МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»
Кафедра «Системы информационной безопасности»

«ДОПУЩЕН К ЗАЩИТЕ»
Зав. кафедрой к.п.н., доцент Р. Ш. Бердибаев
_____ «_____» _____2019 г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: «Защита корпоративного сервера на базе операционной системы Linux»
Специальность: 5В100200 – «Системы информационной безопасности»
Выполнил: Ибрагимұлы Ерман Группа СИБ-15-2
Научный руководитель: ст. преп. Зуева Екатерина Александровна

Консультант:
по экономической части:
к. э. н., профессор Аренбаева Н. Г.
(ученая степень, звание, Ф.И.О)
_____ «27» мая 2019 г.
(подпись)

по безопасности жизнедеятельности:
д. т. н. ст. преп Бейсагаев Ш. Ш.
(ученая степень, звание, Ф.И.О)
_____ «14» мая 2019 г.
(подпись)

по применению вычислительной техники:
ст. преп. Зуева Е. А.
(ученая степень, звание, Ф.И.О)
_____ «25» мая 2019 г.
(подпись)

нормоконтролер:
Аскарова Н. Б., ст. преподаватель
(ученая степень, звание, Ф.И.О)
_____ «31» 05 2019 г.
(подпись)

рецензент:
PhD, ассистент-профессор Алиакпарова С. Т.
(ученая степень, звание, Ф.И.О)
_____ «28» мая 2019 г.
(подпись)

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт Систем управления и информационных технологий

Кафедра систем информационной безопасности

Специальность 5В100200 – «Системы информационной безопасности»

ЗАДАНИЕ
на выполнение дипломного проекта

Студенту: Ибрагимұлы Ерману

Тема проекта: «Защита корпоративного сервера на базе операционной системы Linux»

Утверждена приказом по университету №124 от «26» окт 2018 г.

Срок сдачи законченного проекта «25» мая 2019 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): рассмотреть существующие системы развертывания и обеспечения защиты web-серверов и FTP-серверов, проанализировать, выбрать инструментарий разработки ПО, грамотно спроектировать программный продукт с Nginx в качестве обратного прокси к Apache и обеспечив защиту от перебора паролей, fail2ban, протестировать, сделать аналитику и внести коррективы, рассчитать экономическую эффективность и БЖД.

Перечень вопросов, подлежащих разработке в дипломном проекте или краткое содержание дипломного проекта: представить общую информацию по теме организации и функционированию корпоративных серверов (установка операционной системы, поднятие сервера, установка защиты, защита от LiveCD-монтирования); демонстрация организации беззащитности сервера (проведение атак, анализ уязвимостей); описать процесс разработки, подкрепленный скриншотами, рассчитать экономическую составляющую; сделать технико-экономическое обоснование и необходимые расчеты.

Перечень графического материала (с точным указанием обязательных чертежей):

1 создание сервера, его настройка;
2 принципы работы необходимых составляющих защиты сервера;
3 скриншоты конфигурационных файлов сервера Ubuntu;
4 скриншоты результатов атаки на сервер;

5 скриншоты интерфейса создаваемого средства защиты программного обеспечения;

6 скриншоты результатов тестирования и аналитики безопасности системы.

Основная рекомендуемая литература:

1 Алексеенко К. Web-сервер глазами хакера. - СПб.: БХВ-Петербург, 2019.

2 Романов П.Ю., Лисьев Г.А. Программное обеспечение компьютерных сетей и web-серверов. – М.: Инфра-М, 2015.

3 Жуков Ю. Основы WEB-хакинга. Нападение и защита. – Воронеж: 2011.

4 Полежаев П.Н., Малахов А.К., Сагитов А.М. «Ахиллесова пята» USB-устройств: атака и защита // Философские проблемы информационных технологий и киберпространства. 2015. № 1(9). С. 106–117.

5 Хакимжанов Т.Е. Расчет аспирационных систем. Дипломное проектирование. Для студентов всех форм обучения всех специальностей. – Алматы: АУЭС, 2014.

6 Бекишева А.И. Методические указания к выполнению экономической части дипломной работы для бакалавров специальности 5В0703 – «Информационные системы». – Алматы: АУЭС, 2013.

Конструкции по проекту с указанием относящихся к нему разделов:

Раздел	Консультант	Сроки	Подпись
Глава 1	Зуева Е.А.	14.01.19 – 26.01.19	
Глава 2	Зуева Е.А.	27.01-19 – 4.2.19	
Глава 3	Зуева Е.А.	5.02.19 – 25.5.19	
Глава 4	Аренбаева Тс.Т.	04.03 – 27.05.19	
Глава 5	Бекбасарова Ш.Ш.	14.03.19 – 23.04.19	

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Теоретические аспекты функционирования серверов и организации защиты информации	28.01.19	нет
Создание корпоративного сервера	4.02.19	нет
Демонстрация атак	4.02.19	нет
Анализ уязвимостей		нет
Выбор средств и разработка мер обеспечения информационной безопасности сервера	11.02.19	нет
Реализация мер обеспечения информационной безопасности сервера	11.02.19	нет
Тестирование созданного ПО	18.02.19	нет
Внесение коррективов в работу существующего ПО	18.02.19	нет
Подготовка и проведение необходимых расчетов	25.02.19	нет
Аналитика	4.03.19	нет
Создание выводов и заключения	22.04.19	нет
Оформление материала	29.04.19	нет

Дата выдачи задания «14» января 2019 г.

Заведующий кафедрой _____ (_____)
(Подпись) (Ф.И.О)

Научный руководитель проекта _____ (Зуева Е.А.)
(Подпись) (Ф.И.О)

Задание принял к исполнению студент _____ (Ибрагимов Е.)
(Подпись) (Ф.И.О)

ОТЗЫВ

на дипломный проект студента группы СИБ-15
Алматинского университета энергетики и связи
Ибрагимулы Ермана
на тему:
"Защита корпоративного сервера на базе Linux–сервера"

Дипломный проект связан с вопросами разработки и внедрения мер защиты на корпоративном сервере под управлением операционной системы Linux.

В проекте был детально рассмотрен процесс создания корпоративного сервера, проработка мер защиты, теория и проведение экспериментов по созданию эффективного эшелона линии защиты.

Был проведен анализ существующих защитных мер корпоративных серверов на операционной системе семейства Ubuntu.

Исходя из анализа и опыта разработок подобных систем, дипломник реализовал систему с учетом всех достоинств аналогов, комплексно проработал вопросы безопасности, создал собственную систему «kinza», проанализировал функционал созданной системы, доработал и полностью устранил возможные слабые места в процессе отладки.

Объем в проекте большой и проект объединяет работу по нескольким направлениям:

1) Включение / выключение USB-портов;
2) включение режима «только для чтения»;
3) поиск подключенного USB-устройства;
4) очистка терминала;
5) вывод всех блочных устройств, подключенных к системе;
6) вывод всего подключенного оборудования;
7) список подключенного USB-флеш;
8) отмонтирование USB-флеш;
9) форматирование USB-флеш.

Актуальность и достоинство проекта заключается в том, что разработанная система сочетает в себе компоненты, позволяющие улучшить безопасность любого корпоративного сервера программным методом; в множестве систем эти компоненты реализованы по отдельности, но нет обобщенной системы, которая бы сочетала в себе все эти компоненты и не приводила к конфликту модулей между собой, а в системе дипломника эти ньюансы были учтены.

В связи с вышеизложенным считаю, что работа выполнена на отличном уровне и полностью соответствует требованиям, предъявляемым к дипломным проектам.

Считаю, что дипломная работа заслуживает оценки «отлично» с оценкой 100%, а дипломник присвоения квалификации бакалавра по специальности 5В100200 – "Системы информационной безопасности".

Руководитель, ст. преп. каф. СИБ АУЭС  *Зуева Е. А.*
25.05.2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

РЕЦЕНЗИЯ

на дипломный проект студента
Алматинского университета энергетики и связи
Ибрагимұлы Ермана

Специальность: 5В100200 – "Системы информационной безопасности"
Дипломный проект на тему: «Создание программного обеспечения для демонстрации работы методов стеганографии» выполнена в составе:

а) презентация в электронном виде (22);
б) пояснительная записка на 81 страницах.

Дипломный проект связан с вопросами разработки и внедрения мер защиты на корпоративном сервере под управлением операционной системы Linux.

Пояснительная записка состоит из введения, пяти глав, заключения, списка использованной литературы, приложения.

В проекте был детально рассмотрен процесс создания корпоративного сервера, проработка мер защиты, теория и проведение экспериментов по созданию эффективного эшелона линии защиты.

Был проведен анализ существующих защитных мер корпоративных серверов на операционной системе семейства Ubuntu.

Исходя из анализа и опыта разработок подобных систем, дипломник реализовал систему с учетом всех достоинств аналогов, комплексно проработал вопросы безопасности, создал собственную систему «kinza», проанализировал функционал созданной системы, доработал и полностью устранил возможные слабые места в процессе отладки.

Объем в проекте большой и проект объединяет работу по нескольким направлениям:

1) включение USB-портов / выключение USB-портов;
2) включение режима «только для чтения»;
3) поиск подключенного USB-устройства;
4) очистка терминала;
5) вывод всех блочных устройств, подключенных к системе;
6) вывод всего подключенного оборудования;
7) список подключенного USB-флеш;
8) отмонтирования USB-флеш;
9) форматирование USB-флеш.

Достоинство дипломного проекта: выполнен хороший теоретический анализ проблемы по теме, поставлены и выполнены основные задачи проекта:

- проведение анализа современного состояния корпоративных серверов;
- разработка собственной системы для блокировки USB-портов и реализация соответствующего программного обеспечения;
- исправление ошибок в созданном ПО, внесение корректив;
- разбор экономической части и БЖД.

Замечание: единственным замечанием является чрезмерное количество иллюстрации, описывающих процесс в мельчайших подробностях, которые можно было опустить (не критично).

Оценка работы.

В целом проект выполнен на достойном уровне, соответствующем требованиям, предъявляемым к данным видам работ.

Считаю, что дипломный проект заслуживает оценки «отлично», а Ибрагиұлы Е. присвоения квалификации бакалавра по специальности 5В100200 - "Системы информационной безопасности".

28.05.2019

Рецензент: *Аманжолова С.Т.*

к.т.н., доцент асс. профессор кафедры
«Компьютерная инженерия и информационная безопасность»
Международного университета информационных технологий

Подпись указанного лица удостоверяю
Аманжоловой С.Т.
Инспектор по кадрам

АННОТАЦИЯ

Данный дипломный проект посвящен созданию эффективной системы защиты корпоративного сервера, её тестированию, анализу и устранению уязвимостей корпоративных серверов.

В экономической части произведен анализ возможностей коммерциализации программных продуктов, которые использовались создания программного обеспечения в дипломном проекте, рассчитаны затраты на разработку дипломного проекта и произведен расчет экономической эффективности при построении модели.

В разделе безопасности жизнедеятельности определены оптимальные условии труда, произведен расчет освещенности в помещении, где разрабатывался проект.

АҢДАТПА

Бұл дипломдық жоба корпоративтік серверді жүйелі әрі тиімді қорғау және сараптау, сынақтау. Сонымен қатар әлсіз жүйелерін анықтап жою.

Экономикалық бөлімінде дипломдық жобаны құрау үшін қолданылған бағдарламалық өнімдер экономикалық тұрғыда сарапталынған. Экономикалық жағынан әсіресе модель құралған.

Қауіпсіздік бөлімінде өмір ағымдарында тиімді еңбек жағдайлары анықталған. Жобаның өңделген орнын жарықтандыруы есептелінген.

ANNOTATION

The present diploma project is devoted to the effective corporation server defense, its testing, analysis and elimination of its vulnerabilities.

Economical part includes an analysis of opportunities of program products commercializing, which were used for a creation of a software in diploma project, expenses for development of the diploma project and accounts of economical effectiveness for construction of the model were provided.

In the section of life safety optimal labor conditions are defined, an account of inside lightning of a place where the project has been developed was provided.

Содержание

Введение

Сетевые технологии развиваются с огромной скоростью. Растут вычислительные мощности, пропускная способность, расширяется спектр услуг, предлагаемых ISP (Интернет-провайдер; англ. Internet Service Provider), изобретаются новые механизмы сетевого взаимодействия. Это нацелено на объединения ресурсов и совместную работу тысяч, миллионов пользователей. Все острее стоит вопрос защиты ресурсов и разграничения доступа к ним. К сожалению, третьи лица пытаются получить доступ к конфиденциальной информации, являющейся интеллектуальной собственностью компаний, к сетевым услугам, или же направляют свои усилия на разрушение работоспособности отдельных хостов или всей сети. Чем больше ресурсов компания объединяет в своей корпоративной сети, тем больше создается угроз для них, тем труднее обеспечить сетевую безопасность. Для надежной защиты ресурсов необходимо реализовывать комплексный подход в обеспечение сетевой безопасности корпоративных мультисервисных сетей. Предлагаемые решения перед внедрением должны быть всесторонне протестированы. Это касается не только проверки оборудования и ПО, но и подготовки квалифицированного персонала, способного правильно с ним работать.

Актуальность моей работы состоит в организации эффективных способов защиты от DDOS атак и демонстрации защит аппаратных устройств от взлома, блокировки USB-портов, чтобы защитить сервер от несанкционированного доступа.

Цель дипломного проекта заключается в защите корпоративного сервера на базе операционной системы Linux, через создание программного обеспечения, которое контролирует аппаратно-программный доступ к «железу» сервера.

Задачи, которые я поставил для достижения цели:

- изучить теорию по организации корпоративных серверов;
- поднять сервер;
- протестировать работу сервера (продемонстрировать атаки);
- проанализировать методы взлома на поднятом сервере;
- разработать приложение для блокировки USB-портов на языке Python;
- установить защиту на поднятый сервер;
- протестировать;
- сделать анализ и соответствующие выводы;
- рассмотреть экономическую часть и БЖД.

1 Создание корпоративного сервера

1.1 Корпоративные сервера

Сервер – ключевой элемент корпоративной инфраструктуры.

С точки зрения аппаратных средств сервер – это компьютер, который способен оказывать некоторые услуги другим, подсоединенным к нему компьютерам. Подразумевается, что компьютеры каким-то образом связаны с сервером и друг с другом [1].

Правильно подобрать сервер для организации – это нелегкая задача. Широкий выбор серверных систем требует от руководителей ИТ-служб реалистично оценивать требования к их вычислительной мощности, масштабируемости, надежности и степени готовности. Они должны четко сформулировать требования к серверам, изучить возможности поддержки, а также определить будущие затраты на модернизацию.

Серверы можно классифицировать, например, по классу решаемых задач, а также по количеству обслуживаемых клиентов. Согласно второму подходу, серверы рабочих групп различаются; отдел (отдел); средние организации (среднего класса); предприятие (предприятие).

Надо сказать, что, поскольку в пределах каждого типа сервера конфигурация значительно варьируется, установить четкие границы между ними невозможно. Мощные низкоуровневые компьютеры могут служить серверами начального уровня в верхнем соседнем классе и наоборот.

Отмечу, что классификаций серверов существует довольно много, причем все они в той или иной степени перекрываются. Так, фирмы-производители часто подразделяют выпускаемые серверы по типу исполнения: сверхтонкие (blade), классические напольные (tower), предназначенные для установки в стойки (rack) и с высокой степенью масштабируемости (super scalable). Ультратонкие компьютеры могут не только сэкономить место на каждом сервере, но и снизить энергопотребление. Напольные серверы обеспечивают высокую гибкость при размещении компонентов в корпусе и легко масштабируются. Серверы для установки в стойки, предназначены для консолидации серверных систем в центрах обработки данных и использования с подсистемами внешней памяти. Они могут эффективно использоваться для кластерных решений, когда сами серверы, внешняя память и дополнительные устройства размещаются в одинаковых стойках. Серверы с высокой степенью масштабируемости обычно предназначены для крупных предприятий и способны предоставлять решения практически для любых корпоративных задач.

Ниже описываются некоторые распространенные типы серверов, классифицируемых по классу решаемых задач:

1) Web-серверы.

Web сервер – сервер, принимающий запросы (в основном, от браузера) и возвращающий ответ, как правило, в виде HTML разметки. Проще говоря, он нужен для того, чтобы браузер пользователя мог открыть сайт. Примитивная схема работы: клиент отправляет http-запрос - сервер его обрабатывает и выдает ответ [2].

Большинство веб-серверов реализовано на базе программного обеспечения (ПО) Apache, NGINX, Lighttpd или их комбинации. Это бесплатные программы и их можно скачать с официальных сайтов. Основную массу составляют сервера на базе Linux и FreeBSD. Основная причина – бесплатность и сверхнадежность UNIX-систем. Но есть немного веб-серверов на Windows, также с установленным перечисленным ПО или встроенным IIS.

Apache HTTP Server, условно называемый Apache, представляет собой бесплатное кроссплатформенное веб-серверное программное обеспечение с открытым исходным кодом, выпущенное на условиях Apache License 2.0. Apache разрабатывается и поддерживается открытым сообществом разработчиков под эгидой Apache Software Foundation.

Подавляющее большинство экземпляров Apache HTTP Server работает на дистрибутиве Linux, но текущие версии также работают в Windows и на множестве Unix-подобных систем.

Nginx (стилизованный под NGINX или nginx) - это веб-сервер, который также можно использовать в качестве обратного прокси-сервера, балансировщика нагрузки, почтового прокси-сервера и HTTP-кэша. Программное обеспечение было создано Игорем Сысоевым и впервые опубликовано в 2004 году. Одноименная компания была основана в 2011 году для предоставления поддержки и платного программного обеспечения Nginx плюс.

Nginx - это бесплатное программное обеспечение с открытым исходным кодом, выпущенное на условиях BSD-подобной лицензии. Большая часть веб-серверов используют NGINX, часто в качестве балансировщика нагрузки.

2) Серверы баз данных.

Серверы баз данных используются для обработки бизнес-транзакций и пользовательских запросов. По мере расширения электронного бизнеса прикладные базы данных становятся все более сложными и большими по размеру. Главной особенностью сервера базы данных является его способность быстро извлекать и форматировать данные. Вычислительная мощность и масштабируемость системы играют в этом решающую роль.

3) Файл-серверы.

Файл-сервер – обеспечивает взаимодействие между сетевыми станциями и дает пользователям доступ к файлам, которые необходимы им для работы. Файл-сервер ограничивает несанкционированный доступ к данным.

В вычислительной технике файловый сервер (или файловый сервер) - это компьютер, подключенный к сети, который обеспечивает место для общего доступа к диску, то есть общее хранилище компьютерных файлов (таких как текст, изображение, звук, видео), к которым могут обращаться рабочие станции. которые могут получить доступ к компьютеру, который разделяет доступ через компьютерную сеть. Термин сервер обозначает роль компьютера в схеме клиент-сервер, где клиенты являются рабочими станциями, использующими хранилище. Обычно файловый сервер не выполняет вычислительные задачи и не запускает программы от имени своих клиентов. Он предназначен главным образом для обеспечения возможности хранения и извлечения данных, пока вычисления выполняются рабочими станциями.

Файловые серверы обычно находятся в школах и офисах, где пользователи используют локальную сеть для подключения своих клиентских компьютеров.

4) Брандмауэры.

Брандмауэр, как и имплицирует его «боевое» имя, дает собой средство обеспечения защищенности, задачи которого во многом идентичны с работой пограничников: осматривать любой кусок данных, который пробует пересечь границу сети.

5) Почтовые серверы.

Почтовый сервер (иногда называемый сервером сообщений) занимается как входящими, так и исходящими запросами. Одна из задач почтового сервера - это чтение адресов входящих сообщений и доставка корреспонденции в соответствующие почтовые ящики в пределах интернета. В зависимости от развитости почтового сервера он может предоставлять администратору большую или меньшую степень контроля над локальными почтовыми ящиками, типами и размерами сообщений, которые они в состоянии получать, автоматическими ответами, которые можно составлять.

7) Серверы FTP.

Подобные серверы, работающие на основе протокола File Transfer Protocol, уже много десятилетий назад стали стандартом действительности при перемещении файлов в Интернете. FTP-серверы поддерживают работу обычных файловых менеджеров – пользователей. Сложные FTP-серверы обеспечивают администратору большие возможности управления в том, что касается прав на подключение и совместного использования файлов, типов делимых файлов и их размещения. Конфигурируемые ресурсы, выделяемые ряду соединений с сервером, ограничения на количество передаваемых данных и минимальную скорость передачи и т.п., становятся все более популярными средствами, помогающими повысить безопасность FTP-серверов.

8) Принт-серверы.

Сервер печати - это сервер, который позволяет всем компьютерам, подключенным к сети, печатать документы на одном или нескольких общих

принтерах. Нет необходимости оснащать каждый компьютер собственным печатающим устройством. Кроме того, принимая во внимание все заботы о печати документов, сервер печати освобождает компьютеры для другой работы. Например, сервер печати хранит документы, отправленные для печати, на своем жестком диске, выстраивает их в очередь и печатает в порядке приоритета.

9) Серверы удаленного доступа.

Серверы удаленного доступа – это системы позволяют связываться с офисной сетью по телефонным линиям. Находясь с ноутбуком где-нибудь вдали от офиса, всегда можно скачать нужный файл, получить любую необходимую информацию. При наличии хороших каналов связи разница между работой в офисе и вне его в этом случае практически незаметна.

1.2 Установка ОС

Для работы я выбрал дистрибутив из семейства Linux – Ubuntu и виртуальную машину VmWare.

Ubuntu – это современная полнофункциональная операционная система, основанная на ядре Linux.

Ubuntu - это международное добровольное сообщество людей, объединенных в проект по созданию и разработке программного обеспечения. Целью такого сообщества является стремление донести до окружающих лучшее из собственного опыта.

Ubuntu распространяется совершенно бесплатно. Установив Ubuntu на свой компьютер, можно получить полный набор всех необходимых для работы приложений, а всё, что по каким-то причинам не вошло в стандартную поставку, можно легко установить из Интернета. Использовать Ubuntu, как и всё доступное в этой системе программное обеспечение, можно безо всяких ограничений абсолютно бесплатно и на совершенно законных основаниях. Мало того, можно даже скачать исходный код всех компонентов системы и сделать на его основе собственный продукт.

Ubuntu поддерживается и спонсируется Canonical, однако огромный вклад в развитие этой замечательной операционной системы вносит сообщество – обычные люди, которые стремятся улучшить используемые ими приложения и инструменты. Возможно, и вы когда-нибудь захотите помочь сделать Ubuntu лучшей операционной системой и примите участие в работе сообщества.

Виртуальная машина VMwareWorkstation нужна для работы с несколькими операционными системами, а также для:

- тестирования программного обеспечения;
- запуска программного обеспечения, которое нельзя запустить на Windows;
- для демонстрации DDOS атаки.

VMwareWorkstation - это программное обеспечение для виртуализации, предназначенное для компьютеров с архитектурой x86 - 64, работающих под

управлением Microsoft Windows и Linux. Позволяет пользователю установить одну или несколько виртуальных машин на одном физическом компьютере и запускать их параллельно с ним. Каждая виртуальная машина может работать со своей собственной операционной системой, включая Microsoft Windows, Linux, BSD и MS-DOS. VMwareWorkstation разработана и продается VMware, подразделением корпорации EMC. VMwareWorkstation поддерживает порт USB 3.0.

Скачиваю дистрибутив на официальном сайте (рисунок 1.1).

