяет добавить дополнительные проверки и выполнить нужные действия до того, как будет загружена операционная система. Initramfs (initial RAM filesystem) — это временная файловая система, которая загружается до основного корня файловой системы и предоставляет возможность выполнить операции до полной загрузки системы.

Разработанный подход включает создание скрипта в initramfs, который проверяет наличие USB-флеш-накопителя и, в случае его отсутствия, приостанавливает процесс загрузки на 30 секунд с отображением сообщений на экране. В случае успешного подключения флеш-накопителя процесс загрузки продолжается.

Преимущества:

- Высокая гибкость в настройке процесса загрузки.
- Возможность выполнения сложных операций на уровне операционной системы, таких как проверка серийных номеров и взаимодействие с пользователем.
- Легкость интеграции с другими модулями и драйверами в Linux.

Недостатки:

- Необходимость работы с ядром Linux и его компонентами (например, initramfs), что требует определенных знаний и навыков.
- Может потребоваться модификация ядра для работы с нестандартными устройствами.

Этот метод предоставляет гораздо больше возможностей и контроля по сравнению с предыдущими подходами, так как позволяет выполнять сложные проверки и автоматические действия в зависимости от состояния системы.

Сравнение методов

Сравнив различные подходы к решению проблемы, можно выделить несколько ключевых аспектов:

- Гибкость: Использование скриптов в initramfs и драйвера ядра Linux предоставляет наибольшую гибкость и возможности для настройки, в то время как методы на уровне BIOS/UEFI и GRUB ограничены.
- Сложность внедрения: Модификация загрузчика GRUB или BIOS/UEFI требует меньших усилий по сравнению с модификацией ядра и initramfs, однако с этим подходом связаны ограничения в функционале.
- Уровень безопасности: Решение на уровне ядра Linux и initramfs обеспечивает лучший контроль над процессом загрузки и является более надежным, поскольку доступ к ядру труднее получить для злоумышленников.

Таким образом, использование скриптов в initramfs и изменение ядра представляется наиболее гибким и эффективным методом для защиты от несанкционированного запуска операционной системы.

Техническая документация

В этом разделе приведена пошаговая инструкция по установке и использованию разработанного скрипта для проверки USB-накопителя в процессе загрузки системы.

Предварительные требования

Перед началом установки убедитесь, что у вас есть следующие компоненты:

- Операционная система Linux (на базе Debian/Ubuntu).
- Установлены утилиты для работы с initramfs, такие как initramfs-tools.
- Доступ к правам суперпользователя (root) для модификации загрузчика и скриптов.
- Флеш-накопитель USB с уникальным серийным номером, который будет использоваться для проверки.

Шаг 1: Клонирование репозитория

1. Клонируйте репозиторий с исходным кодом скрипта для initramfs:

```
git clone https://github.com/Bolotnik-ss/os-option1.git
cd os-option1
```

Шаг 2: Получение серийного номера USB-накопителя

Для того чтобы настроить проверку нужного USB-накопителя, сначала необходимо получить его уникальный серийный номер. Для этого выполните следующие шаги:

1. Подключите флеш-накопитель к одному из USB-портов вашего компьютера.

2. Откройте терминал и выполните следующую команду:

```
lsusb
```

Эта команда покажет все подключенные USB-устройства. В выводе будет информация о подключенных устройствах, похожая на следующую:

```
Bus 002 Device 003: ID 1234:abcd Example USB Flash Drive
```

3. Чтобы узнать серийный номер устройства, используйте следующую команду:

```
sudo lsusb -v -d 1234:abcd | grep iSerial
```

Замените 1234:abcd на идентификатор вашего устройства из вывода команды lsusb. Вы получите строку, похожую на:

```
iSerial                 3 1234567890ABCDEF
```

В данном примере серийный номер устройства — 1234567890ABCDEF.

4. Скопируйте серийный номер устройства, так как он потребуется для настройки скрипта.

Шаг 3: Редактирование скрипта

1. Откройте файл скрипта option1.sh для редактирования:

```
nano option1.sh
```

2. Найдите строку, которая отвечает за проверку серийного номера:

```
REQUIRED_SERIAL="5FD325C5E8AF4A8D"
```

3. Замените значение 5FD325C5E8AF4A8D на серийный номер вашего флеш-накопителя, который вы получили на предыдущем шаге.

4. Сохраните изменения и выйдите из редактора (Ctrl + O, затем Enter для сохранения и Ctrl + X для выхода).

