Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

«Модуль ядра для подмены пользовательских данных, передаваемых на съёмные USB-носители»

Студент ИУ7-75Б (Группа) _____ (Подпись, дата) Смирнов П. И. (И. О. Фамилия)

Руководитель курсовой работы _____ (Подпись, дата) Рязанова Н. Ю. (И. О. Фамилия)

2026 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой ИУ7
_____ И. В. Рудаков
«__» сентября 2025 г.

З А Д А Н И Е
на выполнение курсовой работы

по теме

Модуль ядра для подмены пользовательских данных, передаваемых на съёмные USB-носители

Студент группы **ИУ7-75Б**

Смирнов Пётр Ильич

Направленность КуР (учебная, исследовательская, практическая, производственная, др.): учебная.

Источник тематики (кафедра, предприятие, НИР): кафедра.

График выполнения КуР: 25% к 5 нед., 50% к 8 нед., 75% к 11 нед., 100% к 15 нед.

Техническое задание
Разработать модуль ядра для подмены пользовательских данных, передаваемых на USB-накопители с файловой системой FAT32. Использовать механизм kprobes.

Оформление курсовой работы:
Расчетно-пояснительная записка на 40-50 листах формата А4.

Дата выдачи задания «__» сентября 2025 г.

Руководитель НИР _____ **Н. Ю. Рязанова**

Студент _____ **П. И. Смирнов**

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Предотвращение утечек данных через USB-устройства является одной из важнейших задач Data Leak Prevention программно-аппаратных комплексов [1]. Основными механизмами для решения этой задачи в ядре Linux являются драйверы устройств и перехват функций ядра. Данная работа посвящена созданию и исследованию загружаемого модуля ядра для подмены данных, передаваемых на USB-накопители, с использованием перехвата управления у функций.

1 Аналитическая часть

1.1 Постановка задачи

В соответствии с заданием на курсовую работу необходимо разработать модуль ядра для подмены пользовательских данных, передаваемых на USB-накопители. Подмена данных осуществляется изменением значения каждого байта данных, с использованием XOR операции. Для достижения поставленной цели необходимо решить следующие задачи:

1. провести обзор функций и структур ядра, представляющих возможность реализации разрабатываемого модуля;
2. провести анализ методов перехвата управления у функций ядра;
3. провести анализ файловых систем внешних накопителей;
4. реализовать и протестировать модуль.

1.2 Архитектура подсистемы USB в Linux

Основные компоненты, реализующие работу с USB-устройствами [2]:

1. драйвер устройства;
2. ядро USB (USB Core) – общий код, управляющий всей подсистемой;
3. драйвер хост-контроллера (HCD) – реализует программно-аппаратное взаимодействие.

Схема взаимодействия драйверов USB:

1. драйвер устройства создает или получает запрос на передачу информации;
2. драйвер инициализирует запрос всей необходимой информацией:
   - тип запроса (чтение, запись, управление);
   - адрес устройства и номер конечной точки (endpoint);
   - указатель на буфер данных;

— размер буфера данных;

— функция обратного вызова (callback) – которая будет вызвана, когда операция завершится.

3. драйвер отправляет запрос в ядро USB;

4. ядро и HCD выполняют необходимую низкоуровневую работу, чтобы выполнить этот запрос на физической шине USB;

5. когда операция завершена (данные получены, отправлены или произошла ошибка), HCD вызывает функцию обратного вызова для этого запроса, чтобы уведомить драйвер устройства о результате.

1.3 Анализ основных структур ядра для работы с USB

Для управления внешним накопителем, таким как USB-устройство в операционной системе представлено несколько структур.

1.3.1 struct User Request Block (URB)

URB — структура, инкапсулирующая запрос на передачу данных, передаваемый драйвером USB-устройства низкоуровневому драйверу хост-контроллера. Каждая операция ввода-вывода через USB шину оформляется в виде отдельного URB, что обеспечивает единый интерфейс для работы с различными типами USB-передач.

Типы передач, поддерживаемые URB

URB поддерживает все четыре типа передач, определенные в спецификации USB [3]:

1. управляющие посылки (Control Transfers), используемые для конфигурирования во время подключения и в процессе работы для управления устройствами.

2. сплошные передачи (Bulk Data Transfers) сравнительно больших пакетов без жестких требований ко времени доставки. Пакеты имеют поле данных

размером 8, 16, 32 или 64 байт. Приоритет этих передач самый низкий, они могут приостанавливаться при большой загрузке шины.

3. прерывания (Interrupt) - короткие передачи, имеют спонтанный характер и должны обслуживаться не медленнее, чем того требует устройство.

4. изохронные передачи (Isochronous Transfers) - непрерывные передачи в реальном времени, занимающие предварительно согласованную часть пропускной способности шины и имеющие заданную задержку доставки.

Ключевые поля структуры URB

— struct usb_device *dev – представление устройства USB;

— int status – статус выполнения;

— void *transfer_buffer – указатель на область данных для передачи;

— u32 transfer_buffer_length – размер передаваемого буфера;

— struct scatterlist *sg – указатель на начало списка разрозненных буферов для передачи;

— int num_sgs – длина списка разрозненных буферов;

— u32 actual_length – фактически переданное количество байт;

— usb_complete_t complete – функция обратного вызова.

1.3.2 struct usb_device

2 Конструкторская часть

3 Технологическая часть

4 Исследовательская часть

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. DLP: предотвращаем утечки. [Электронный ресурс]. — Режим доступа: https://habr.com/ru/companies/otus/articles/798787/ (дата обращения: 03.12.2025).
2. Linux-USB Host Side API. [Электронный ресурс]. — Режим доступа: https://microsin.net/programming/arm-working-with-usb/linux-usb-host-side-api.html (дата обращения: 03.12.2025).
3. Шина USB и FireWire. [Электронный ресурс]. — Режим доступа: https://cxem.net/comp/comp56.php (дата обращения: 03.12.2025).

ПРИЛОЖЕНИЕ А