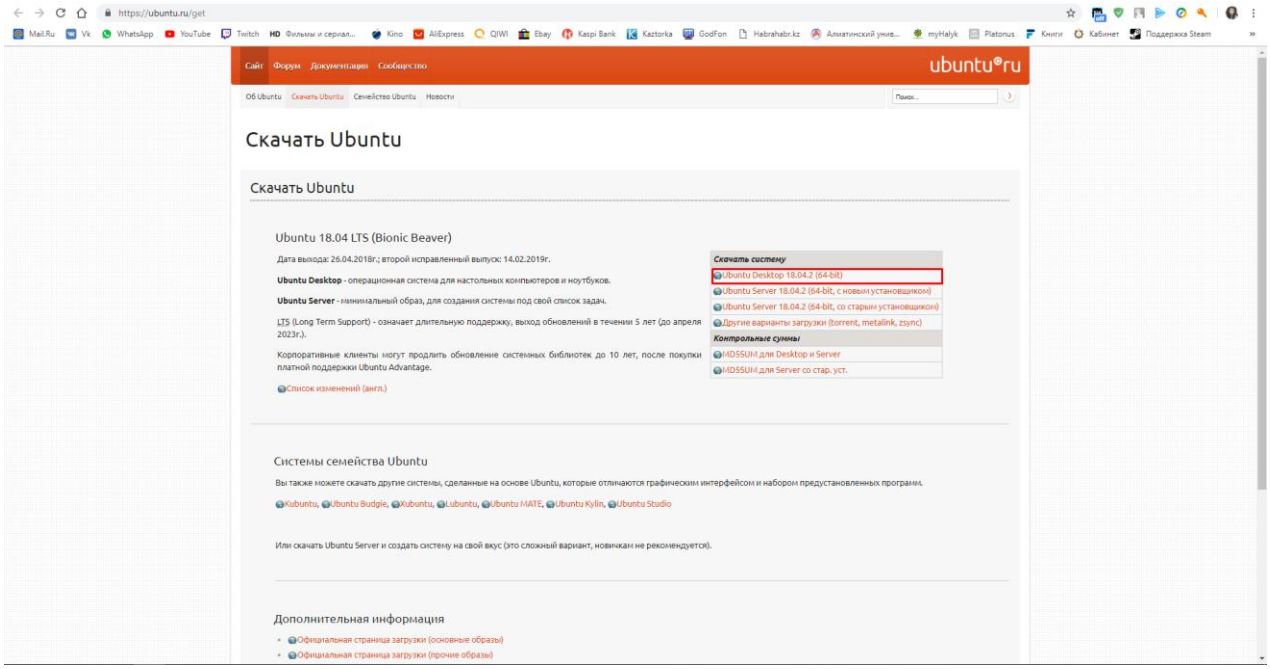

Рисунок 1.1 – Официальный сайт Ubuntu

Выбираю место для загрузки дистрибутива (рисунок 1.2).

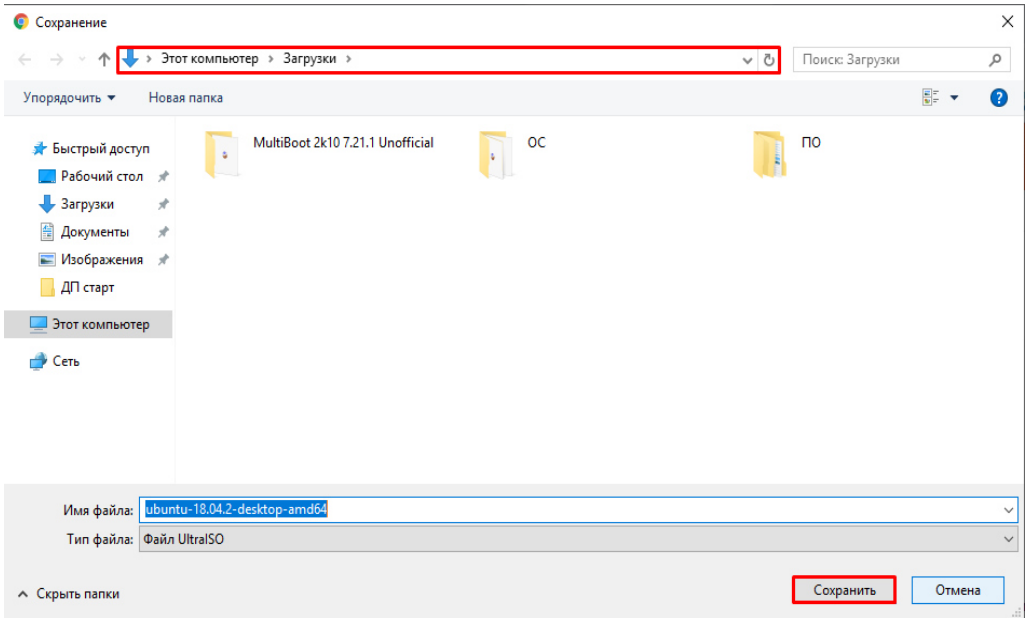

Рисунок 1.2 – Выбор места загрузки

После того, как я скачал нужный дистрибутив, начинаю устанавливать его на виртуальную машину.

Открываю виртуальную машину и создаю новую машину. В ней я буду проводить всю работу, а именно здесь будет функционировать сервер (рисунок 1.3).

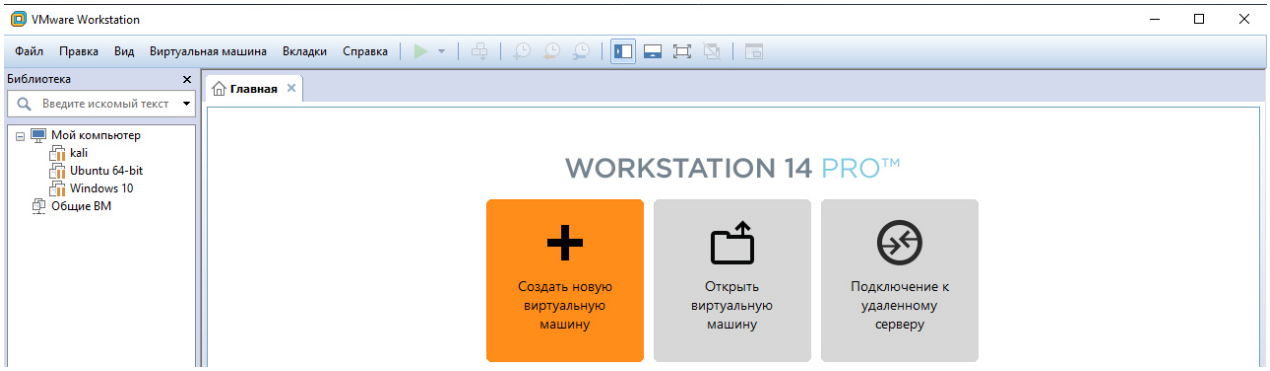

Рисунок 1.3 – Создание новой виртуальной машины

Выбираю тип конфигурации «обычный» (рисунок 1.4).

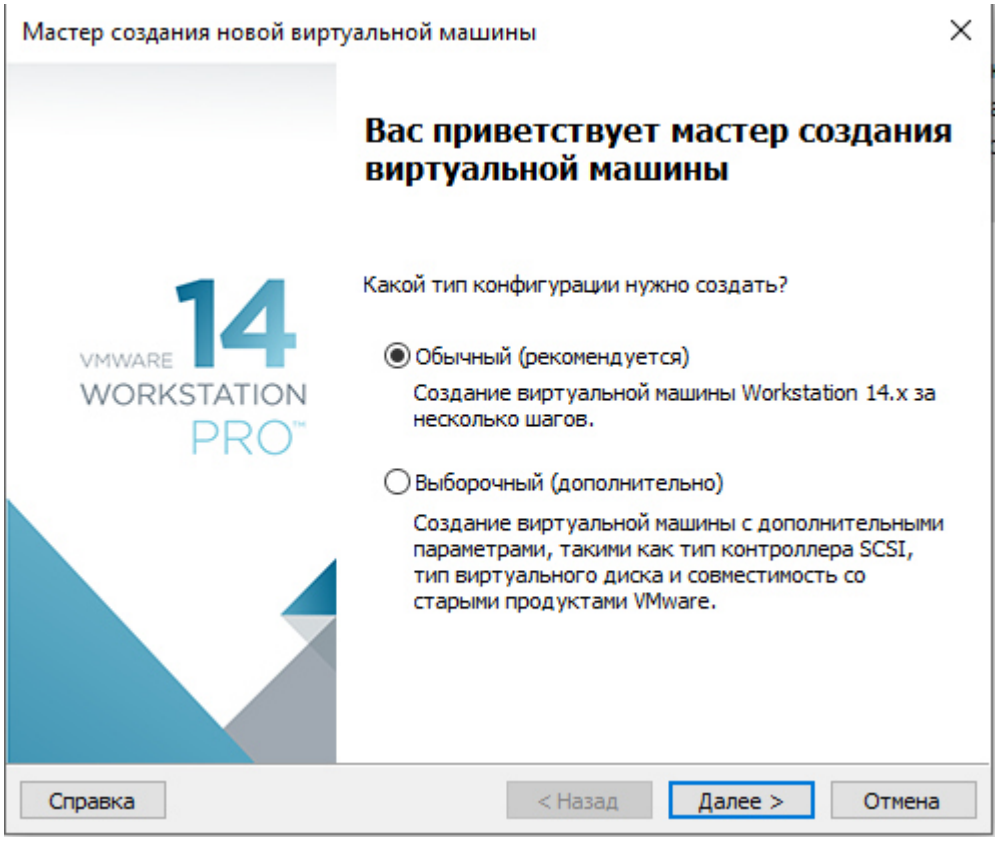

Рисунок 1.4 – Мастер создания виртуальной машины

Указываю ISO-образ установки (рисунок 1.5).

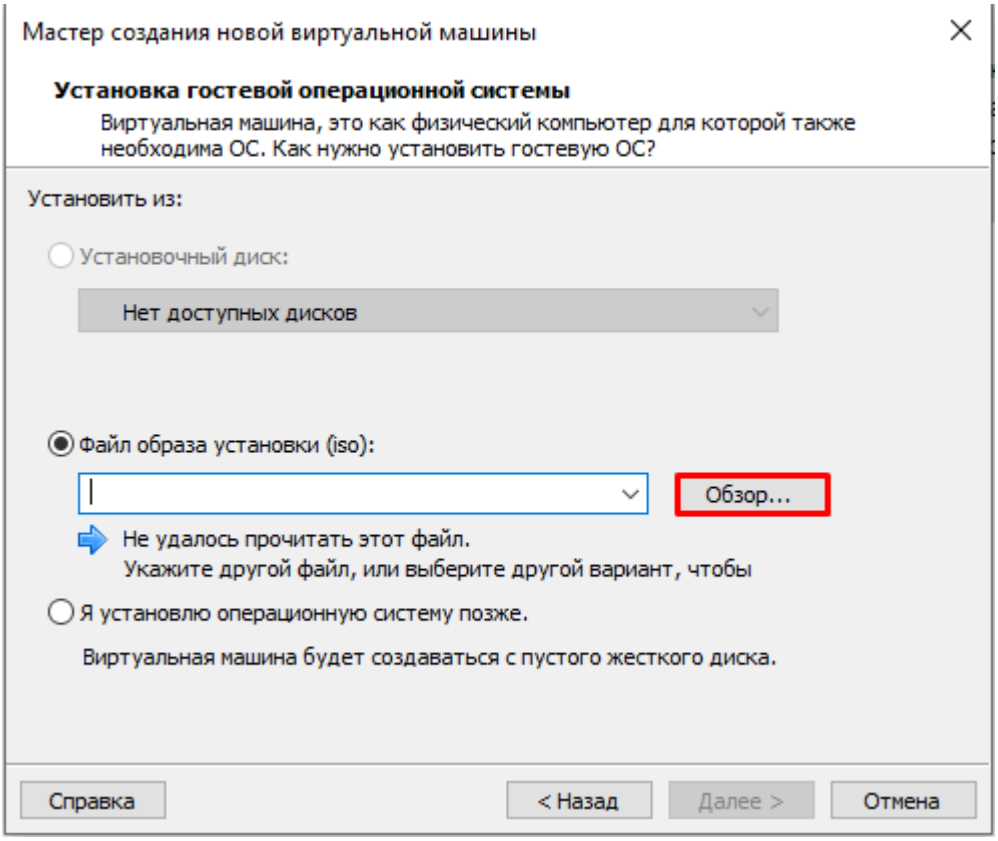

Рисунок 1.5 – Выбор ISO-образа

Выбираю ISO-образ для установки (рисунок 1.6).

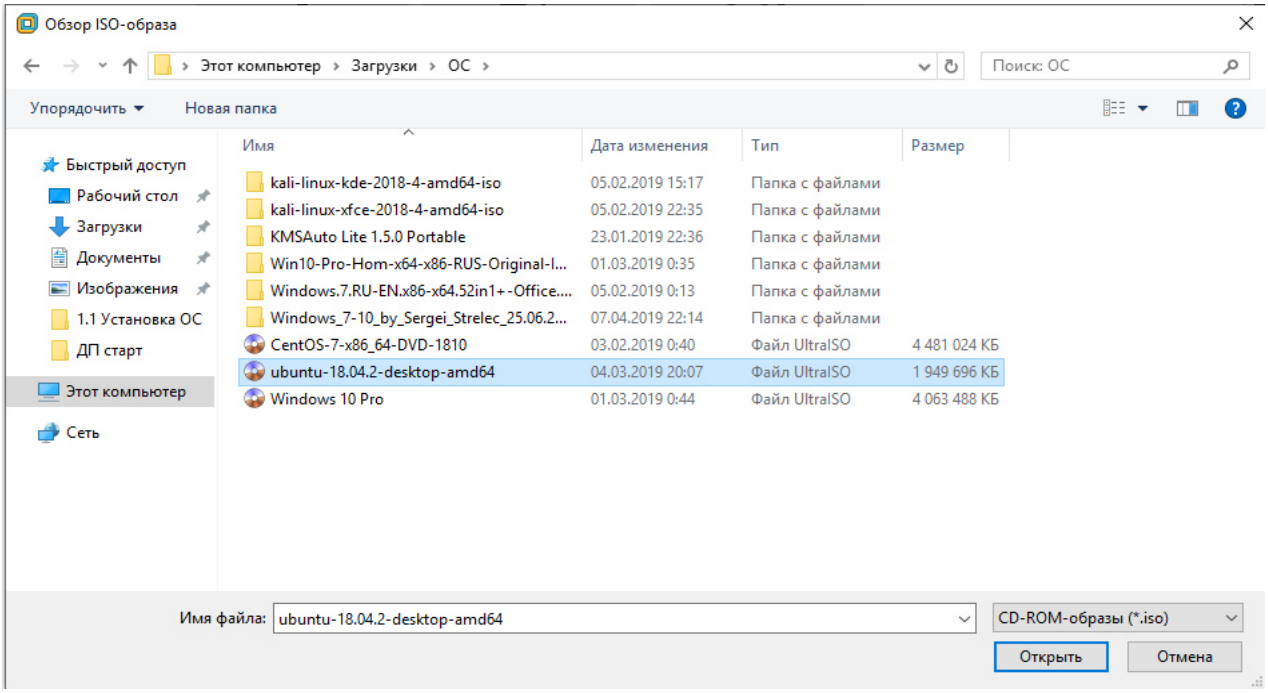

Рисунок 1.6 – Выбор ISO-образа

После того как ISO-образ был обнаружен, нажимаю «Далее» (рисунок 1.7).

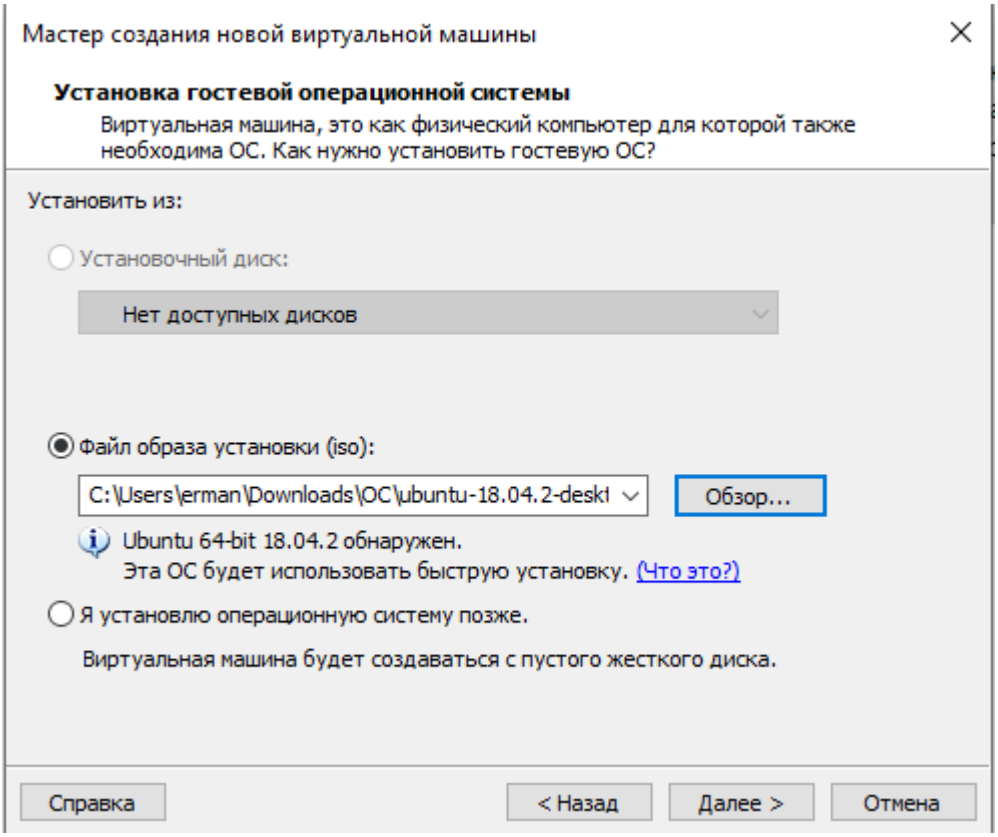

Рисунок 1.7 – Выбор ISO-образа

Ввожу данные для персонализации Linux (рисунок 1.8).

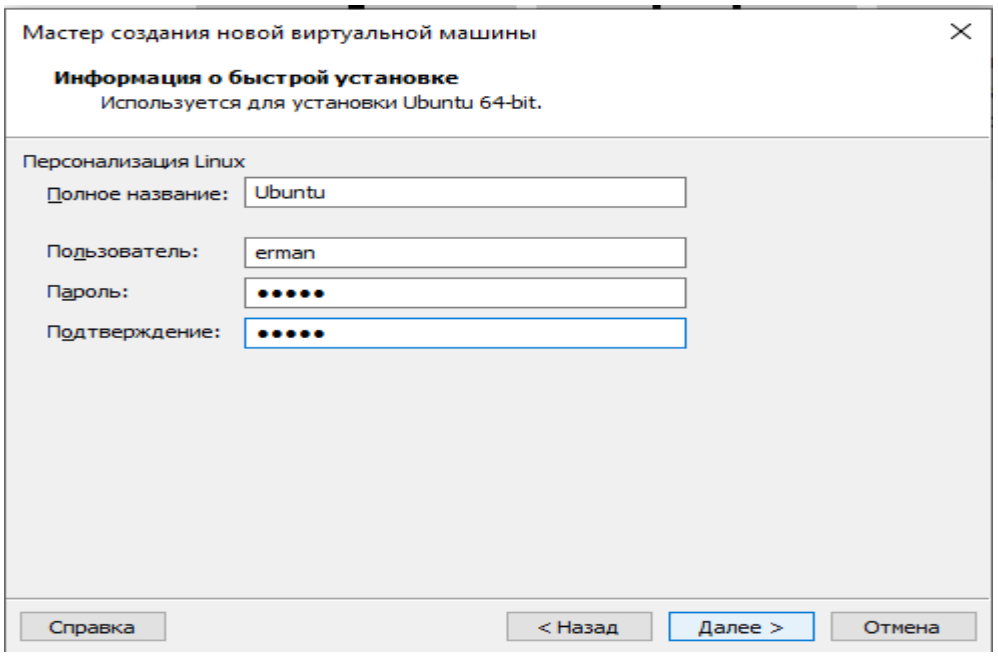

Рисунок 1.8 – Персонализация Linux

Ввожу имя виртуальной машины и указываю место расположения, где будет храниться виртуальная машина (рисунок 1.9).

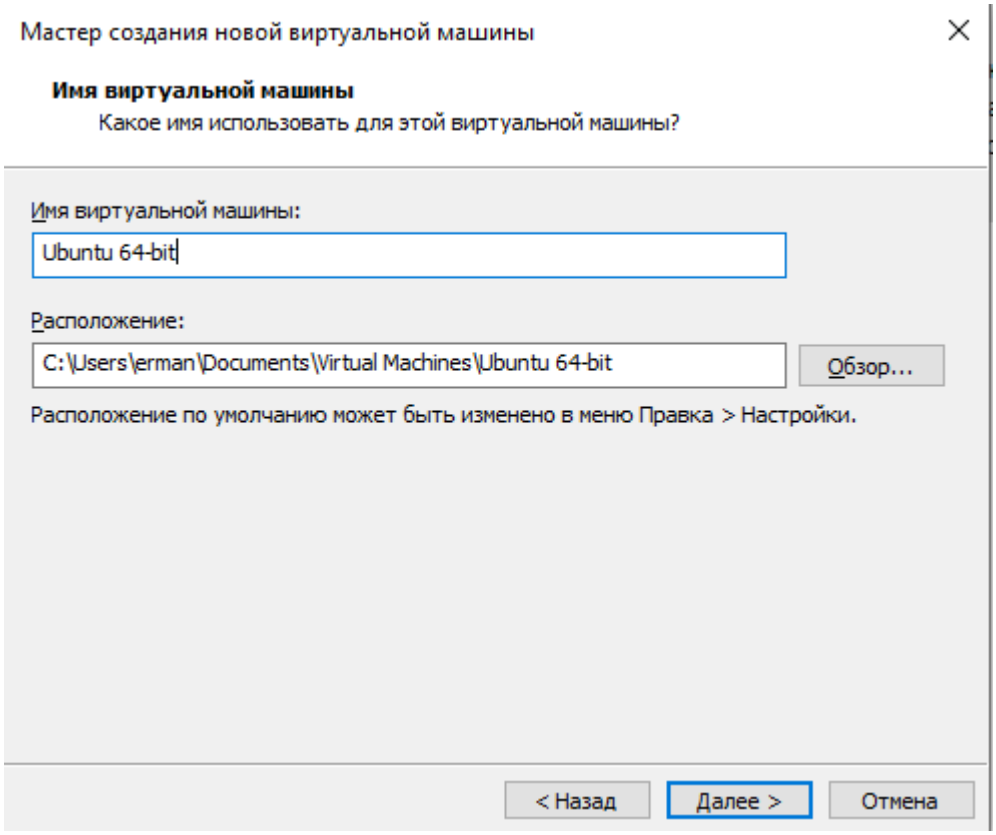

Рисунок 1.9 – Имя виртуальной машины и место расположения

Указываю размер диска для виртуальной машины и выбираю пункт «Разделить виртуальный диск на несколько файлов» (рисунок 1.10).

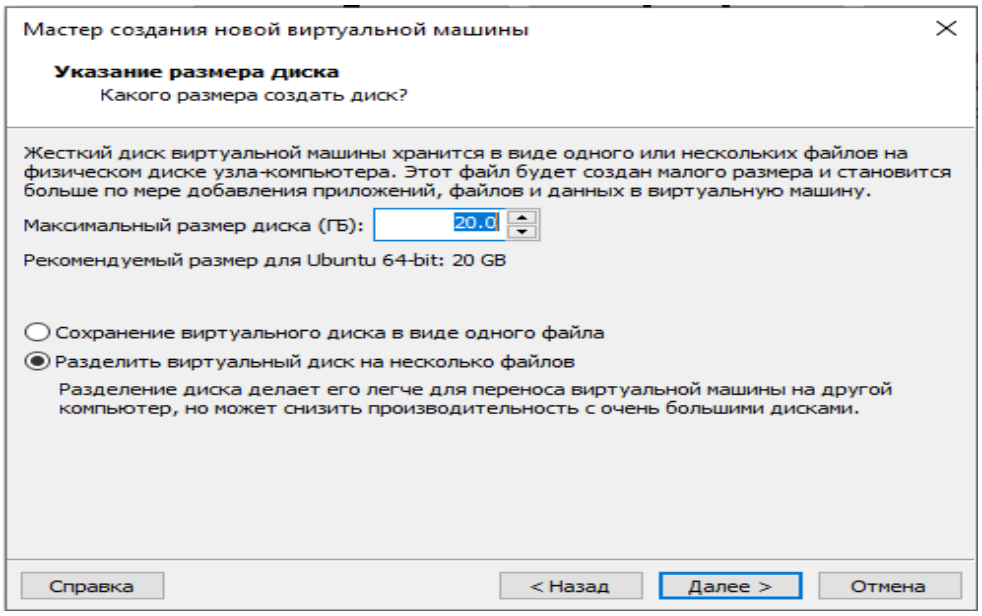

Рисунок 1.10 –Указание размера диска

Виртуальная машина автоматически задала минимальные характеристики для данной операционной системы, для меня эти характеристики не подходят, я настраиваю их вручную, выбрав пункт «Настройка оборудования» (рисунок 1.11).