Шаг 4: Копирование скрипта в нужную директорию

1. Скопируйте скрипт из репозитория в каталог /etc/initramfs-tools/scripts/init-bottom:

```
sudo cp option1.sh /etc/initramfs-tools/scripts/init-bottom/
```

2. Убедитесь, что скрипт имеет права на выполнение:

```
sudo chmod +x /etc/initramfs-tools/scripts/init-bottom/option1.sh
```

Шаг 5: Пересборка initramfs

1. После того как скрипт будет скопирован в нужную директорию, необходимо пересобрать initramfs, чтобы изменения вступили в силу:

```
sudo update-initramfs -u
```

Шаг 6: (При необходимости) Обновление конфигурации GRUB

1. Если в процессе загрузки вы не видите текстовые сообщения, а видите лишь графический интейрфейс загрузчика операционной системы, убедитесь, что в конфигурации GRUB отсутствуют параметры quiet splash. Откройте файл конфигурации GRUB для редактирования:

```
sudo nano /etc/default/grub
```

2. В строке GRUB_CMDLINE_LINUX_DEFAULT уберите параметры quiet splash:

```
GRUB_CMDLINE_LINUX_DEFAULT=""
```

3. Сохраните изменения и обновите конфигурацию GRUB:

```
sudo update-grub
```

Использование разработанного скрипта

1. При следующей загрузке системы, если нужный USB-накопитель не будет подключен, загрузка будет приостановлена на 30 секунд, и на экране отобразится сообщение с обратным отсчетом. По истечении времени система автоматически выключится.

2. Если флеш-накопитель будет подключен до завершения таймера, таймер остановится, и система продолжит загрузку в штатном режиме.

Удаление скрипта и восстановление состояния

Если вам нужно удалить скрипт и вернуть систему в прежнее состояние:

1. Удалите скрипт из каталога initramfs:

```
sudo rm /etc/initramfs-tools/scripts/init-bottom/option1.sh
```

2. Пересоберите initramfs:

```
sudo update-initramfs -u
```

3. (При необходимости) Восстановите параметры quiet splash в конфигурации GRUB и пересоберите его:

```
sudo nano /etc/default/grub
```

Вставьте обратно:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
```

Затем обновите конфигурацию:

```
sudo update-grub
```

Решение проблем

- Не отображается сообщение при отсутствии флеш-накопителя: Убедитесь, что вы удалили параметры quiet splash в конфигурации GRUB и обновили конфигурацию с помощью update-grub.
- USB-накопитель не распознается: Проверьте настройки скрипта и убедитесь, что серийный номер устройства совпадает с тем, который указан в конфигурации скрипта.

Выводы:

В ходе выполнения курсового проекта была разработана система защиты от несанкционированного запуска операционной системы Linux, которая проверяет наличие флеш-накопителя с заданным серийным номером в процессе загрузки.

Основными результатами работы являются:

- Реализован скрипт, который выполняет проверку подключения флеш-накопителя и обеспечивает задержку загрузки системы с таймером в случае его отсутствия.
- Скрирпт успешно интегрирован в процесс формирования initramfs, что позволяет его выполнение уже на этапе начальной загрузки.
- Предусмотрено автоматическое выключение компьютера, если флеш-накопитель не был подключен в течение 30 секунд, а также предусмотрена остановка таймера при подключении нужного устройства.
- Реализована простая настройка скрипта с помощью редактирования переменной, что делает систему гибкой и удобной для использования.
- Обеспечена возможность восстановления исходного состояния системы в случае необходимости.

В ходе реализации проекта были получены навыки работы с процессом загрузки Linux, а также с инструментами для создания и модификации initramfs и GRUB. Работа с ядром и конфигурациями загрузчика расширила понимание внутреннего устройства операционных систем.

В результате выполнения проекта была создана функциональная система защиты от несанкционированного запуска, которая может быть применена для повышения уровня

безопасности систем, требующих дополнительной защиты от неавторизованного доступа. Это решение может быть полезным для использования в рабочих и учебных средах, где необходимо ограничить возможность загрузки системы без определенных устройств.

Таким образом, проект успешно решает поставленную задачу и предоставляет эффективное решение для защиты операционной системы от несанкционированного запуска.

Приложение. Листинг скрипта проверки:

option1.sh:

```
#!/bin/sh

SERIAL_NUMBER="5FD325C5E8AF4A8D"

check_usb_device() {
    for dev in /sys/bus/usb/devices/*; do
        if [ -e "$dev/serial" ]; then
            SERIAL=$(cat "$dev/serial")
            if [ "$SERIAL" = "$SERIAL_NUMBER" ]; then
                return 0
            fi
        fi
    done
    return 1
}

main() {
  echo "Проверка наличия USB-устройства с серийным номером $SERIAL_NUMBER..." > /dev/
    if check_usb_device; then
        echo "Устройство найдено. Продолжаем загрузку." > /dev/console
        return 0
    fi

    echo "Устройство не найдено. Ожидание подключения..." > /dev/console
    for i in $(seq 30 -1 1); do
        echo "Осталось $i секунд..." > /dev/console
        sleep 1
        if check_usb_device; then
            echo "Устройство подключено. Продолжаем загрузку." > /dev/console
            return 0
        fi
    done

    echo "Таймер истек. Выключение системы." > /dev/console
    poweroff
}
main
```