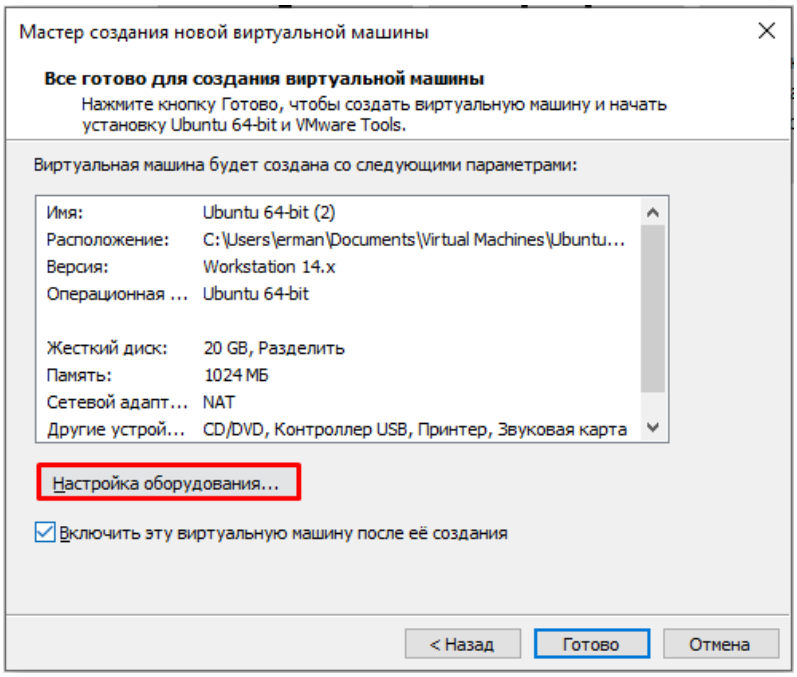

Рисунок 1.11 – Настройка оборудования

Выбрав пункт «Память», я увеличил его до 2 GB (рисунок 1.12).

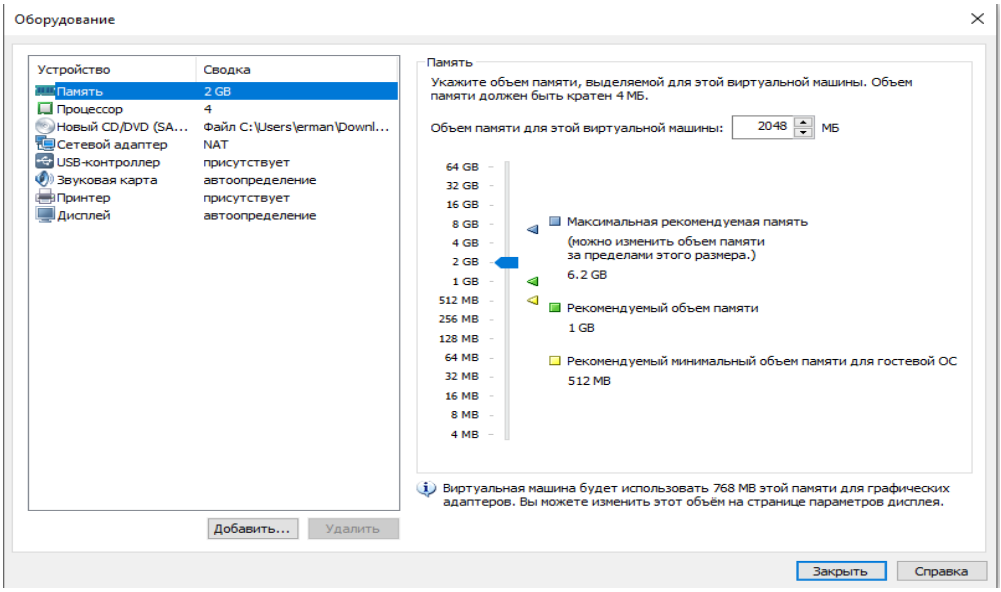

Рисунок 1.12 – Настройка оборудования

Закончив с настройкой других характеристик виртуальной машины, нажимаю «Готово» для установки виртуальной машины (рисунок 1.13).

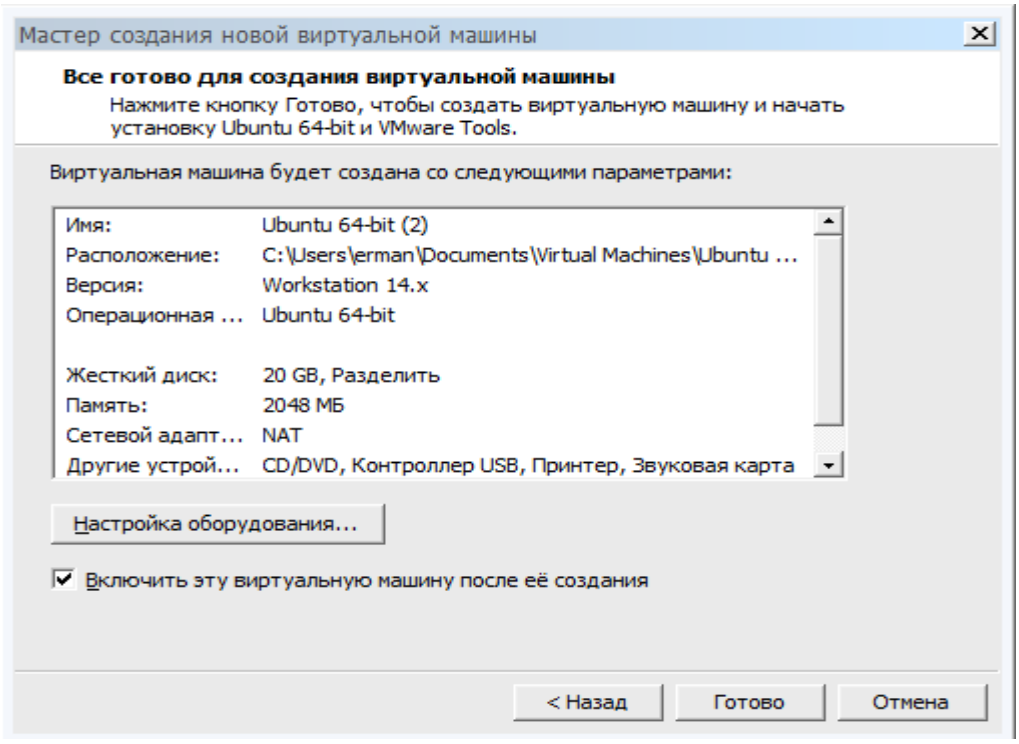

Рисунок 1.13 – Настройка оборудования

После всех этих процедур я создал машину в виртуальной машине. Дальше я запускаю виртуальную машину и начинаю установку (рисунок 1.14).

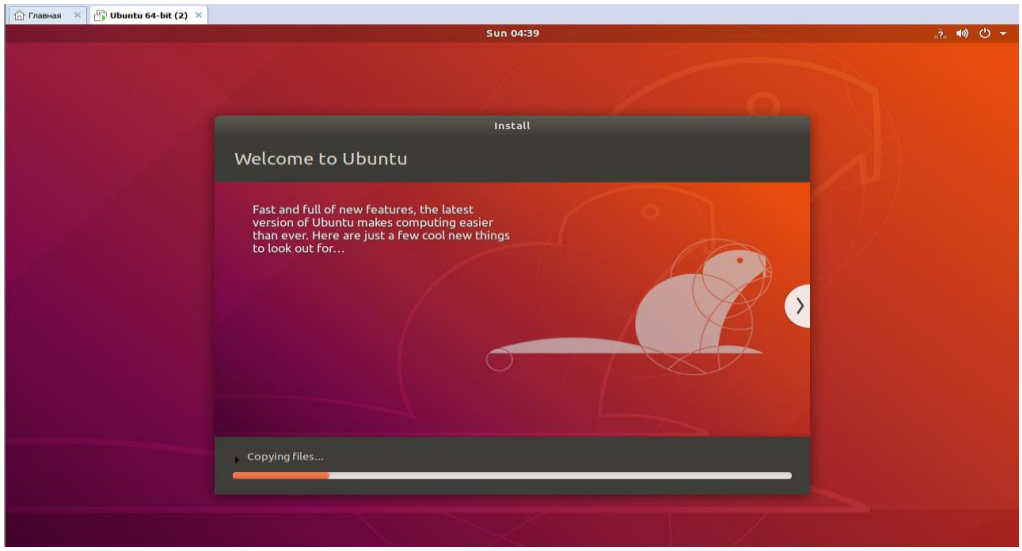

Рисунок 1.14 –Установка операционной системы

Рабочий стол установленной операционной системы показан на рисунке (рисунок 1.15).

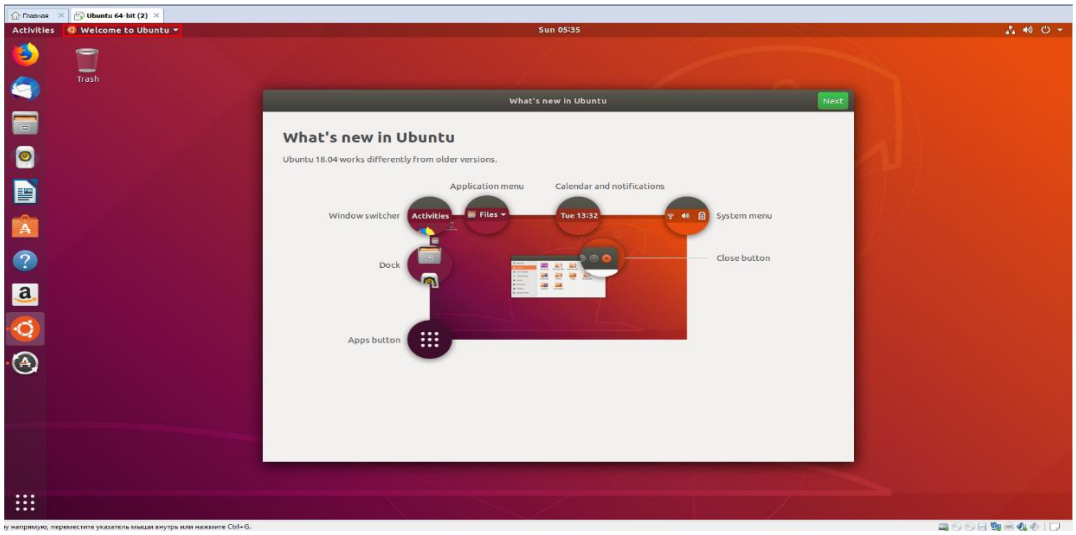

Рисунок 1.15 – Рабочий стол установленной системы

1.3 Поднятие сервера

Открываю терминал, нажимаю на правую кнопку мыши на рабочем столе и выбираю пункт «Open Terminal» (рисунок 1.16).

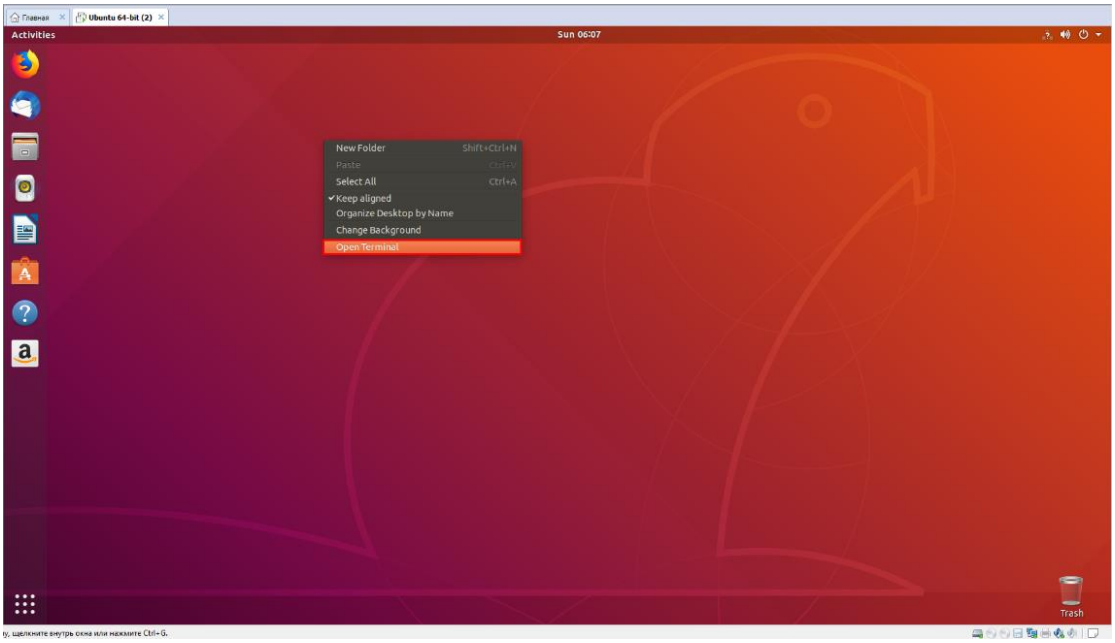

Рисунок 1.16 – Открытие терминала

Авторизуюсь под пользователем «root». С суперправами для дальнейшей полной настройки (рисунок 1.17).

```
erman@ubuntu:~$ sudo su
[sudo] password for erman:
root@ubuntu:/home/erman#
```

Рисунок 1.17 – Авторизация под root

Обновляю локальный индекс пакетов (рисунок 1.18).

```
root@ubuntu:/home/erman# sudo apt update
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Fetched 252 kB in 2s (119 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ubuntu:/home/erman#
```

Рисунок 1.18 – Обновление локального индекса пакетов

Устанавливаю пакет Apache2 (рисунок 1.19).

```
root@ubuntu:/home/erman# sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
0 upgraded, 9 newly installed, 0 to remove and 2 not upgraded.
Need to get 1,713 kB of archives.
After this operation, 6,920 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Рисунок 1.19 – Установка пакета Apache2

Проверяю список приложений пользователя (рисунок 1.20).

```
root@ubuntu:/home/erman# sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
```

Рисунок 1.20 – Список приложений пользователей

Как видно из рисунка для Apache доступно три профиля:

1) Apache: этот профиль открывает порт 80 (не шифрованный веб-трафик).
2) ApacheFull: этот профиль открывает порты 80 (не шифрованный веб-трафик) и 443 (трафик шифруется с помощью TLS/SSL).
3) ApacheSecure: этот профиль открывает только порт 443 (трафик шифруется с помощью TLS/SSL)

Выбираю профиль «Apache», потому что я не настраивал SSL для сервера поэтому включаю только порт 80 (рисунок 1.21).

```
root@ubuntu:/home/erman# sudo ufw allow 'Apache'
Rules updated
Rules updated (v6)
```

Рисунок 1.21 – Включение профиля

Проверяю статус соединений, что HTTP трафик разрешен (рисунок 1.22).

```
root@ubuntu:/home/erman# sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache                     ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)
```

Рисунок 1.22 – Статус UFW

Проверяю статус сервера Apache2 (рисунок 1.23).

```
root@ubuntu:/home/erman# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Sun 2019-04-28 06:20:03 PDT; 4min 38s ago
  Process: 4325 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUCCESS
  Process: 4249 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/S
  Process: 4775 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCE
 Main PID: 4779 (apache2)
    Tasks: 55 (limit: 2311)
   CGroup: /system.slice/apache2.service
           ├─4779 /usr/sbin/apache2 -k start
           ├─4780 /usr/sbin/apache2 -k start
           └─4781 /usr/sbin/apache2 -k start

Apr 28 06:20:03 ubuntu systemd[1]: Starting The Apache HTTP Server...
Apr 28 06:20:03 ubuntu apachectl[4775]: AH00558: apache2: Could not reliably det
Apr 28 06:20:03 ubuntu systemd[1]: Started The Apache HTTP Server.
lines 1-18/18 (END)
```

Рисунок 1.23 – Проверка веб-сервера

Узнаю IP-адрес сервера (рисунок 1.24).

```
root@ubuntu:/home/erman# hostname -I
192.168.184.148
```

Рисунок 1.24 – IP-адрес сервера

Открываю браузер и перехожу по IP-адресу (рисунок 1.25).

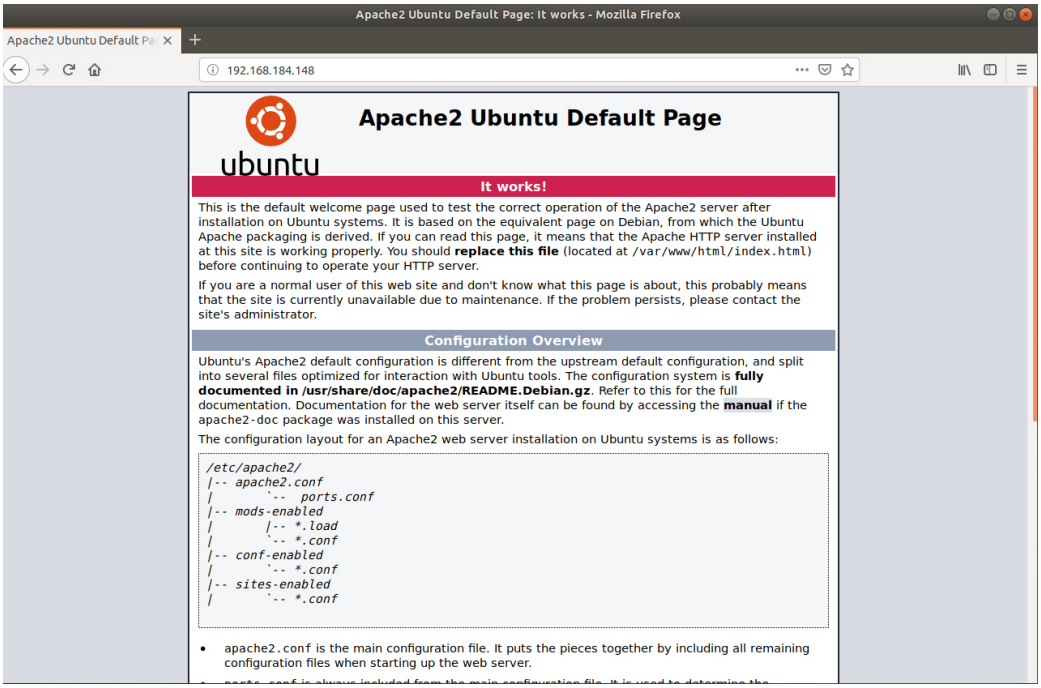

Рисунок 1.25 – Дефолтная страница Apache для Ubuntu

Устанавливаю FTP (рисунок 1.26).

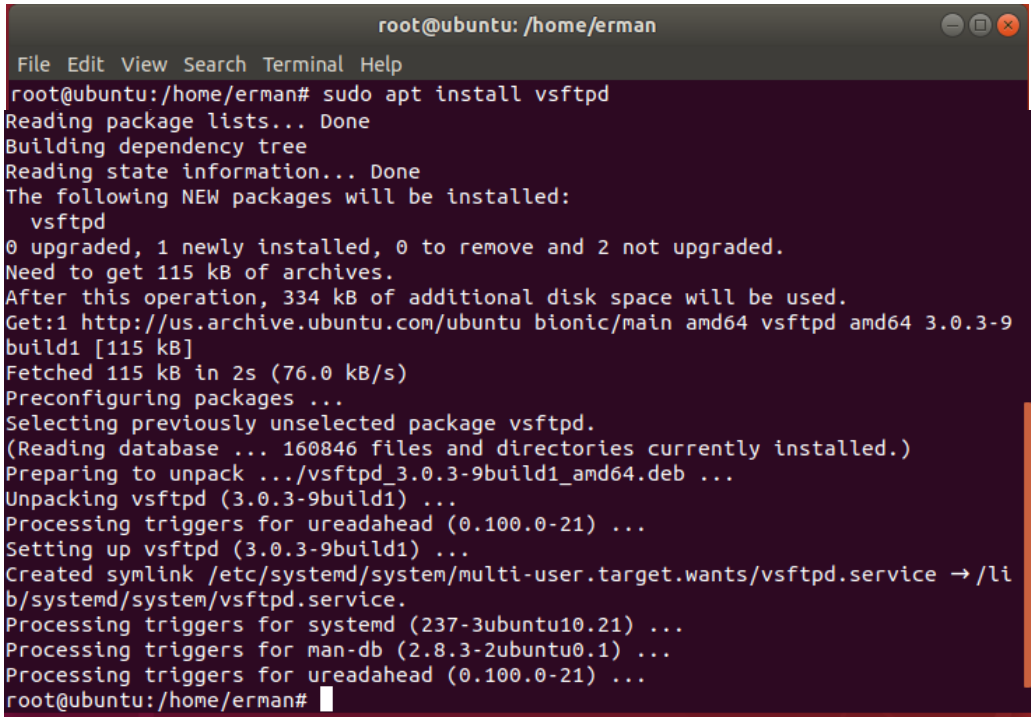

Рисунок 1.26 – Установка FTP

Разрешаю соединения по 20 и 21 порту (рисунок 1.27).

```
root@ubuntu:/home/erman# sudo ufw allow 20/tcp
Rule added
Rule added (v6)
root@ubuntu:/home/erman# sudo ufw allow 21/tcp
Rule added
Rule added (v6)
```

Рисунок 1.27 – Установка правил для соединения по 20 и 21

Проверяю статус разрешенных соединений (рисунок 1.28).

```
root@ubuntu:/home/erman# sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache                     ALLOW       Anywhere
20/tcp                     ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)
20/tcp (v6)                ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)

root@ubuntu:/home/erman#
```

Рисунок 1.28 – Статус файрволла ufw

Запускаю FTP (рисунок 1.29).

```
root@ubuntu:/home/erman# sudo systemctl start vsftpd
```

Рисунок 1.29 – Запуск FTP

Проверяю статус FTP сервера (рисунок 1.30).

```
root@ubuntu:/home/erman# sudo systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset: e
   Active: active (running) since Sun 2019-04-28 06:37:59 PDT; 1min 25s ago
 Main PID: 5601 (vsftpd)
    Tasks: 1 (limit: 2311)
   CGroup: /system.slice/vsftpd.service
           └─5601 /usr/sbin/vsftpd /etc/vsftpd.conf

Apr 28 06:37:59 ubuntu systemd[1]: Starting vsftpd FTP server...
Apr 28 06:37:59 ubuntu systemd[1]: Started vsftpd FTP server.
lines 1-10/10 (END)
```

Рисунок 1.30 – Статус FTP

Вывод

В данной главе я рассмотрел основные понятия и определения, связанные с корпоративным сервером. Установил операционную систему Ubuntu и поднял на нем сервер также установил защиту на сервер, защитил аппаратное устройство от LiveCD монтирования.

Для защиты сервер Apache я использовал Nginx в качестве обратного прокси к Apache. Объединив два веб-сервера для большей эффективности, используя Nginx, как статический фронтенд и Apache — как Backend я защитил веб сервер от DDOS-атаки.

Для защиты сервера FTP я выбрал fail2ban. Он позволяет на основе анализа логов блокировать тех, кто злоупотребляет доступностью сервера по сети. Например, защитить почтовые ящики от взлома путем перебора паролей или многократного запроса какого-либо ресурса.

Для защиты аппаратного устройства от LiveCD монтирования установил пароль на вход в BIOS Setup. Тем самым злоумышленник не сможет загрузиться с LiveCD.

2 Конфигурирование и тестирование

2.1 Общее конфигурирование

Устанавливаю защиту на сервер Apache2.

Делаю Ngnix как front-end для Apache2. Статические файлы отдает nginx, а динамикой занимается Apache. На рисунке 2.1 представлена блок-схема слияние служб двух веб-серверов [3].

Apache и Nginx — два самых распространённых веб-сервера в мире. Оба они способны обслуживать веб-сайты под большими нагрузками. Каждый веб-сервер имеет свои преимущества и недостатки. Причины популярности каждого из серверов хорошо известны: мощность Apache и скорость Nginx. Тем не менее, оба сервера имеют недостатки: Apache имеет ограничения памяти сервера, в то время как Nginx, эффективный для статических файлов, нуждается в помощи php-fhm или аналогичных модулей для динамического контента.

Тем не менее можно объединить два веб-сервера для большей эффективности, используя Nginx, как статический фронтенд и Apache — как Backend.

В данной главе я рассматриваю процедуру установки и настройки работы двух веб-серверов с целью использования преимуществ каждого из них.

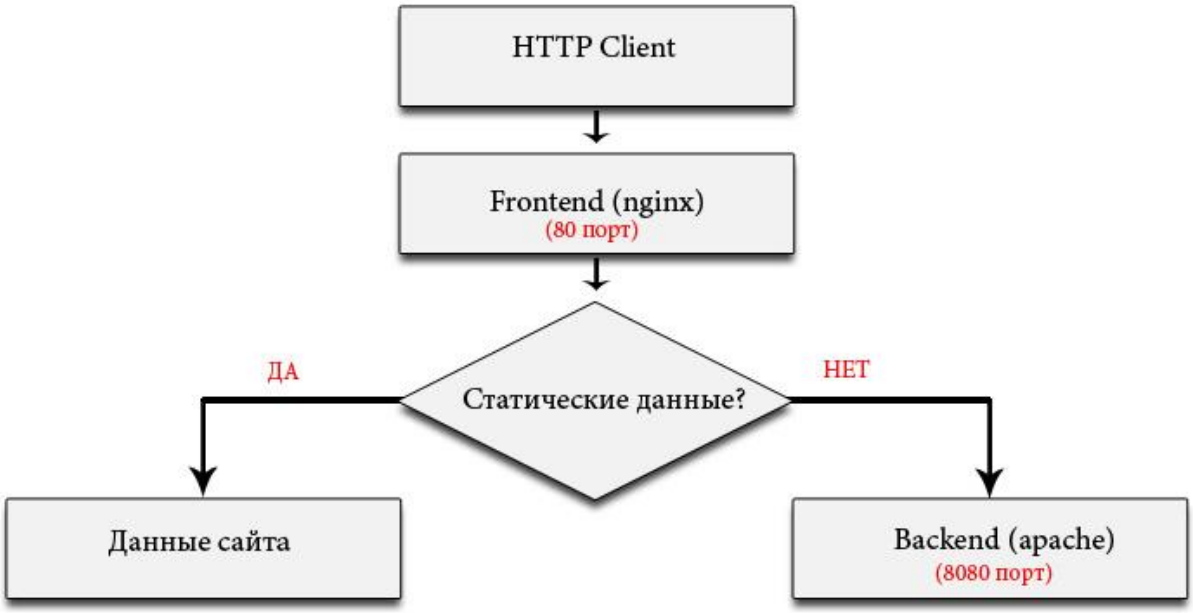

Рисунок 2.1 – Блок-схема работы слияния двух веб-серверов

Нужно перебросить Apache на порт 8080, поскольку на основном 80 порту соединения будет слушать Nginx. Для этого откроем конфигурационный файл портов Apache:

Открываю конфигурационный файл порта Apache2 (рисунок 2.2).

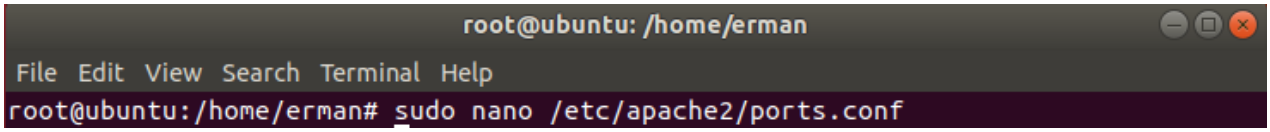

Рисунок 2.2 – Конфигурационный файл Apache2

Меняю Listen 80 на 8080, потому что на основном 80 порту соединения будет слушать Ngnix (рисунок 2.3).

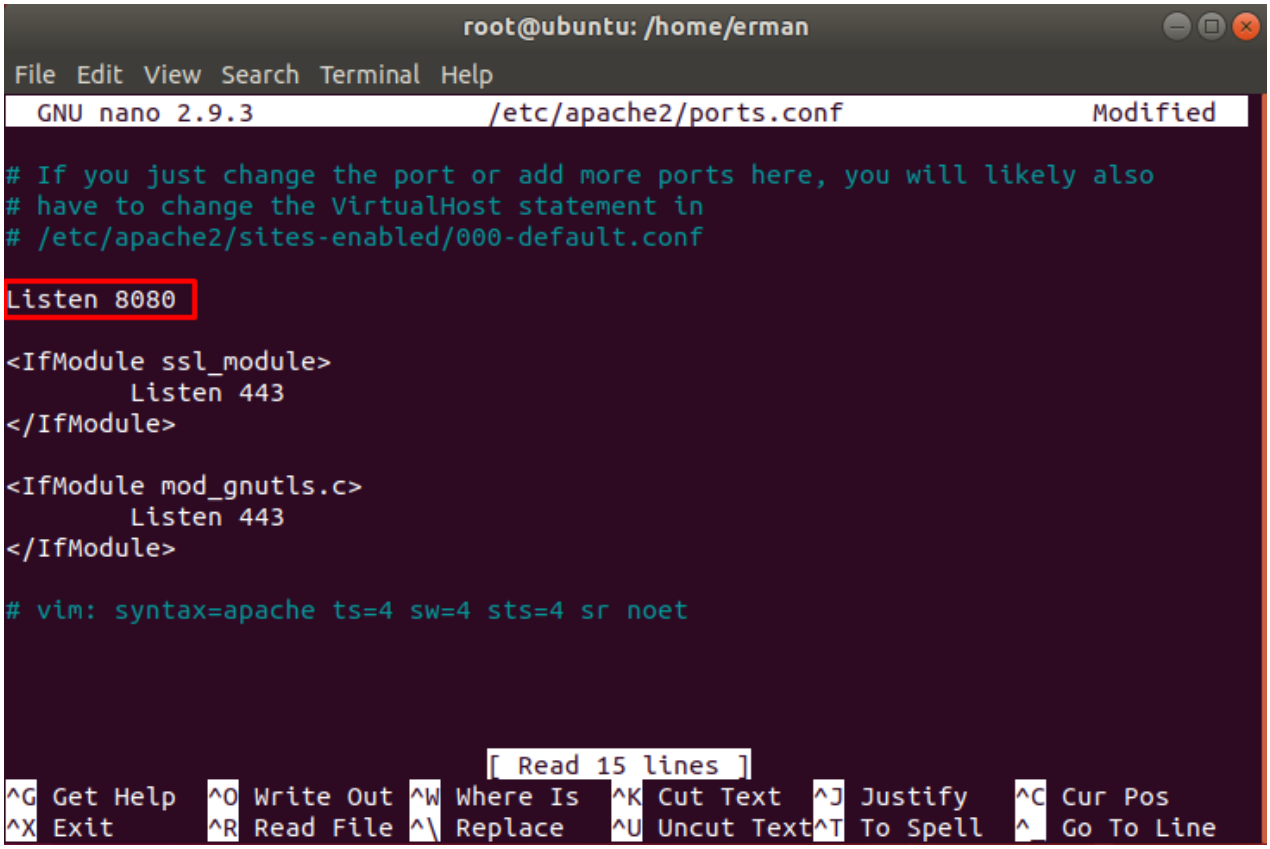

Рисунок 2.3 – Конфигурационный файл Apache

Ввожу команду для редактирования конфигурационного файла веб-сайта (рисунок 2.4).

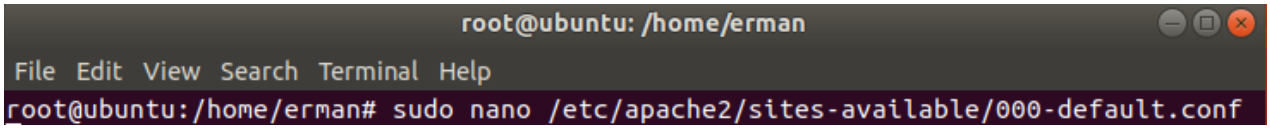

Рисунок 2.4 – Конфигурационный файл веб-сайта

Открываю конфигурационный файл веб-сайта (рисунок 2.5). Заменяю его содержимое на:

<VirtualHost *:8080>

ServerAdmin webmaster@localhost

DocumentRoot /var/www

Потому что на основном 80 порту соединения будет слушать Ngnix.

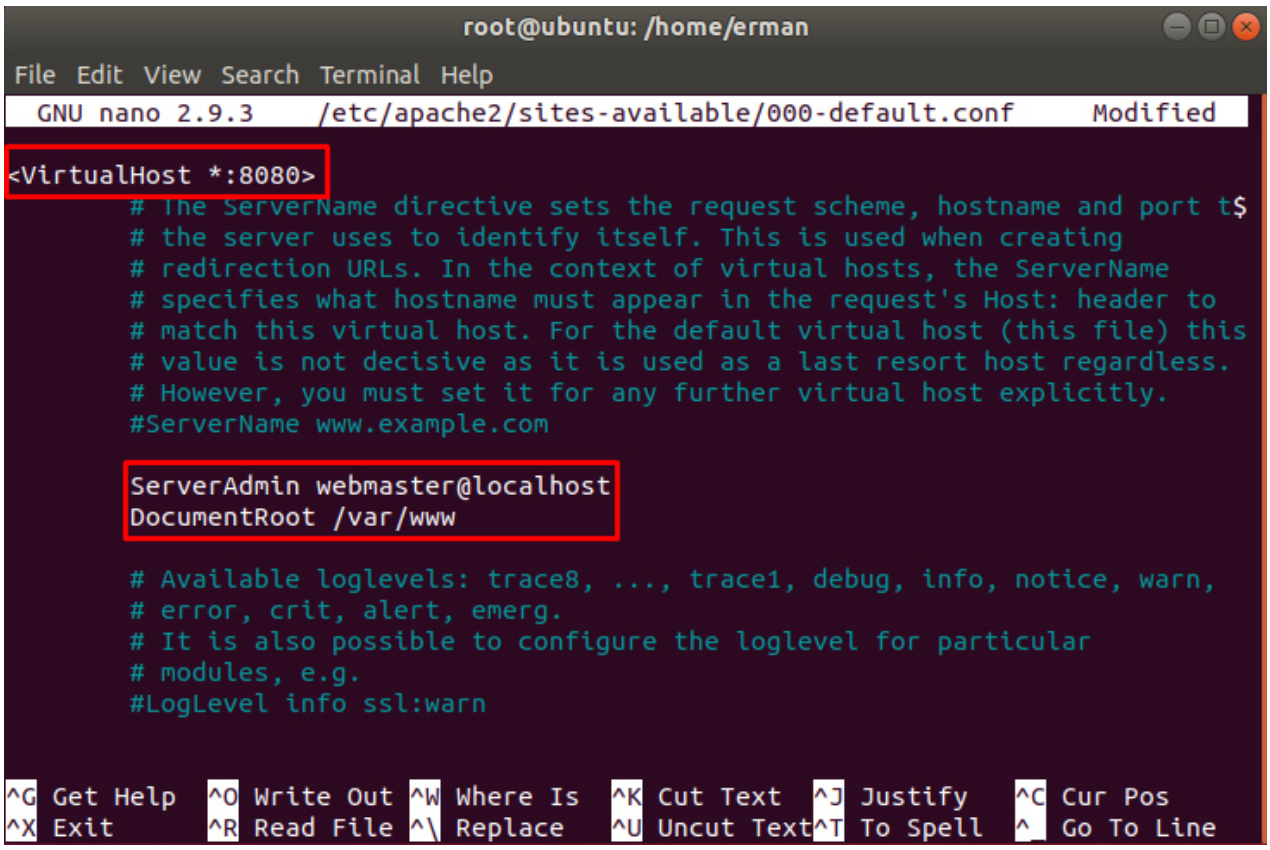

Рисунок 2.5 - Конфигурационный файл веб-сайта

Перезагружаю сервер, чтобы изменения вступили в силу (рисунок 2.6).

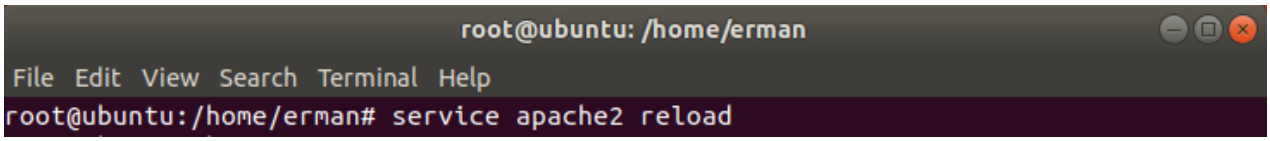

Рисунок 2.6 – Перезагрузка Apache

Убеждаюсь, что Apache переключился на порт 8080 (рисунок 2.7).

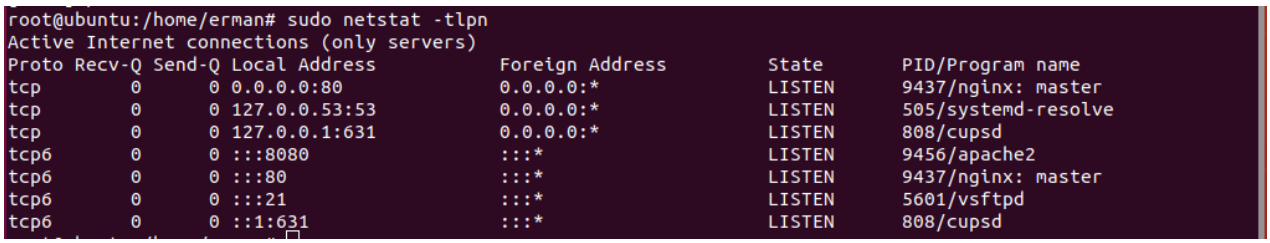

Рисунок 2.7 – Статус интернет-соединений

Устанавливаю Nginx (рисунок 2.8).

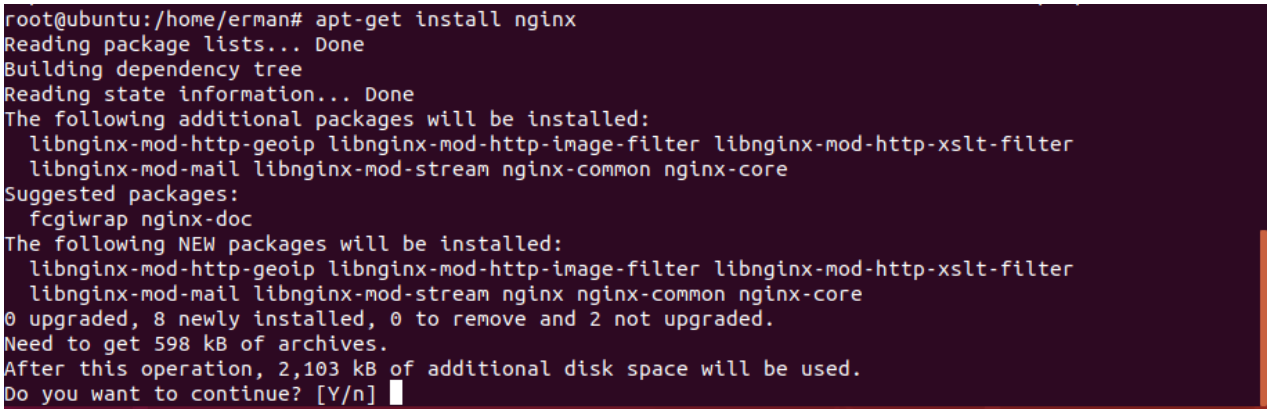

Рисунок 2.8 – Установка Nginx

Редактирую конфигурационный файл Ngnix (рисунок 2.9).

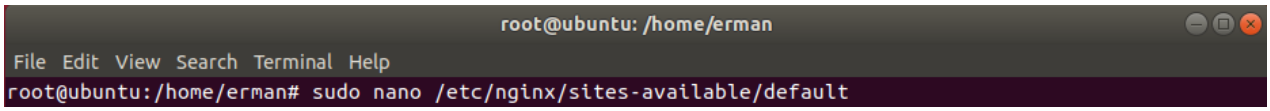

Рисунок 2.9 – Конфигурационный файл

Заменяю его содержимое (рисунок 2.10) на:

```
server {
        listen   80;
        index index.php index.htm index.html;
        server_name example.com;
        location / {
                try_files $uri $uri/ /index.php$args;
        }
        location ~ \.php$ {
                proxy_pass http://localhost:8080;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP  $remote_addr;
                proxy_set_header X-Forwarded-For $remote_addr;
                proxy_set_header Host $host;
                proxy_pass http://127.0.0.1:8080;
        }
        location ~ /\. {
                deny all;
        }
}
```

Потому что на основном 80 порту соединения будет слушать Nginx.

```
server {
        listen 80;
#       listen [::]:80;
#
#       server_name example.com;
#
#       root /var/www/example.com;
        index index.php index.htm index.html;
#
        location / {
                try_files $uri $uri/ /index.php;
        }

location ~ \.php$ {
 proxy_pass http://localhost:8080;
 proxy_set_header Host $host;
 proxy_set_header X-Real-IP $remote_addr;
 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 proxy_set_header X-Forwarded-Proto $scheme;
 }

 location ~ /\. {
 deny all; #запрет доступа к .htaccess
 }
}
```

Рисунок 2.10 – Конфигурационный файл

Проверяю конфигурационный файл на валидность (рисунок 2.11).

```
root@ubuntu:/home/erman# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@ubuntu:/home/erman#
```

Рисунок 2.11 – Проверка конфигурационного файла на валидность

После подтверждения валидности конфигурационного файла перезагружаю Nginx, чтобы изменения вступили в силу (рисунок 2.12).

```
root@ubuntu:/home/erman# service nginx reload
```

Рисунок 2.12 – Перезагрузка Ngnix

На этом установка и настройка защиты завершена. Дальше делаю установку защиты для FTP-сервера [7]. Скачиваю fail2ban (рисунок 2.13).

```
root@ubuntu: /home/erman
File  Edit  View  Search  Terminal  Help
root@ubuntu:/home/erman# apt-get install fail2ban
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-pyinotify whois
Suggested packages:
  mailx monit sqlite3 python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban python3-pyinotify whois
0 upgraded, 3 newly installed, 0 to remove and 405 not upgr
aded.
Need to get 398 kB of archives.
After this operation, 2,110 kB of additional disk space wil
l be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/universe a
md64 fail2ban all 0.10.2-2 [329 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu bionic/main amd64
 python3-pyinotify all 0.9.6-1 [24.7 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu bionic/main amd64
 whois amd64 5.3.0 [43.7 kB]
85% [3 whois 0 B/43.7 kB 0%]
```

Рисунок 2.13 – Установка fail2ban

Смотрю список файлов в каталоге fail2ban (рисунок 2.14).

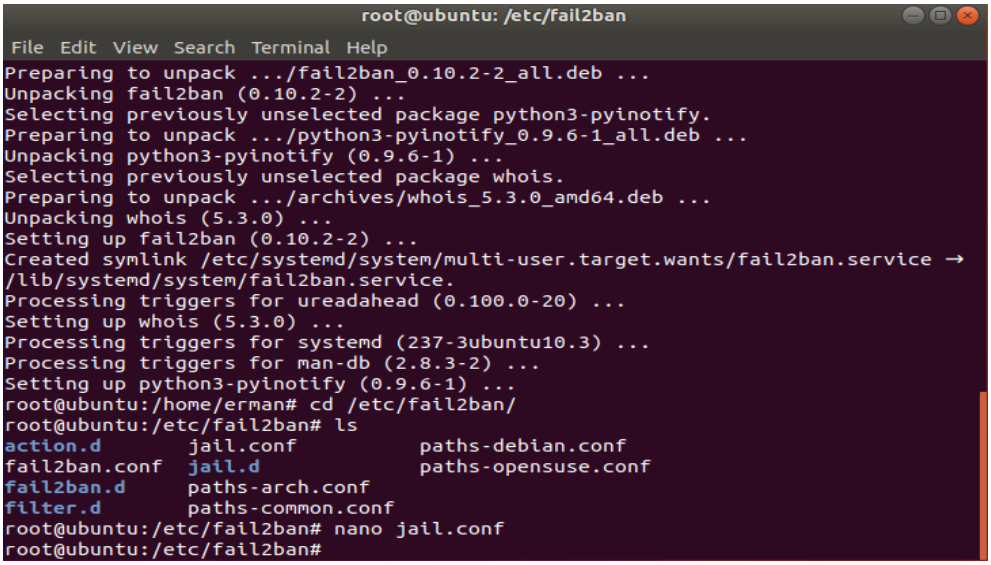

Рисунок 2.14 – Каталог файлов fail2ban

Для защиты FTP-сервера с помощью fail2ban можно использовать следующие параметры:

[vsftpd]
enabled = true
port = ftp,ftp-data,ftps,ftps-data
filter = vsftpd
logpath = /var/log/vsftpd.log
action = iptables[name=VSFTPD, port=21, protocol=tcp]
bantime = 600
maxretry = 3
findtime = 1800

1) enabled – состояние (true/false) фильтра, показывающее, включен он или выключен;
2) bantime – время в секундах, в течение которого подозрительный IP-адрес будет заблокирован;
3) port – порт целевого сервиса. Принимается буквенное или цифирное обозначение;
4) filter – фильтр (критерий поиска), который будет использоваться для поиска подозрительных действий;
5) logpath – расположение лог-файла, в котором фильтр будет искать подозрительную активность на основе описанных критериев;
6) action – действие, совершаемое в случае срабатывания правила. В квадратных скобках указаны название для правила, сетевой порт и протокол для блокирования;
7) bantime – время, на которое будет блокироваться IP-адрес;
8) maxretry – количество действий, которые разрешено совершить до бана;
9) findtime – время в секундах, в течение которого учитывается maxretry.

Меняю параметры конфигурационного файла jail.conf (рисунок 2.15).

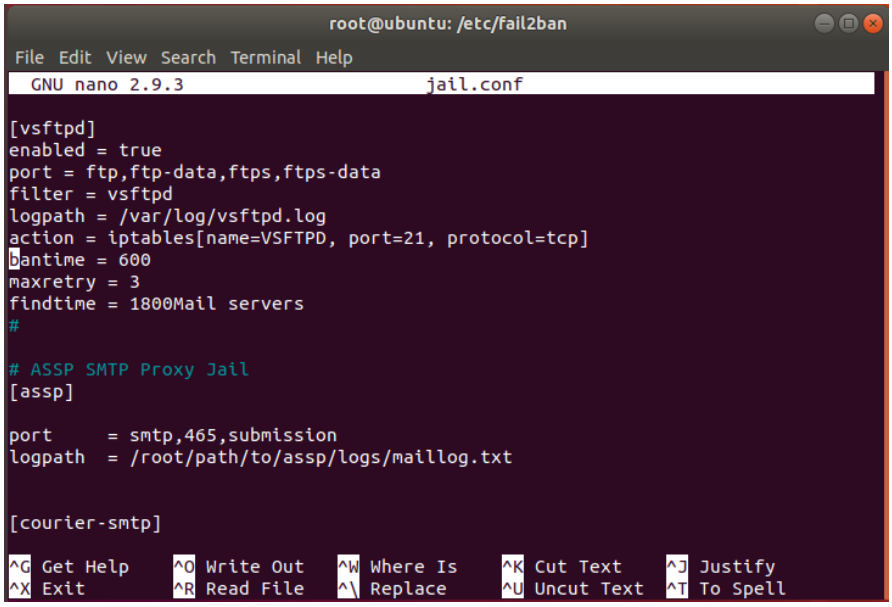

Рисунок 2.15 – Конфигурация файла jail.conf

2.2 Блокировка от LiveCD-монтирования

Злоумышленник может загрузиться с LiveCD, после чего с помощью команды chroot заменить корневую файловую систему LiveCD файловой системой сервера и опять получить над ним полный контроль. Для предотвращения таких действий надо установить пароль на вход в BIOS Setup, что не позволит злоумышленнику изменить порядок загрузки и загрузиться с LiveCD.

Установить пароль в начале операционной системы, нужно перезапустить или запустить (если компьютер выключен), нажать клавиши «Esc», «F2», «Delete» или которые указаны производителем для доступа к BIOS. Как только перехожу на вкладку «Security» (рисунок 2.16) и открываю строку «PasswordonBoot».

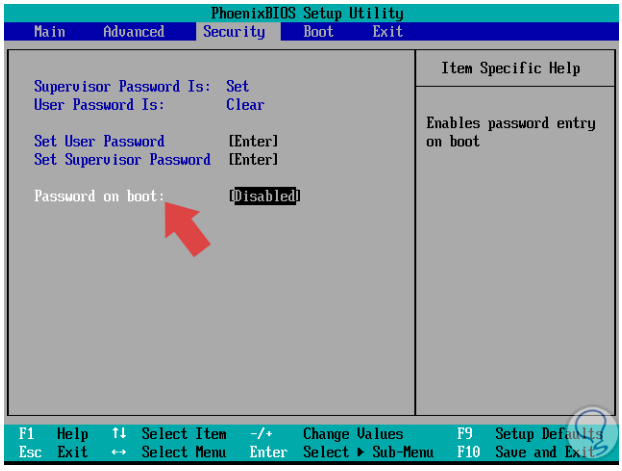

Рисунок 2.16 – Вкладка Security

Чтобы эта опция была активной, нужно определить пароль в строке «Set user password». Далее включаю параметр «Enabled» во вкладке «Password on boot» (рисунок 2.17).

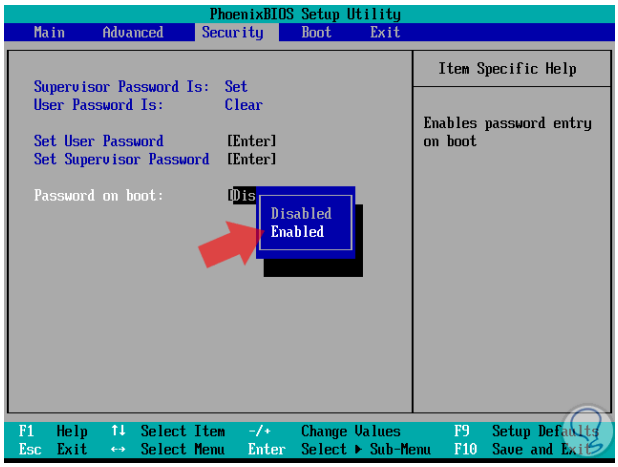

Рисунок 2.17 – Вкладка password on boot (enabled)

Нажатие «Enter» в этой строке и система отобразит следующий параметр, в котором выберу опцию «Enabled».

После того, как пароль включен, нажимаю клавишу «F10», чтобы сохранить изменения (рисунок 2.18).

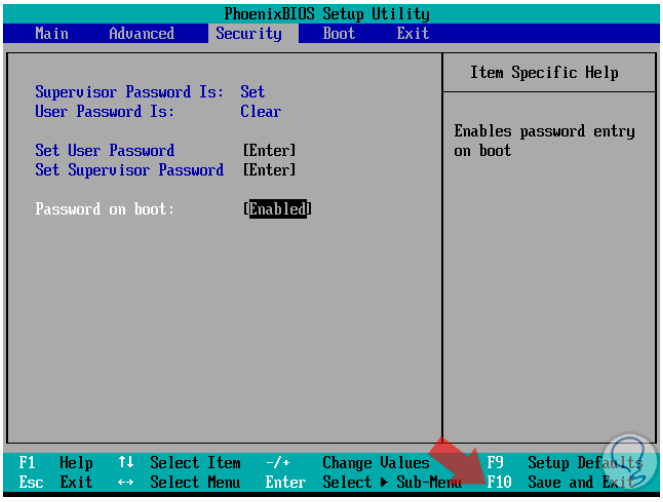

Рисунок 2.18 – Сохранение настроек

Таким образом, когда система запускается, фиксирую ввод пароля (рисунок 2.19):

Рисунок 2.19 – Авторизация в BIOS

2.3 Демонстрация атак

Для демонстрации атак понадобится Kali Linux. Первая цель будет сервер apache2. Для данной атаки использую скрипт xerxes.

Xerxes – мощный инструмент для DDOS-атаки с использованием Kali Linux. Для того, чтобы его скачать нужно перейти в терминал и прописать команду git clone и адрес https://gtihub.com/zanyajamal/xerxes (рисунок 2.20).

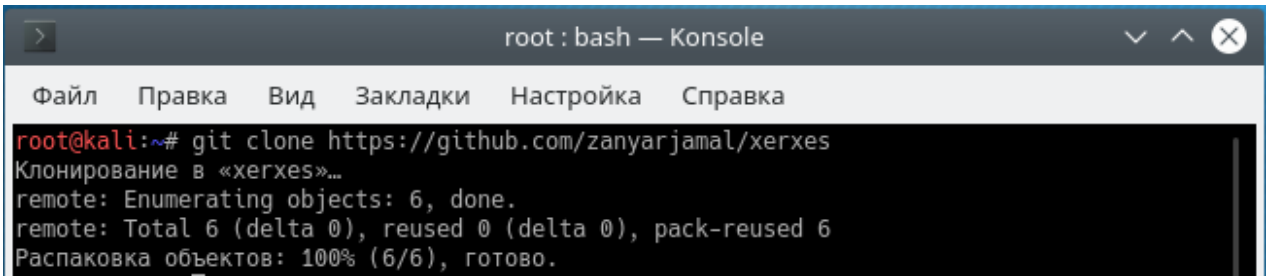

Рисунок 2.20 – Скачивание Xerxes

Перехожу в директорию xerxes (рисунок 2.21).

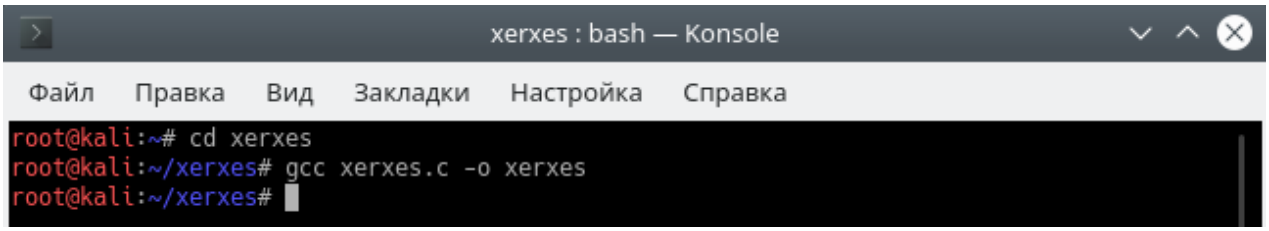

Рисунок 2.21 – Переход в директорию Xerxes

Запускаю DDOS атаку по IP-адресу (рисунок 2.22).

Рисунок 2.22 – Запуск DDOS атаки по IP-адресу

Демонстрация запущенной DDOS атаки (рисунок 2.23).

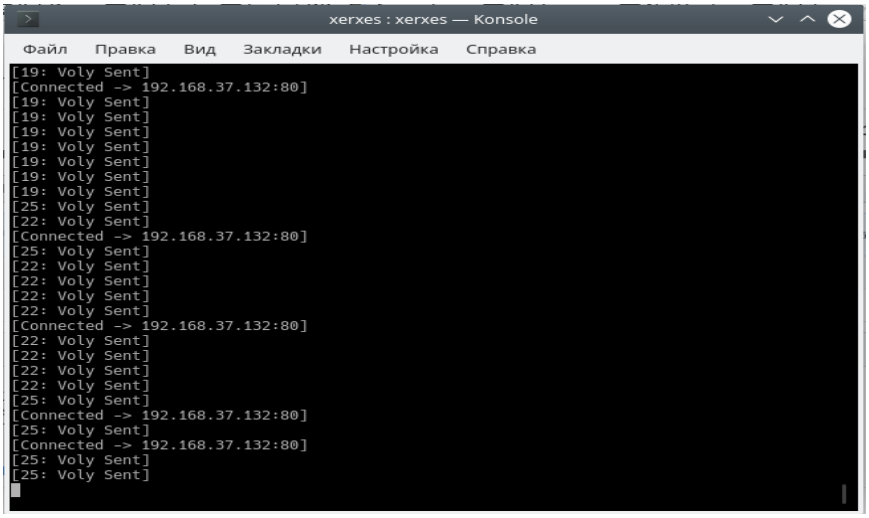

Рисунок 2.4 – Запущенная DDOS атака

Перехожу на сайт Apache (рисунок 2.24).

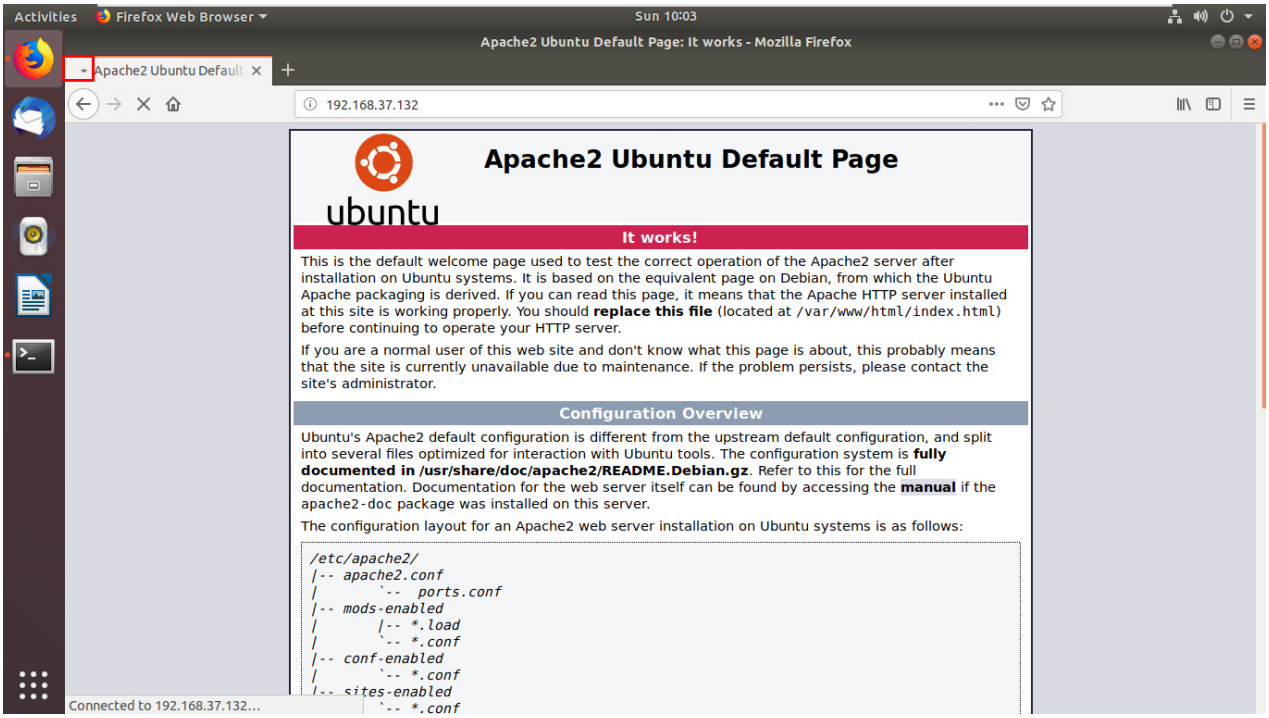

Рисунок 2.24 – Сайт Apache2

Видно, что сайт нагружен, подгружает изображение очень долго (рисунок 2.25).

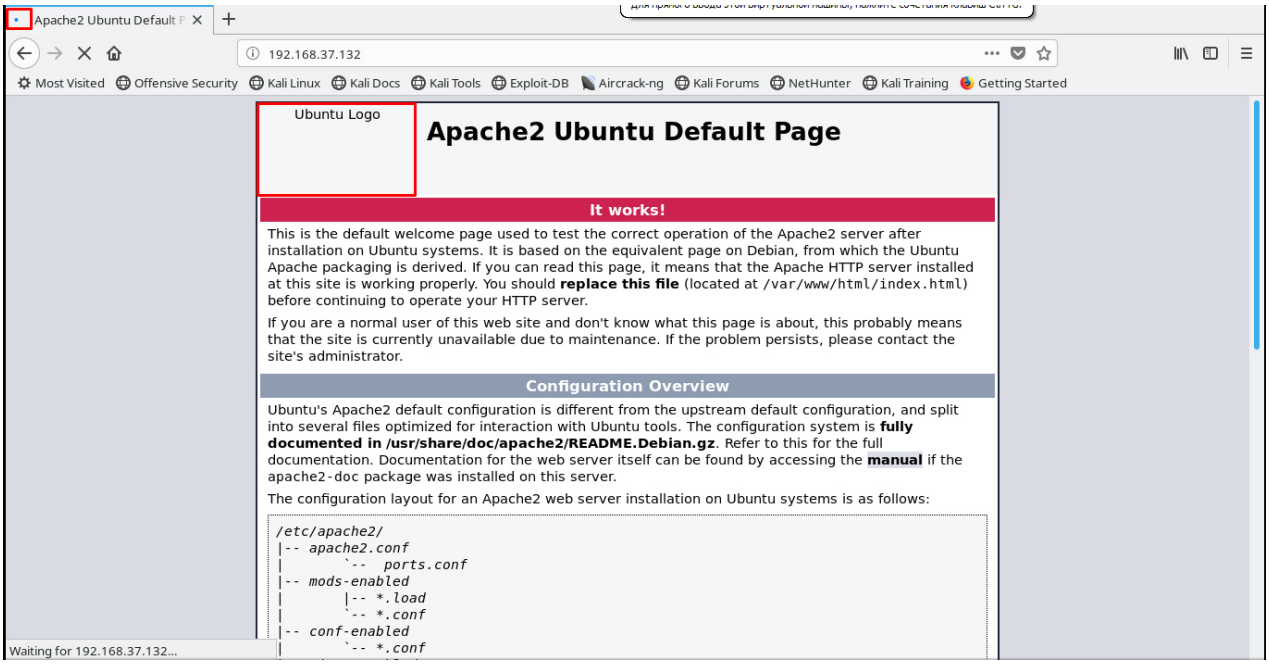

Рисунок 2.25 – Сайт Apache2

Вторая цель это FTP-сервер. Для того, чтобы взломать FTP-сервер буду тоже использовать Kali Linux. Понадобится программа Hydra (рисунок 2.26).

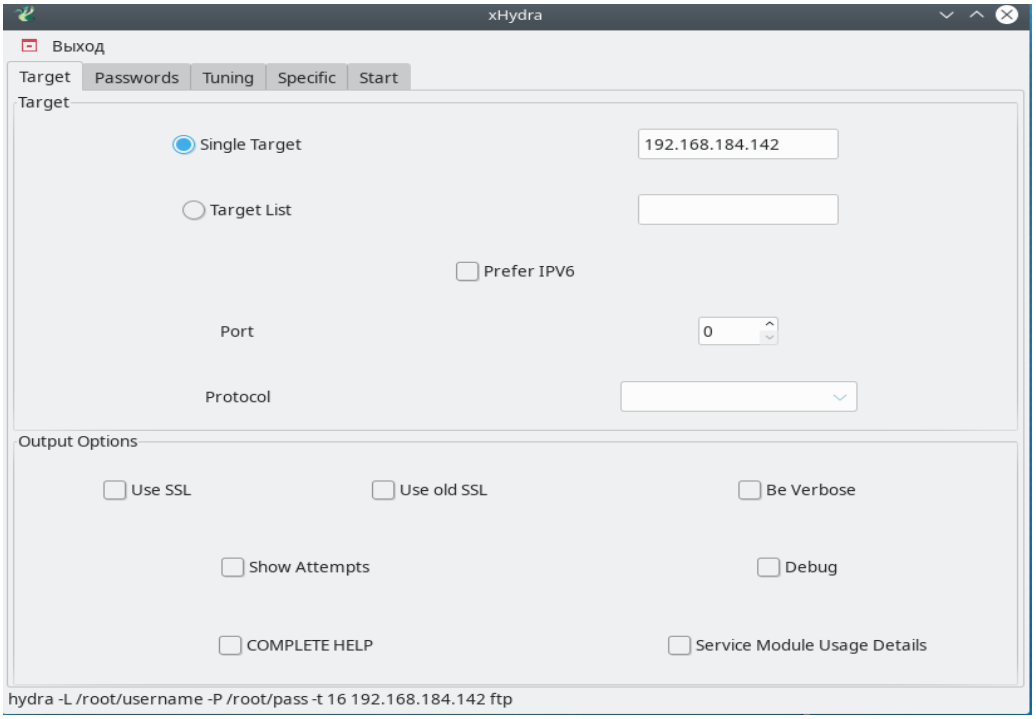

Рисунок 2.26 – Hydra, выбор цели

Указываю базу логинов и паролей (рисунок 2.27).

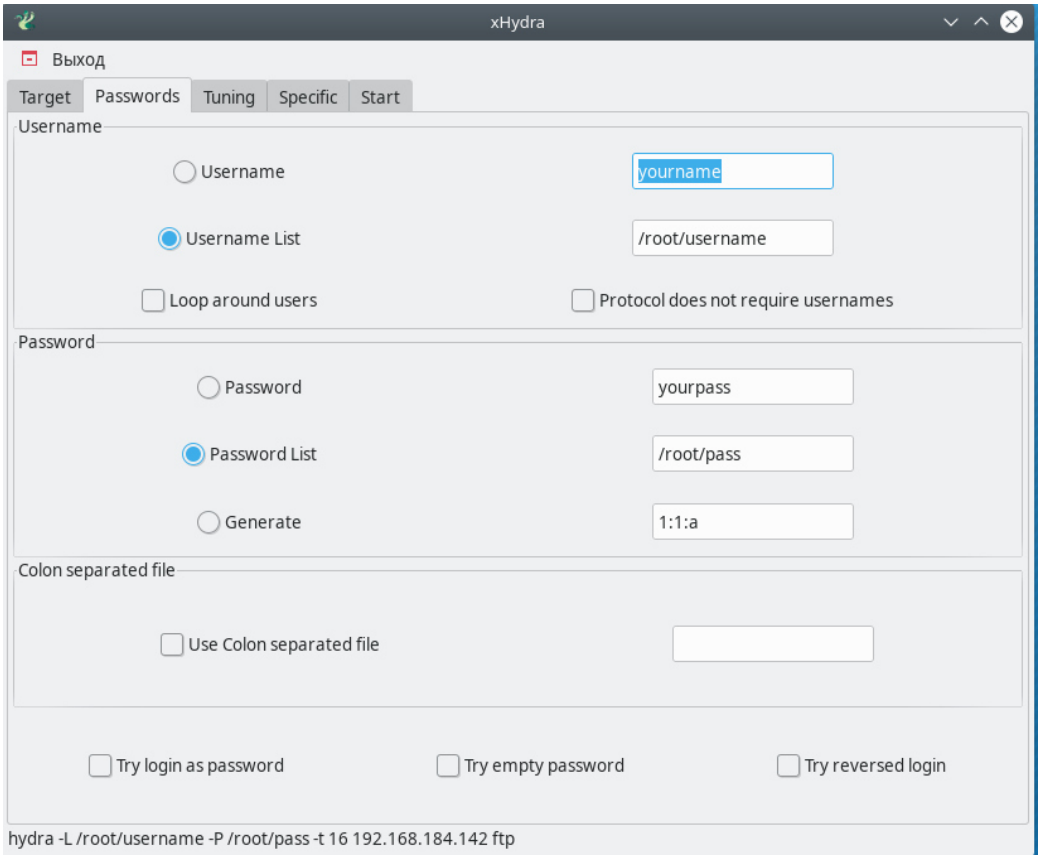

Рисунок 2.27 – Hydra, база логина и паролей

Вкладка «Start» позывает информацию о текущем bruteforce, что bruteforce удался и выводит информацию (рисунок 2.28).

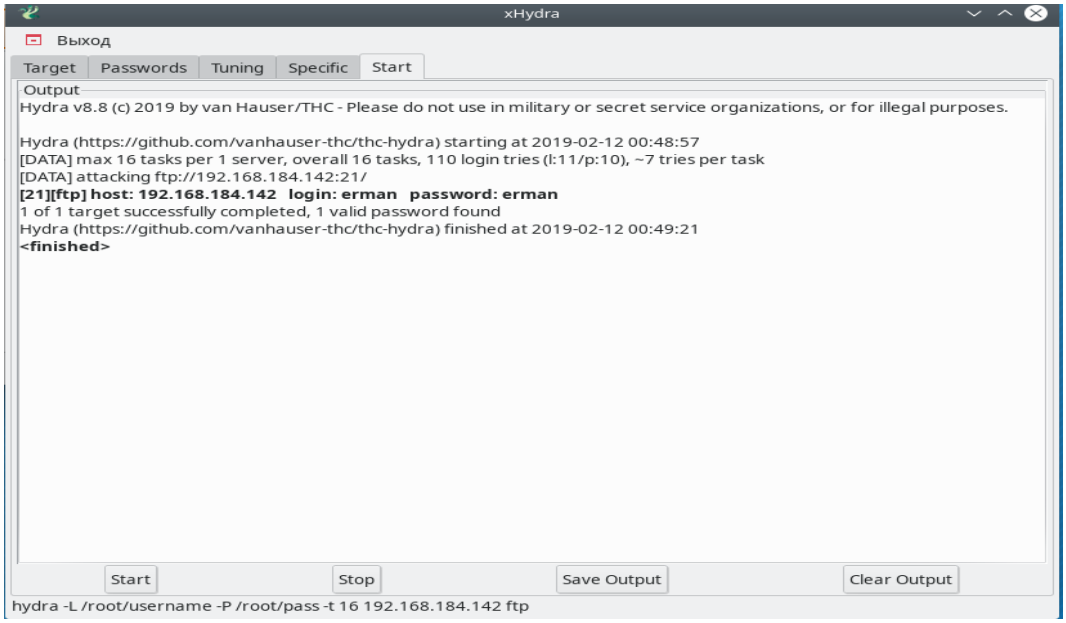

Рисунок 2.28 – Вкладка Start в Hydra

2.4 Анализ уязвимостей

Выполнив атаку на сервер Apache2, видно, что его стандартная защита уязвима внешним атакам. На рисунках 2.29 и 2.30 видно, что сервер не справляется с DDOS атакой по IP-адресу [4].

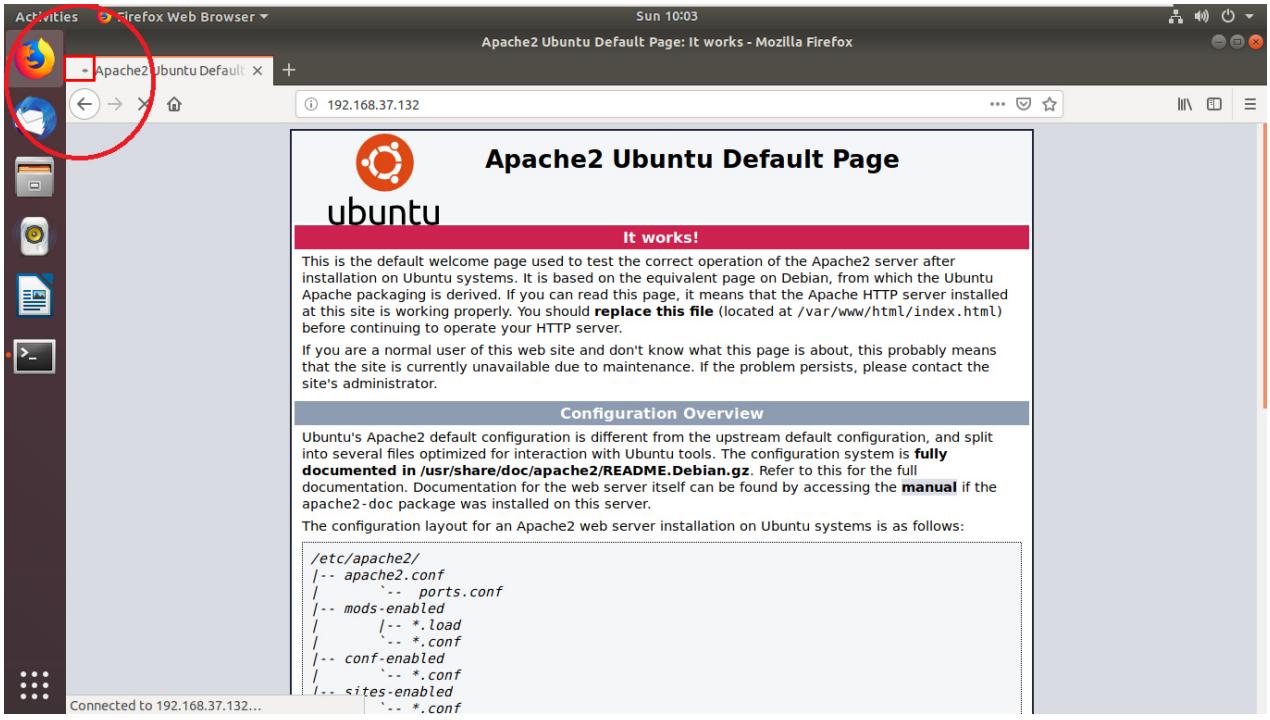

Рисунок 2.29 – Сайт Apache2

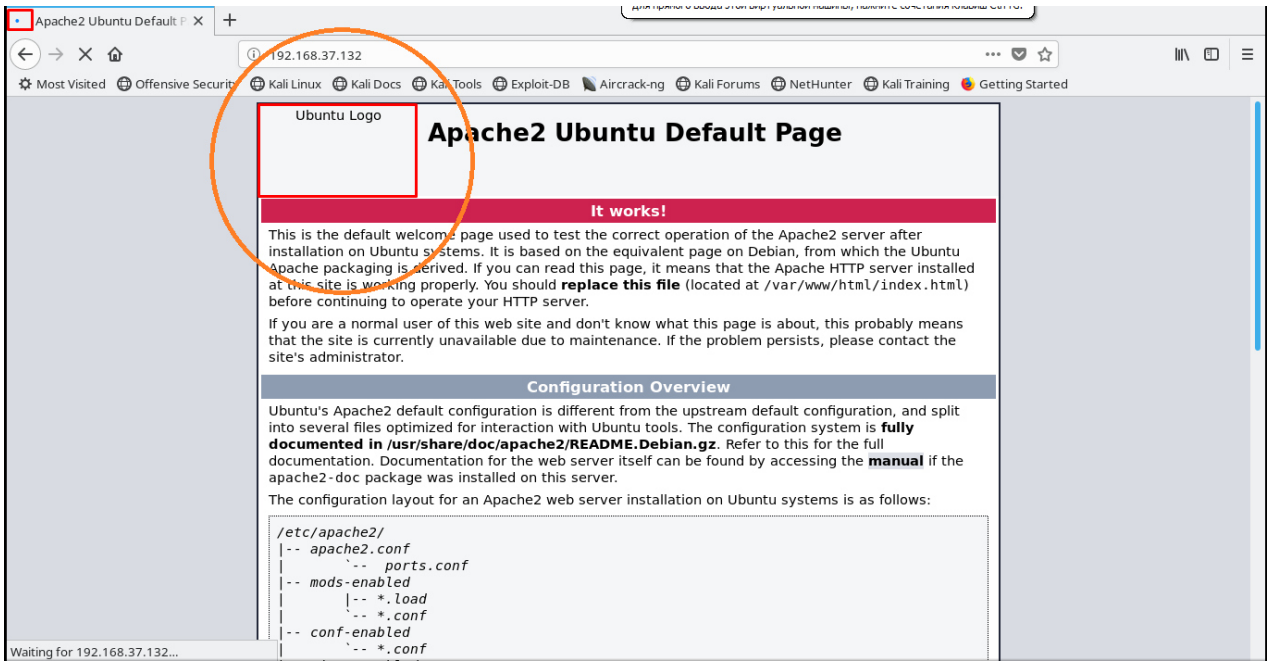

Рисунок 2.30 – Сайт Apache2

После установки защиты на сервер Apache2, сервер стал адекватно откликаться на запросы. Сайт сразу подгружается и подгружает изображение Ubuntu (рисунок 2.31).

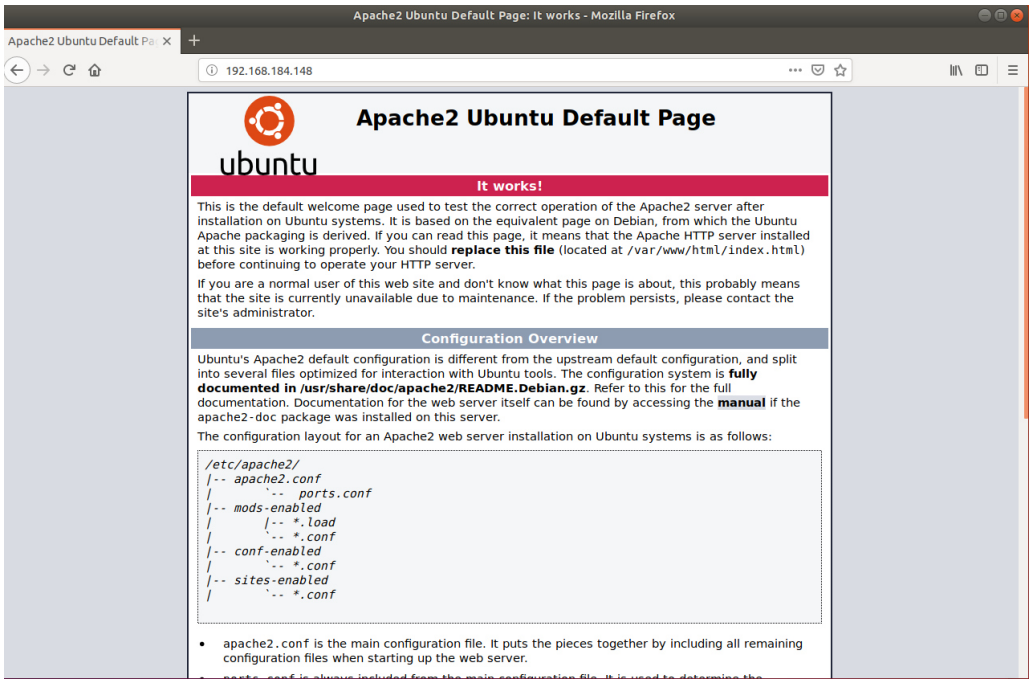

Рисунок 2.31 – Сайт Apache2

Выполнив атаку на сервер FTP, видно, что он уязвим внешним атакам. Из рисунка 2.32 видно, что сервер уязвим к bruteforce.

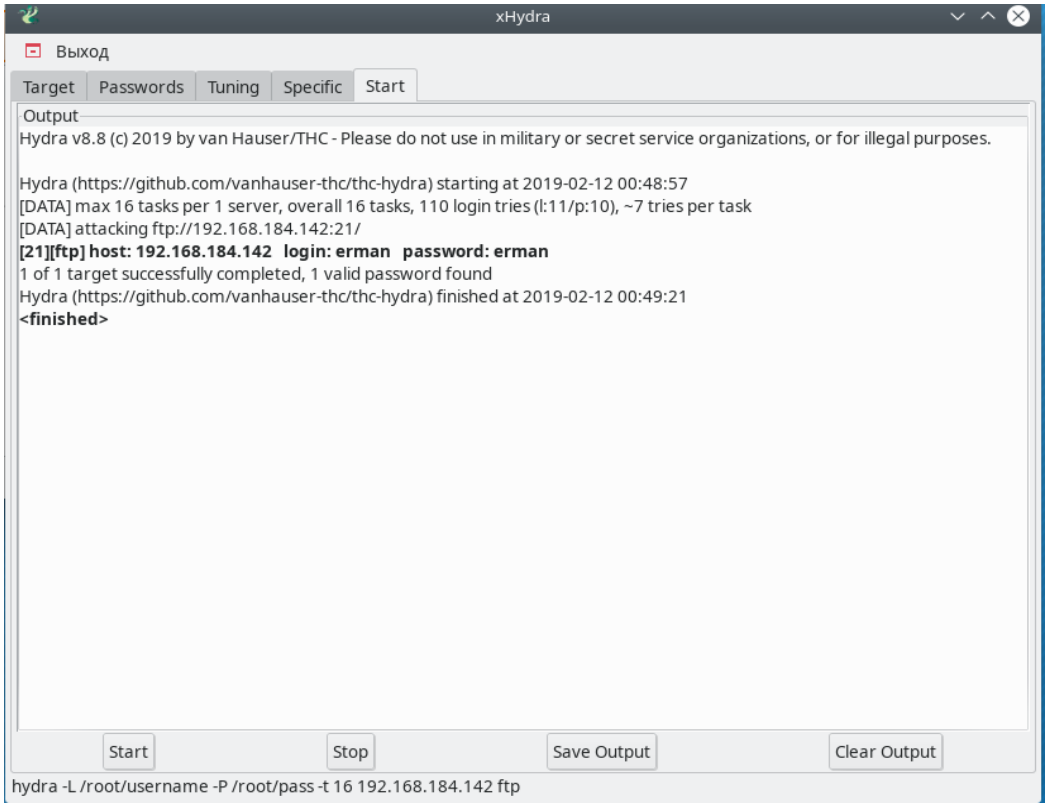

Рисунок 2.32 – Вкладка Start

После установки защиты на сервер FTP, сервер перестал быть уязвимым к bruteforce, это видно на рисунках 2.33 и 2.34.

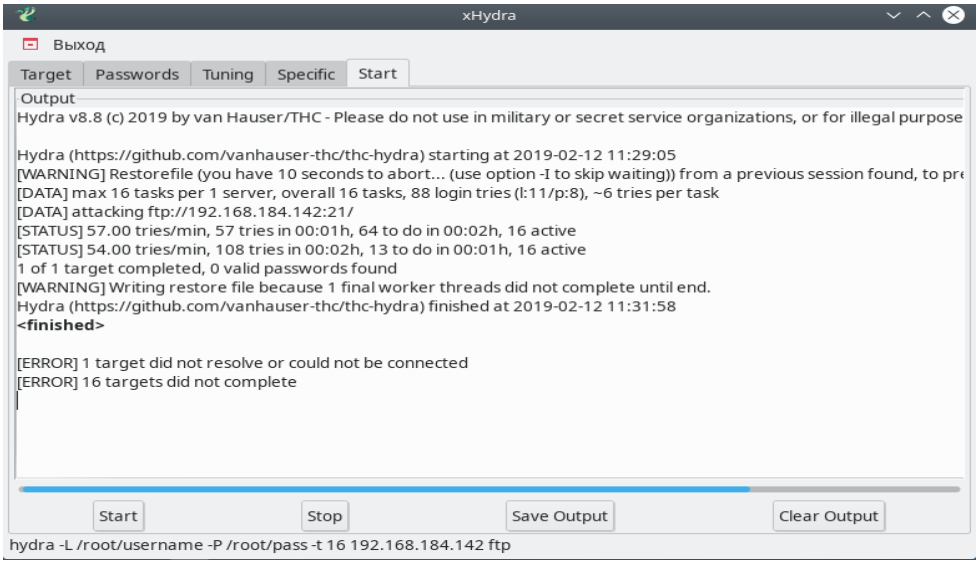

Рисунок 2.33 – Вкладка Start

Защита сразу же заблокировала адрес атакующего (рисунок 2.34).

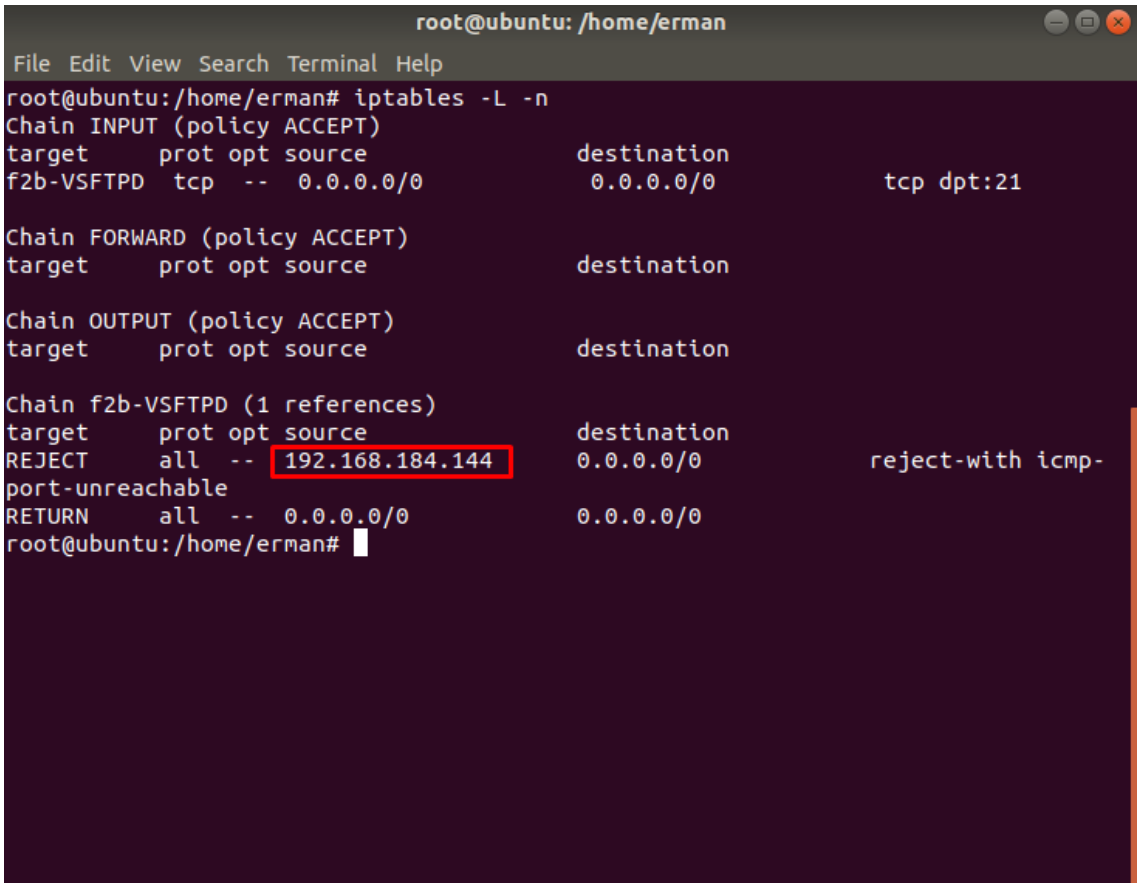

Рисунок 2.34 – Заблокированный IP-адрес

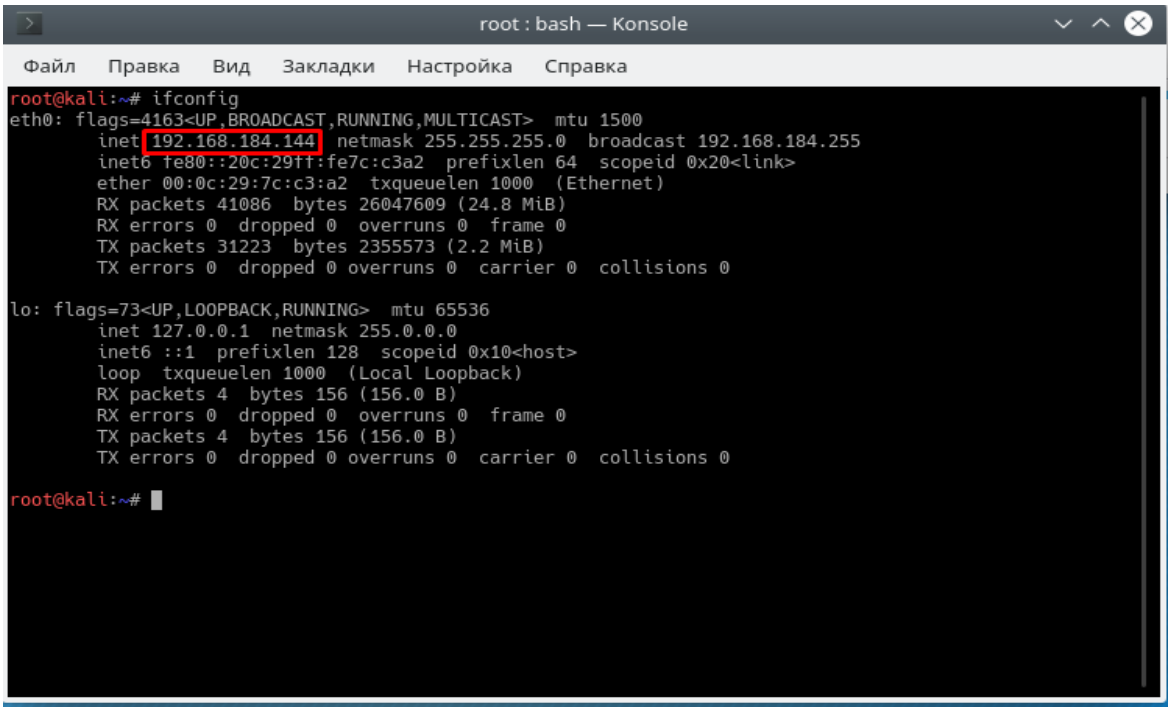

Рисунок 2.35 – IP-адрес атакующего

Вывод

В данной главе я рассмотрел атаки на сервера и проанализировал их уязвимости.

Для защиты сервер Apache я использовал Nginx в качестве обратного прокси к Apache.

Apache и Nginx — два самых распространённых веб-сервера в мире. Оба они способны обслуживать веб-сайты под большими нагрузками. Каждый веб-сервер имеет свои преимущества и недостатки. Причины популярности каждого из серверов хорошо известны: мощность Apache и скорость Nginx.

Оба сервера имеют недостатки: Apache имеет ограничения памяти сервера, в то время как Nginx, эффективный для статических файлов, нуждается в помощи php-fhm или аналогичных модулей для динамического контента.

Можно объединить два веб-сервера для большей эффективности, используя Nginx, как статический фронтенд и Apache — как Backend.

Для защиты сервера FTP я выбрал fail2ban. Описывая fail2ban в двух словах, можно сказать, что он позволяет на основе анализа логов блокировать тех, кто злоупотребляет доступностью сервера по сети. Например, защитить почтовые ящики от взлома путем перебора паролей или многократного запроса какого-либо ресурса.

Проанализировав уязвимости серверов, я защитил сервера на программном уровне, но сервера уязвимы на аппаратном уровне т.е к ним можно подключиться через USB-порт тем самым злоумышленники могут нанести необратимый ущерб серверу, чтобы этого избежать нужно заблокировать USB-порты.

Блокировки USB-портов необходимы для защиты ноутбуков, персональных компьютеров и рабочих станций корпоративной сети с целью предотвращения утечки информации, порчи и кражи личных данных, других вредоносных действий злоумышленников.

3 Выбор средств и разработка мер обеспечения защиты сервера

3.1 Анализ предметной области при защите USB-портов

Интерфейс USB стал одним из самых распространённых сегодня, поскольку обеспечивает удобную и быструю передачу данных. Работой этого интерфейса управляет USB-хост, который обнаруживает подключение и отключение USB устройств, управляет передачей данных, обеспечивает питанием подключённые устройства. Однако производители не защищают USB-устройства от перепрошивки, а USB-хосты не проверяют их на подлинность. Класс хакерских атак, основанный на уязвимости USB-устройств, получил название BadUSB. Проведя реверс-инжиниринг конкретного устройства, можно создать и записать в него вредоносный код. Использование USB-устройства с модифицированной прошивкой очень опасно, т.к. эксплойт запускается в процессе инициализации устройства, а существующие антивирусные решения пока не могут сканировать служебную область памяти. Прошить микроконтроллер USB-устройства в большинстве случаев можно прямо с компьютера через USB-разъём. Записанный в прошивку устройства вредоносный код может, например, имитируя клавиатуру, произвести необходимые действия за пользователя на заражаемом компьютере. Или, имитируя сетевое устройство, изменить сетевые настройки таким образом, что пользователь будет просматривать интернет-сайты через подконтрольные злоумышленнику промежуточные серверы. Кроме того, имитируя USB-флеш, вредоносный код может загрузить и запустить на компьютере с включенным автозапуском вирусную программу. Такой вирус может скопировать себя и на другие USB-устройства, подключённые в данный момент к компьютеру. Скомпрометированная система также может распространять вирус на остальные устройства, которые пользователь будет подключать к системе. Поскольку хосты не проверяют USB-устройства на подлинность, то это может повлечь за собой неконтролируемый рост заражённых аппаратных устройств, которые в настоящее время нечем проверять.

Возможность блокировки USB-портов является необходимой мерой для защиты ноутбуков, персональных компьютеров и рабочих станций корпоративной сети с целью предотвращения утечки информации, порчи и кражи личных данных, других вредоносных действий злоумышленников.

3.2 Проектирование ПО защиты USB-портов

Программа для блокировки USB-порта может как полностью блокировать USB-порты, так и выполнять лишь определенные действия с целью защиты USB-порта (блокировка на запись / чтение / удаление данных с USB-носителя). Кроме того, можно предотвратить автоматический запуск подключаемых съёмных USB носителей, выбрав параметр «Запретить автозапуск для всех устройств».

Функции защиты USB-портов могут выполняться специальными программами для блокировки USB портов. Блокировка USB-портов является одной из функций ПО и может быть использована для предотвращения подключений неавторизованных USB-носителей, копирования информации.

Прежде чем я начал создавать программу, я должен составить функциональную схему (блок-схему). На рисунке 7.1 представлена блок-схема режимов работы моей программы.

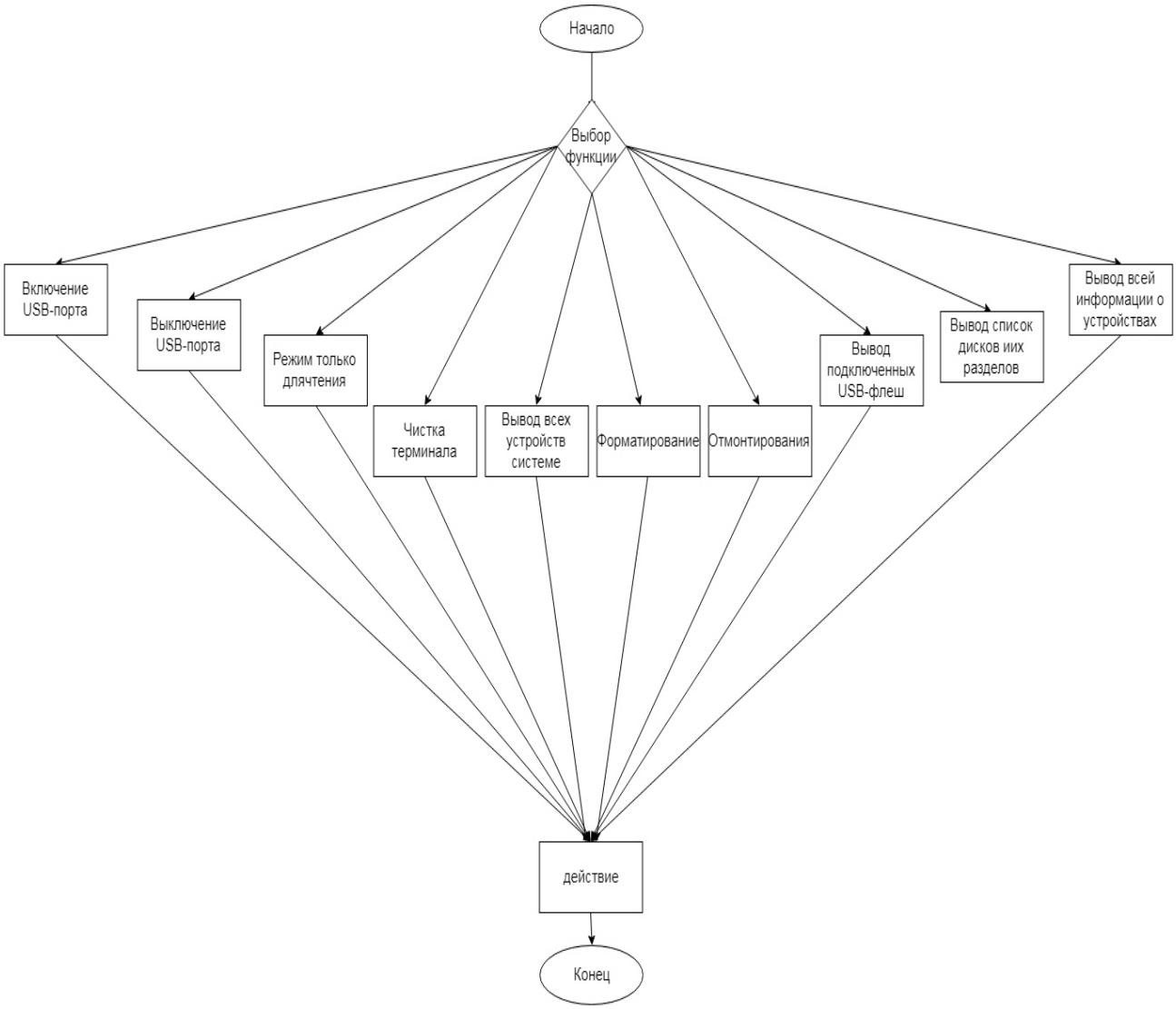

Рисунок 3.1 – Функциональная схема режимов работы

Своему приложению, которое я создаю в рамках дипломного проекта я дал название «kinza». Я выбрал среду программирования Python, на которой будет создано приложение. Выбрал среду программирования Python, потому что Python — это один из самых популярных языков программирования для Linux. На нем написано множество различных инструментов и библиотек. Кроме того, Python популярен среди разработчиков, потому что на нем очень просто и быстро программировать.

Шаги для создания программы:

1) запускаю терминал (рисунок 3.2);
2) создаю файл с расширением (py) (рисунок 3.2);
3) даю полный доступ к файлу (рисунок 3.3);
4) открываю его и пишу программный код (рисунки 3.4 и 3.5);
5) запускаю терминал (рисунок 3.1);
6) ввожу команду для запуска программы (рисунок 3.1);
7) работаю с программой (рисунки 3.1-3.24);
8) завершение работы с программой.

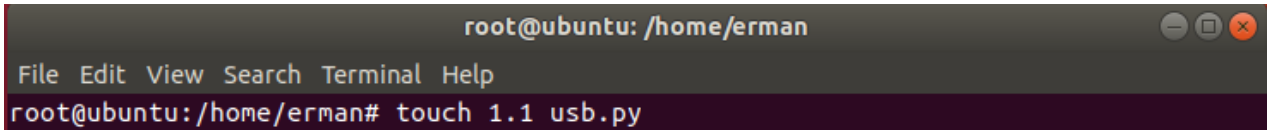

Рисунок 3.2 – Создание файла

Даю полный доступ к файлу (рисунок 3.3).

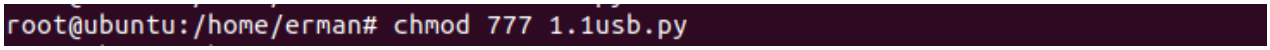

Рисунок 3.3 – Права на файл

Открываю файл с расширение (py) (рисунок 3.4).

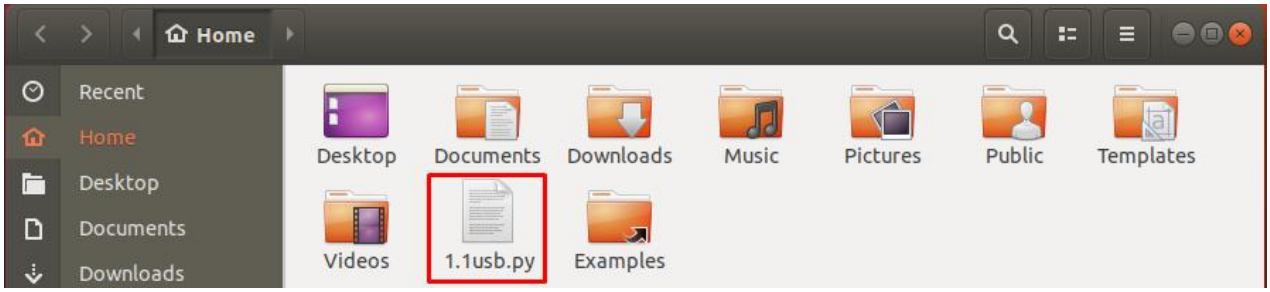

Рисунок 3.4 – Файл с расширением (py)

Пишу код для программы (рисунок 3.5).

Рисунок 3.5 – Код программы

Убедиться в работоспособности программы можно с помощью листинга программы, приложенного к дипломному проекту (приложение А).

3.3 Демонстрация и тестирование защиты ПО при защите USB-портов

Запускаю программу с помощью команды python3 usb.1.1.py (рисунок 3.6).

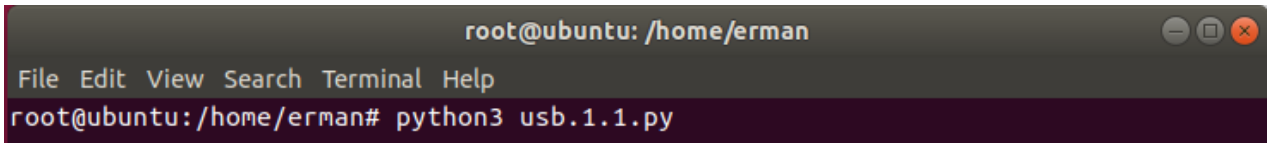

Рисунок 3.6 – Запуск ПО

Запущенная программа с интерфейсом выглядит как показано на рисунке 3.7.

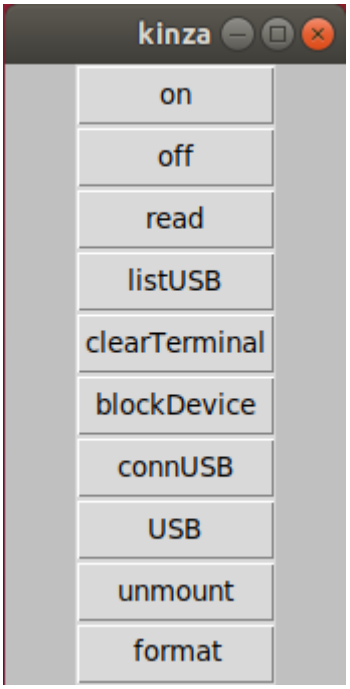

Рисунок 3.7 – Интерфейс программы «kinza»

Открываю домашний каталог и вижу смонтированную флеш (рисунок 3.8).

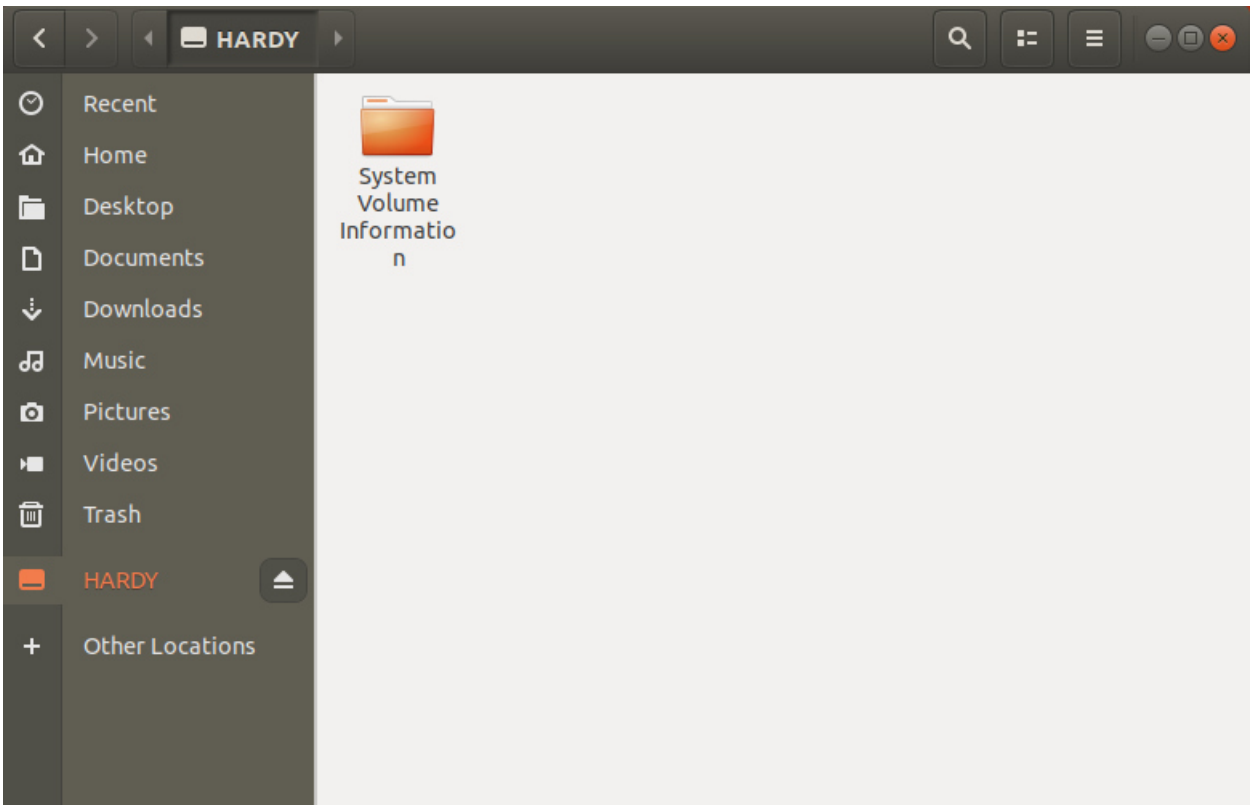

Рисунок 3.8 – Не заблокированная USB-флеш

Нажимаю на кнопку «off» для того, чтобы отключить USB-порт (рисунок 3.9).

Рисунок 3.9 – Кнопка блокирования USB-флеш

Ввожу пароль для выполнения данной функции (рисунок 3.10).

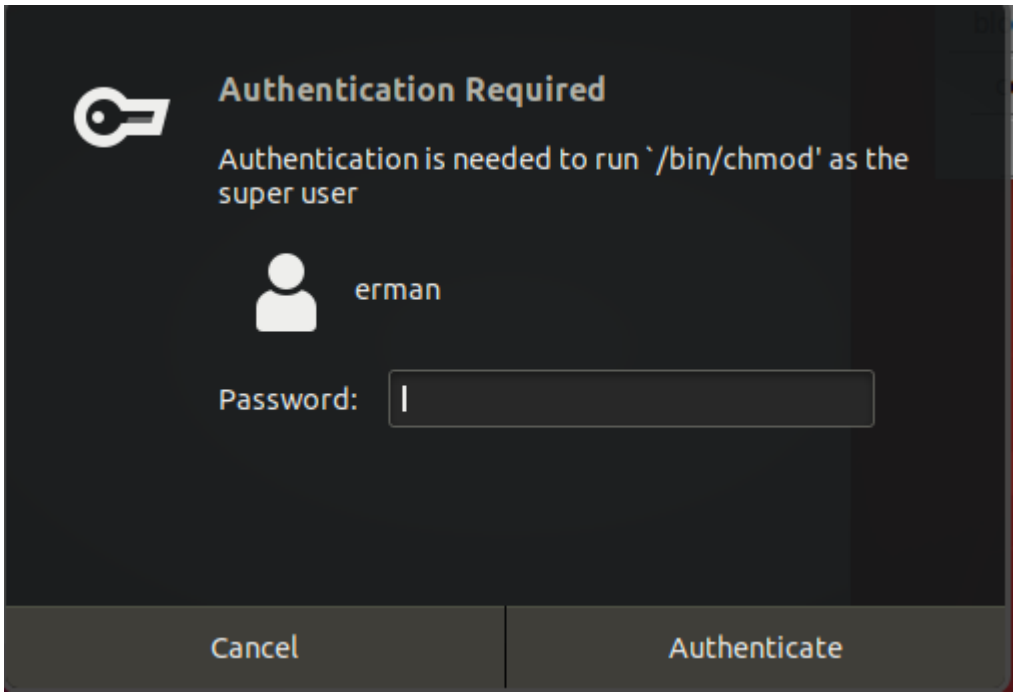

Рисунок 3.10 – Авторизация на выполнения функции блокирования

Пытаюсь открыть смонтированную флеш (рисунок 3.11).

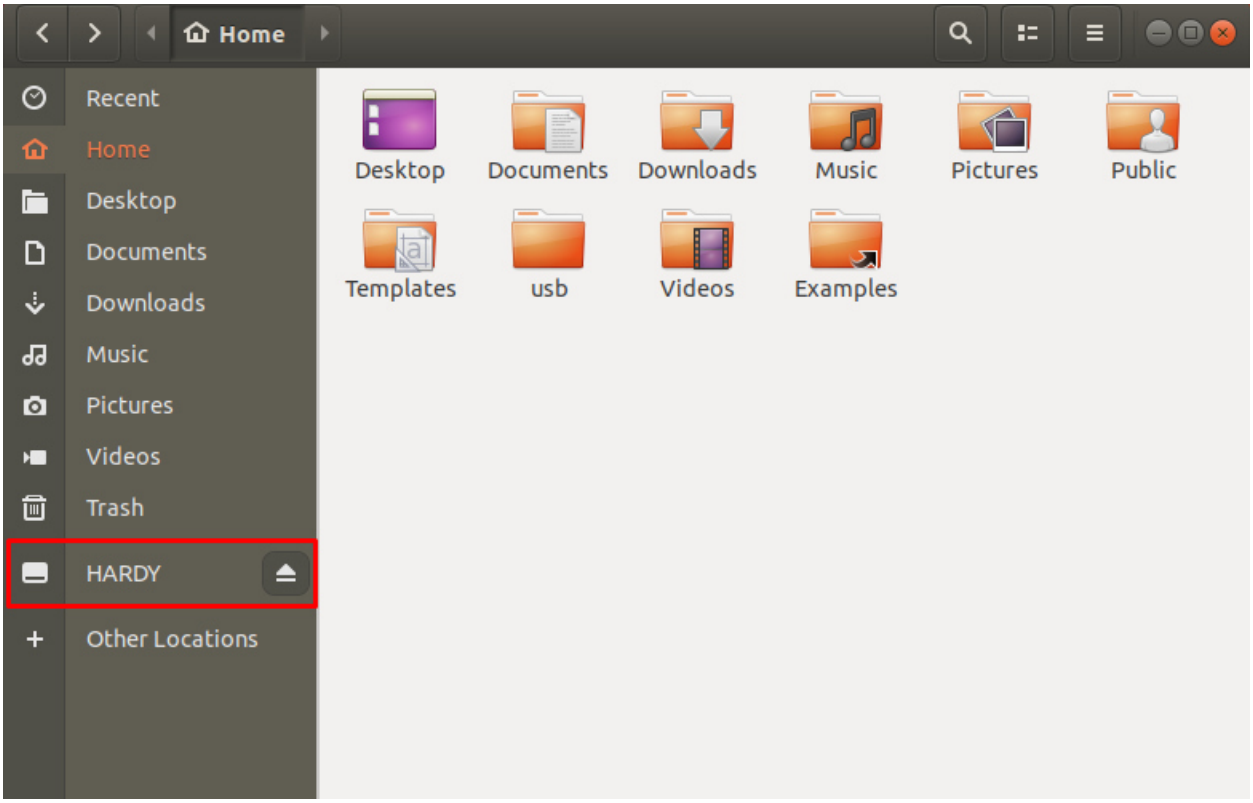

Рисунок – 3.11 USB-флеш

После нажатия на ссылку смонтированной USB-флеш выходит сообщение, что USB-флеш заблокирован (рисунок 8.7).

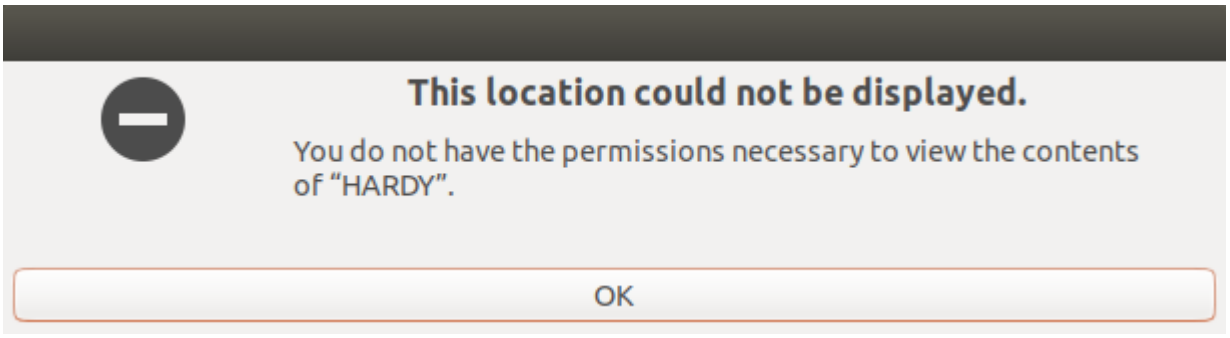

Рисунок – 3.12 Сообщение о заблокированном USB-порта

Нажимаю на кнопку «on» для того, чтобы включить USB-порт (рисунок 3.14).

Рисунок – 3.14 Кнопка включения USB-порта

Открываю смонтированную флеш и она снова работает (рисунок 3.15).

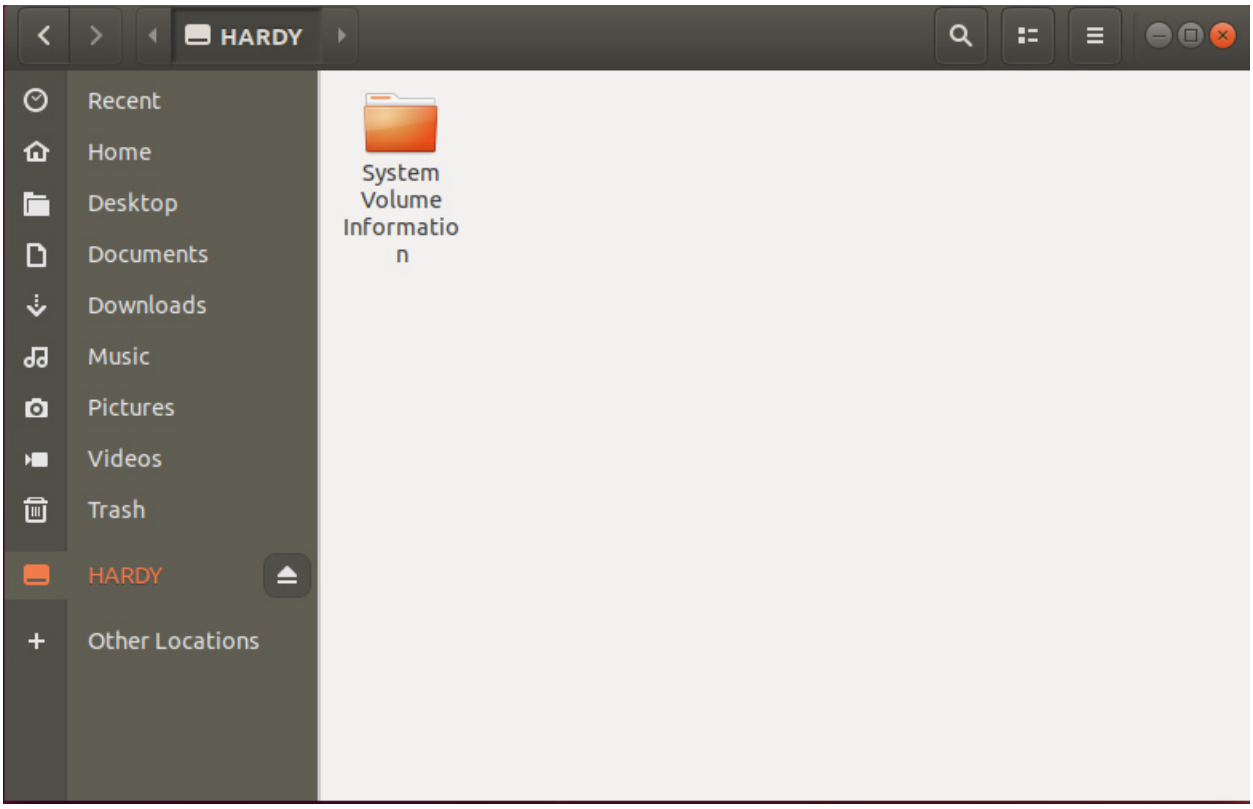

Рисунок 3.15 –USB-флеш

Нажимаю на кнопку «read», чтобы сделать USB-флеш только для чтения (рисунок 3.16).

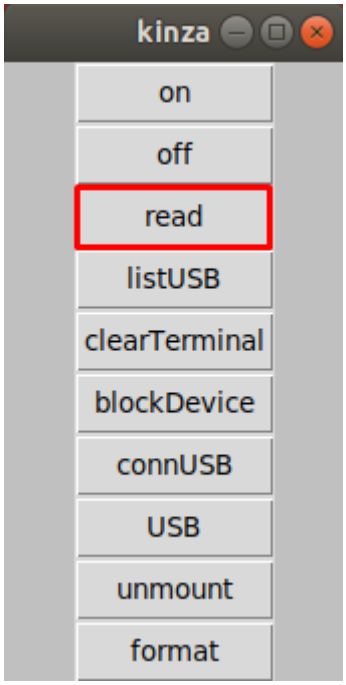

Рисунок 3.16 – Кнопка только для чтения USB-флеш

Нажимаю на кнопку «listUSB», чтобы сделать поиск подключенных USB-устройств (рисунок 3.17).

Рисунок 3.17 - Поиск подключенного USB-устройства

Окно терминала, где выведены все подключенные USB-устройства (рисунок 3.18).

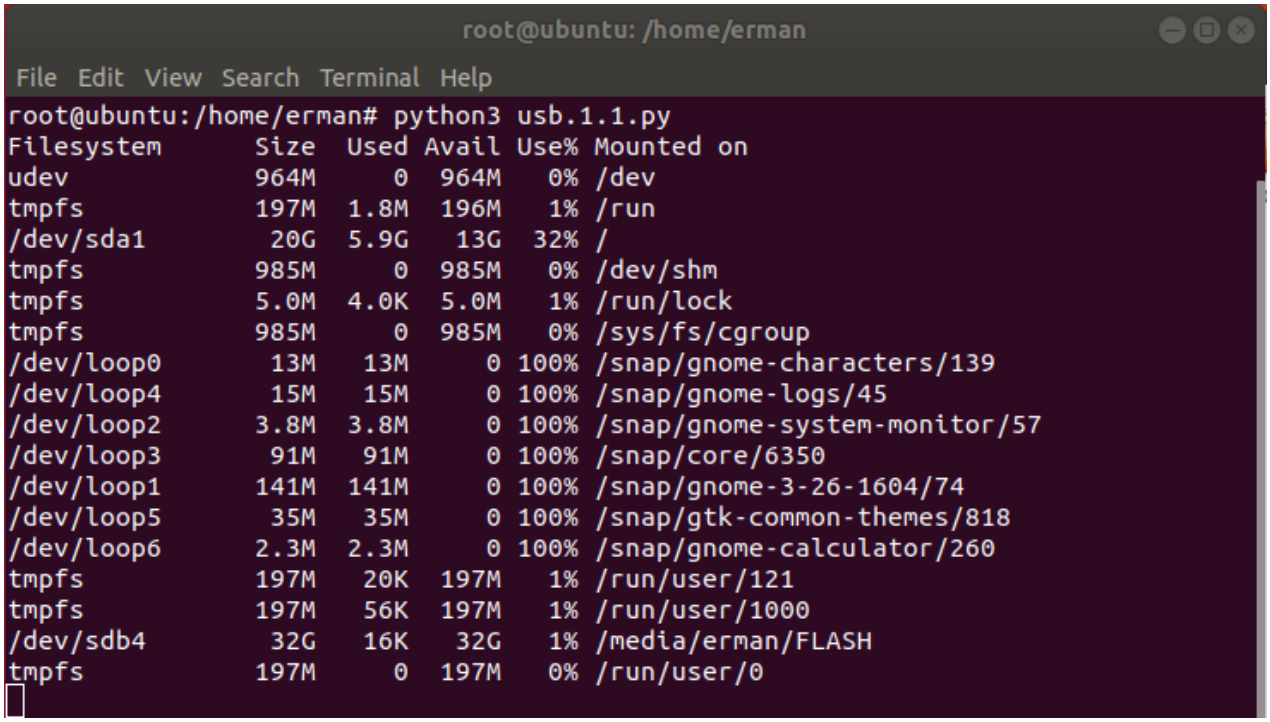

Рисунок 3.18 - Поиск подключенного USB-устройства

Данная кнопка служит для очистки терминала (рисунок 3.19).

Рисунок 3.19 – Очистка терминала

Нажимая на данную кнопку, выводятся все блочные устройства, подключенные к системе (рисунок 3.20).

Рисунок 3.20 – Все блочные устройства, подключенные к системе

Окно терминала, где выведен список подключенных блочных устройства (рисунок 3.21).

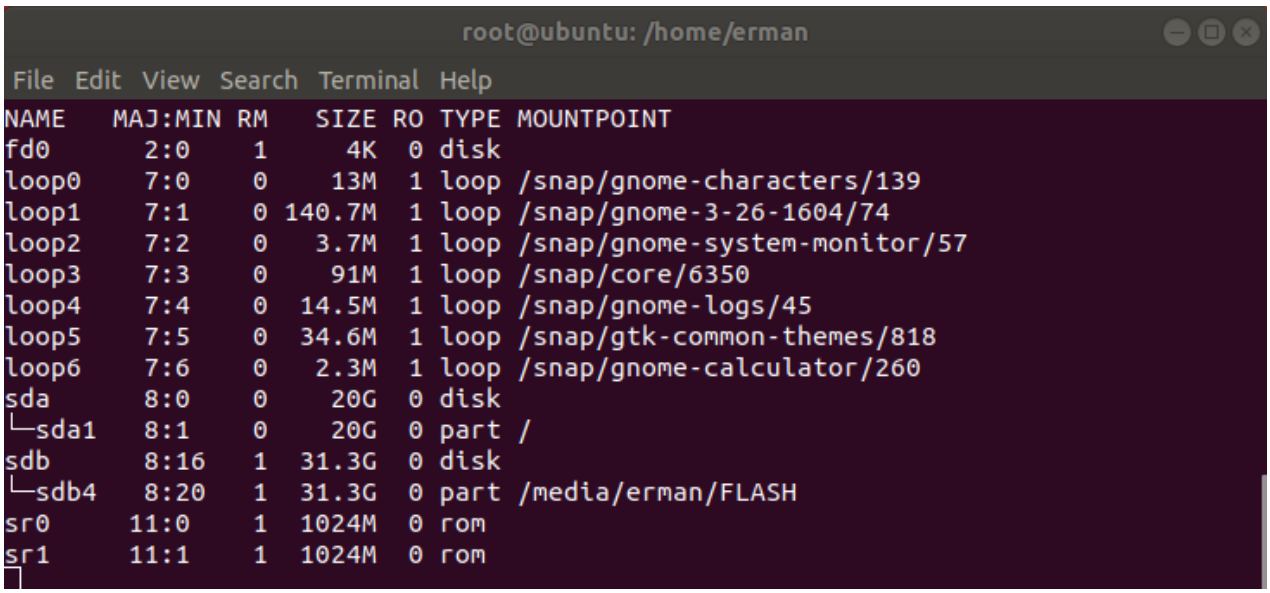

Рисунок 3.21 – Список подключенных блочных устройств

Данная кнопка выводит все подключенное оборудование (рисунок 3.22).

Рисунок 3.22 – Вывод всего подключенного оборудования

Окно терминала, где выведен список всего подключенного оборудования (рисунок 3.23).

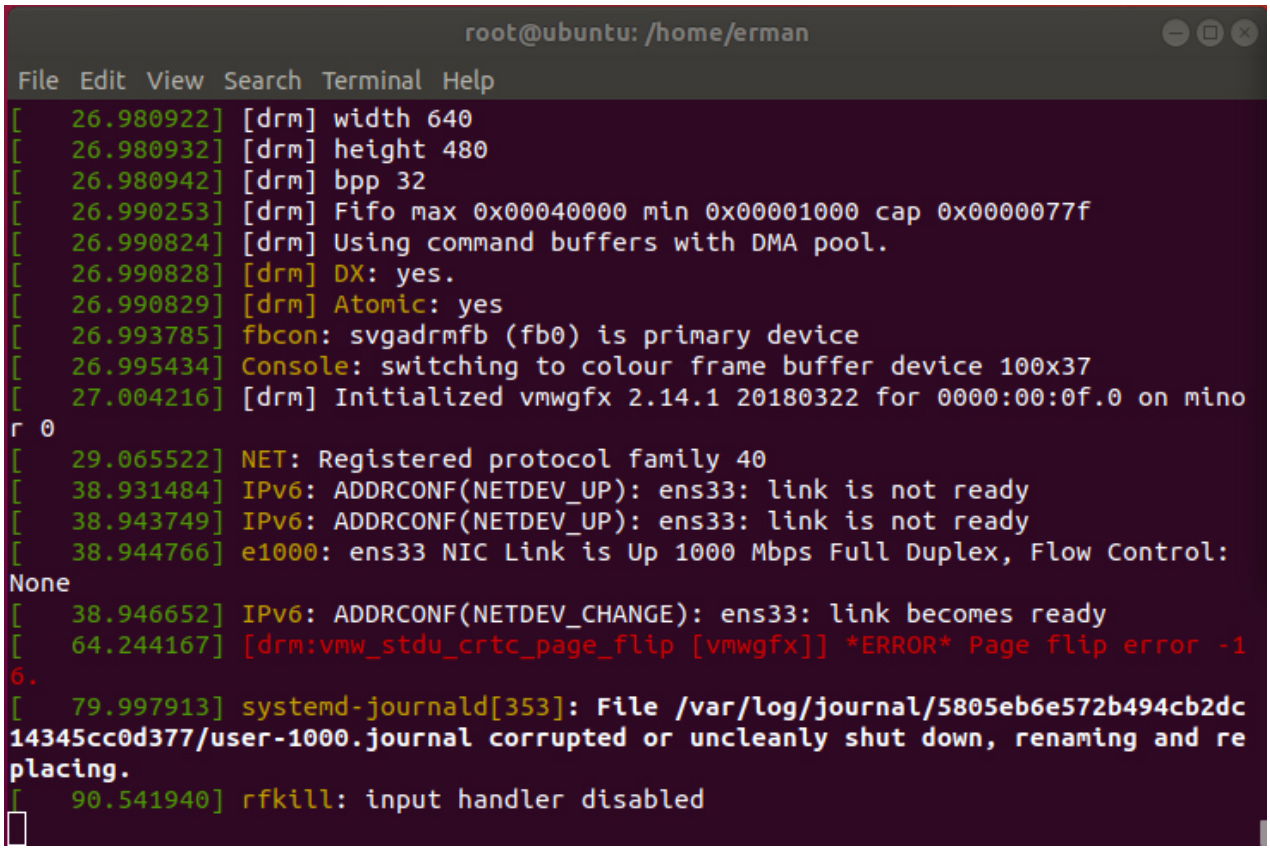

Рисунок 3.23 – Вывод всего подключенного оборудования

Данная кнопка выводит подключенные USB-устройств (рисунок 3.24).

Рисунок 3.24 – Список подключенных USB

Окно терминала, где выведен список подключенных USB-устройств (рисунок 3.25).

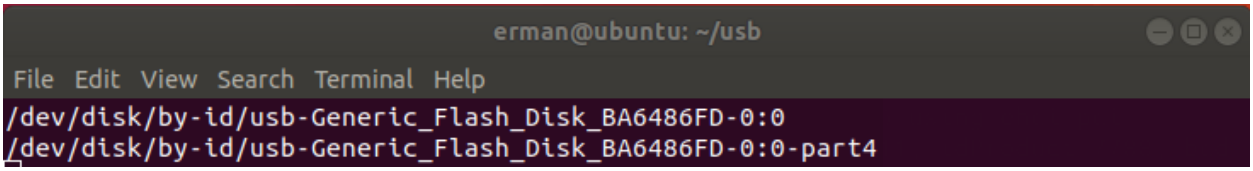

Рисунок 3.25 – Список подключенных USB

Кнопка для отмонтирования устройства на рисунке 3.26.

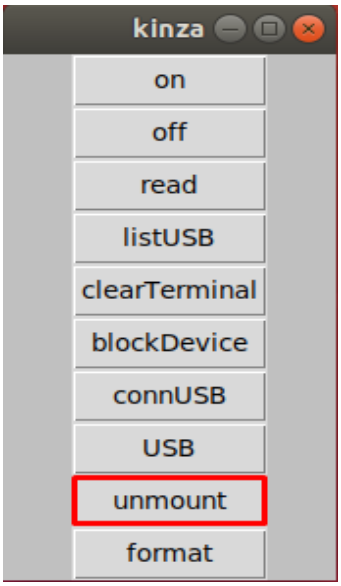

Рисунок 3.26 – Кнопка отмонтирования устройства

Окно терминала, где выведена информация о том, что устройство отмонтировано (рисунок 3.27).

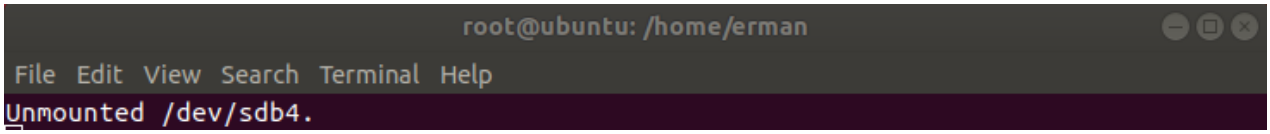

Рисунок 3.27 – Отмантирование устройства

Данная кнопка форматирует USB-флеш (рисунок 3.28).

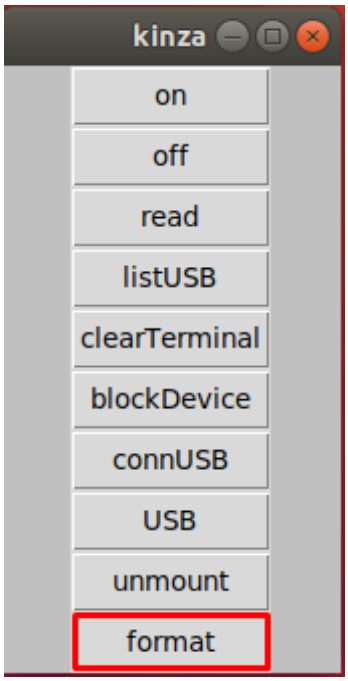

Рисунок 3.28 – Форматирования флешки

Окно терминала, где выведена информация о том, что USB-флеш отформатировалась (рисунок 3.29).

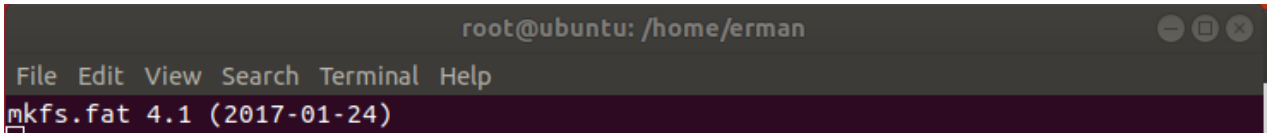

Рисунок 3.29 – Форматирование USB-флеш

Вывод

В данной главе я провел анализ предметной области, выделив предпосылки. Пришел к выводу, что блокировка USB-портов является необходимой мерой для защиты ноутбуков, персональных данных компьютеров и рабочих станций корпоративной сети с целью предотвращения утечки информации, порчи и кражи личных данных, других вредоносных действий злоумышленников. И поэтому в проекте я реализую практическую защиту от данной уязвимости, детально описав процесс создания и тестирования программного обеспечения защиты USB-портов.

Процесс реализации был разбит на несколько этапов. Я составил функциональную схему работы программы. Затем я приступил к созданию приложения для защиты USB-портов на языке Python под названием «kinza», которое выполняет защитные функции предотвращения утечки информации по USB-портам в операционной системе Linux Ubuntu:

1) включение USB-портов;
2) выключение USB-портов;
3) включение режима «только для чтения»;
4) поиск подключенного USB-устройства;
5) очистка терминала;
6) вывод всех блочных устройств, подключенных к системе;
7) вывод всего подключенного оборудования;
8) список подключенного USB-флеш;
9) отмонтирования USB-флеш;
10) форматирование USB-флеш.

Мною разработанное приложение «kinza» для применения на корпоративном сервере являет собой реализации на актуальную и важную тематику.

Программа для блокировки USB-порта может полностью выполнять свои функции, отлажена и работоспособна. Блокировка USB-портов – одна из функций ПО и может быть использована для предотвращения подключений неавторизованных USB носителей, копирования информации. Протестировав функционал приложения для защиты USB-портов можно сказать, что данное ПО успешно справляется с поставленной задачей.

4 Экономическая часть

На сегодняшний день существует довольно много программного обеспечения, используемого для целей и методов защиты USB-портов.

4.1 Технико-экономическое обоснование

В дипломном проекте рассматривается разработка системы управления флеш накопителем через программное обеспечение.
Модель, разрабатываемая в рамках дипломной работы, предназначен для операционной системы Linux.

4.2 Расчет трудоемкости разработки модели

Для определения сложности разработки модели приводится список всех основных этапов и видов работ, которые необходимо выполнить. Форма разделения работ на этапы с указанием сложности их выполнения приведена в таблице 4.1.

Таблица 4.1 - Распределение трудоемкости и времени на определенный этап

Этапы разработки ПО	Вид работы	Трудоемкость, чел. час.
Этап 1	Постановка задач	10
Этап 2	Знакомство с материалами проекта	15
Этап 3	Поиск и изучение подобных программ	25
Этап 4	Изучение форматов usb-флеш-накопителей	25
Этап 5	Работа с usb-флеш-накопителями	30
Этап 6	Реализация кода	56
Этап 7	Подготовка интерфейса приложения	10
Этап 8	Анализ приложения	20
Этап 9	Внедрение	15
Итого: трудоемкость выполнения дипломного проекта		206

4.3 Расчет затрат на разработку ПО

Общая сумма затрат на материальные ресурсы (Зм) определяется по формуле:

$$ЗМ = \sum Pi \times Цi, \quad (4.1)$$

где Pi - расход i-го вида материального ресурса, натуральных единиц;
Цi - цена за единицу i-го вида материального ресурса, тг;
i - вид материального ресурса;

n - количество видов материальных ресурсов.

Расчет затрат на материальные ресурсы производится по форме, приведенной в таблице 4.2.

Таблица 4.2 - Затраты на материальные ресурсы

Наименование материала	Компания производитель	Количество	Цена	Общая сумма
Тетрадь	Gvendin	1	200	200
Ручка	Apatix	1	70	70
Бумага	Снежок	1	800	800
Флеш	Trancendb3	1	4200	4200
Модем	Tp-Link3000	1	12700	12700
Итого				17970

Таблица 4.3 - Затраты на операционные системы и программное обеспечение

Наименование материала	Компания производитель	Количество	Цена	Общая сумма
Ноутбук	Lenovoz570	1	128000	128000
Принтер	Canon3010	1	25000	25000
Мышь	Motospeed	1	5000	5000
Операционные системы	Ubuntu	1	---	---
Программное обеспечение	LibreOffice	1	---	---
	Kaspersky Anti-Virus for Linux Server 8	1	10000	10000
	Python 3.6	1	---	---
Итого				168000

$$Зм = 17\ 970 + 168\ 000 = 185970\ (тг)$$

Для реализации программного обеспечения необходимы материалы на сумму 185 970 тенге.

4.4 Расчет затрат на электроэнергию

Поскольку в процессе производства используется электрооборудование, необходимо рассчитать стоимость электроэнергии. Затраты на электроэнергию для производственных нужд включают стоимость электроэнергии на оборудование и дополнительные нужды [10].

Согласно Таблице 4.1 разработка программного обеспечения требует около 206 часов. Теперь необходимо рассчитать стоимость электроэнергии,

которая будет потрачена в течение 206 часов. Для принтера расчет будет производиться в течение 24 часов, поскольку нет необходимости постоянно использовать принтер.

$$Э = З_{эл.эн.обор} + З_{доп.нуж}, \quad (4.2)$$

где $З_{эл.эн.обор}$ – затраты на электроэнергию оборудования;
$З_{доп.нуж.}$ – затраты электроэнергии на дополнительные нужды.

Расходы электроэнергии на оборудование рассчитывается по формуле:

$$З_{эл.эн.обор} = \sum W \times К_{исп} \times S \times T \quad (4.3)$$

где W – потребляемая мощность, Вт;
$К_{исп}$ – коэффициент использования ($К_{исп}$ = 0,9);
T – время работы;
S – тариф без НДС (1кВт/ч = 25 тг).

Сводные результаты расчета затрат на электроэнергию представлены в таблице 4.4.

Таблица 4.4 - Затраты на электроэнергию

Наименование приборов	Паспортная мощность, кВт	Коэф-т мощности	Время работы оборудования, Ч	Цена ЭЭ тг/кВт;	Сумма, тг.
Ноутбук	0,06	0,9	206	25	278,1
Принтер	0,4	0,9	24	25	216
Модем	0,04	0,9	206	25	185,4
Освещение	0,3	0,7	206	25	1081,5
Итого					1761

$$З_{эл.эн.обор.(ноутбук)} = 0{,}06 * 0{,}9 * 206 * 25 = 278{,}1 \text{ (тенге)}$$
$$З_{эл.эн.обор.(принтер)} = 0{,}4 * 0{,}9 * 24 * 25 = 216 \text{ (тенге)}$$
$$З_{эл.эн.обор.(модем)} = 0{,}04 * 0{,}9 * 206 * 25 = 185{,}4 \text{ (тенге)}$$
$$З_{эл.эн.обор.(освещение)} = 0{,}3 * 0{,}9 * 206 * 25 = 1081{,}5 \text{ (тенге)}$$
$$З_{эл.эн.обор.} = 1\,761 \text{ (тенге)}$$

Затраты на дополнительные потребности берутся по укрупненному показателю в размере 5% от затрат на оборудование:

$$З_{доп.нуж} = 5\% \times З_{эл.эн.обор}, \quad (4.4)$$

Затраты на дополнительные потребности рассчитан по формуле (4.4):

$$З_{доп.нуж} = 0{,}05 \times 1\ 761 = 88{,}05 \text{ (тенге)}.$$

Таким образом суммарные затраты на электроэнергию составляют:

$$Э = 88{,}05 + 1\ 761 = 1849{,}05 \text{ (тенге)}.$$

4.5 Расчет затрат на оплату труда

Для разработки программного обеспечения, как указывалось ранее, необходимо два работника:

- руководитель проекта – управление рабочим временем, корректировка рабочих процессов, координация, изучение предметной области;
- разработчик – разработка ПО, тестирование и сопровождение.

Сумму расходов на оплату труда можно рассчитать по следующей формуле:

$$З_{тр} = \sum ЧС_i * T_i \quad (4.5)$$

где $ЧС_i$ - часовая ставка i-го работника, тг;

T_i - трудоемкость разработки модели, чел.×ч; i - категория работника;

n - количество работников, занятых разработкой ПП.

Во время реализации проекта рабочее время участников не равномерно, поэтому имеет смысл установить часовую ставку каждого работника и общий объем заработной платы.

Часовую ставку сотрудника можно рассчитать по следующей формуле:

$$ЧС_i = \frac{ЗП_i}{ФРВ_i} \quad (4.6)$$

где $ЗП_i$ - месячная заработная плата i-го работника, тг;

$ФРВ_i$- месячный фонд рабочего времени i-го работника, час.

Месячная заработная плата руководителя равняется 180 000 тенге и месячная заработная плата разработчика равняется 150 000 тенге. Рассчитаем часовую ставку каждого работника согласно формуле (4.6):

$$ЧС_{руководитель} = \frac{180\ 000}{22 * 8} = 1\ 022{,}72 \text{ тг/ч}$$

$$ЧС_{разработчик} = \frac{150\ 000}{22 * 8} = 852{,}3 \text{ тг/ч}$$

Часовая ставка руководителя составляет 1 022,72 (тг/ч), трудоемкость разработки равняется 100 часам. Часовая ставка разработчика составляет

852,3 (тг/ч), трудоемкость разработки равняется 206 часам. Согласно формуле (4.5) можно рассчитать сумму расходов на заработную плату работников:

$$З_{тр} = 1022{,}72 * 100 + 852{,}3 * 206 = 102272 + 175573{,}8 = 277845{,}8$$

Расчеты затрат по оплате труда показаны в таблице (4.5).

Таблица 4.5 – Расчет заработной платы

Категория работника	Квалификация	Трудоемкость разработки ПП, час.	Часовая ставка, тг/ч	Сумма, тг.
Руководитель	Проектный руководитель	100	1022,72	102272
Разработчик	Программист	206	852,3	175573,8
Итого:				277845,8

4.6 Расчет затрат по социальному налогу

Социальный налог – согласно Налоговому кодексу Республики Казахстан он составляет 9,5 % от ФОТ (фонда оплаты труда). Следует отметить, что пенсионные отчисления не облагаются социальным налогом. Социальный налог можно рассчитать по следующей формуле:

$$С_н = (ФОТ - ПО) * 0{,}095 \quad (4.7)$$

где ПО - отчисления в пенсионный фонд, они составляют 10% от ФОТ.

$$ПО = 277845{,}8 * 0{,}1 = 27\,784{,}58 \text{ тенге}$$
$$С_н = (277845{,}8 - 27\,784{,}58) * 0{,}095 = 23\,755{,}82 \text{ тенге}$$

Результаты расчетов представлены в таблице (4.6):

Таблица 4.6 – Начисление социального налога

Категория работника	Количество человек	Заработная плата, тг	Пенсионные отчисления,тг	Социальный налог, тг
Руководитель	1	102272	10 273	8714,1
Разработчик	1	175573,8	19 177	16 396, 1
Итого:				23 755,82

4.7 Амортизация основных фондов и прочие затраты

Нормы амортизации ОФ необходимо определить в соответствии с налоговым кодексом РК. Амортизацию ОФ можно определить по следующей формуле:

$$A_{\text{г}} = \frac{\text{С}_{\text{об}} * \text{Н}_{\text{а}}}{100} \quad (4.8)$$

где, $\text{С}_{\text{об}}$ – стоимость оборудования;

$\text{Н}_{\text{а}}$ – норма амортизации (норма амортизация = 25);

Формула (4.8) позволяет рассчитать нужную сумму для амортизационных отчислений за год для ноутбука:

$$A_{\text{г}} = \frac{128\,000 * 25}{100} = 32\,000 \text{ тенге}$$

Теперь необходимо рассчитать норму амортизации за период разработки:

$$A_{\text{г(ноутбук)}} = \frac{32\,000 * 26}{365} = 2279{,}4 \text{ тенге}$$

$$A_{\text{г(принтер)}} = \frac{6250 * 26}{365} = 445{,}2 \text{ тенге}$$

$$A_{\text{г(модем)}} = \frac{2540 * 26}{365} = 180{,}9 \text{ тенге}$$

Подобным образом необходимо рассчитать норму амортизации для всего оборудования. Результаты расчетов приведены в таблице (4.7).

В часть «Прочие затраты» включаются расходы на арендную плату, включая коммунальные платежи, затраты на лицензирование и сертификацию, расходы на рекламу, канцелярские и прочие хозяйственные расходы.

Месячная оплата за интернет – 4990 тенге. Общая стоимость интернета за весь период определяется по формуле:

$$\text{Р}_{\text{и}} = \text{Т}_{\text{и}} \times \text{С}. \quad (4.10)$$

где $\text{Т}_{\text{и}}$– Время использования интернета, мес.

С – стоимость интернета за 1 месяц.

Общая стоимость интернета за весь период рассчитан по формуле (4.10):

$$\text{Р}_{\text{и}} = 3 \times 9\,200 = 27\,600 \text{ (тенге)}.$$

Прочие затраты = 27 600 (тенге).

Таблица 4.7 – Амортизация ОФ

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Сумма амортизации за год, тг	Сумма амортизации за время разработки, тг
Ноутбук	128000	25	32 000	2 279,4
Принтер	25000	25	6 250	445,2
Модем	12700	20	2 540	180,9
Итого:			40 790	2 905,5

Смета расходов на разработку ПО.

На основе всех представленных расчетов необходимо оформить смету расходов на разработку ПО согласно форме, которая приведена в таблице (4.8). На рисунке 4.1 продемонстрирована диаграмма рабочих расходов.

Таблица 4.8 – Смета затрат на разработку ПО

Статьи затрат	Сумма, тг
Затраты на оборудование	168 000
Затраты на программное обеспечение	0
Затраты на оплату труда	277845,8
Социальные налоги	23 755,82
Затраты на электроэнергию	1761
Амортизация основных фондов	2 905,5
Прочие расходы	27 600
Итого по смете:	502762,25

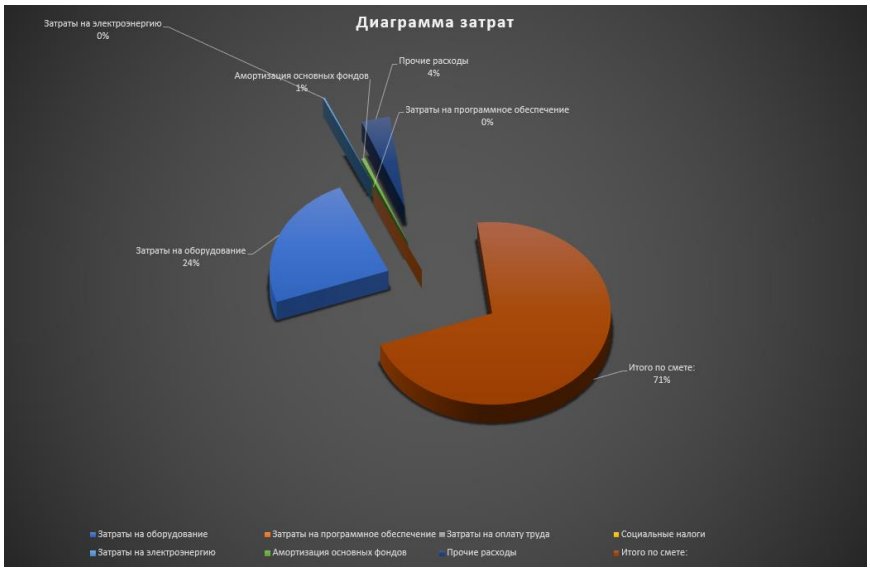

Рисунок 4.1 – Диаграмма затрат

4.8 Определение возможной (договорной) цены ПО

Стоимость программного обеспечения определяется на основе качества разработанного продукта, сроков его разработки и производительности продукта. Стоимость $Ц_Д$ для программного обеспечения можно рассчитать по следующей формуле:

$$Ц_Д = З_{нир}\left(1 + \frac{P}{100}\right), \quad (4.13)$$

где $З_{нир}$ – затраты на разработку программного обеспечения, тг;

Р – средний уровень рентабельности ПО, (%). Данный параметр принят равным 25%.

$$Ц_Д = 502\,762{,}25\left(1 + \frac{25}{100}\right) = 502762{,}25 + 502762{,}25 * 0{,}25$$
$$= 502762{,}25 + 125690{,}56 = 628452{,}81 \text{ тенге}$$

$$Ц_Д = 502\,762{,}25 + 125\,690{,}562 = 628452{,}81 \text{ тенге}$$

Далее необходимо определить стоимость реализации с учетом НДС, ставка НДС устанавливается законодательством РК. На 2019 года ставка НДС составляет 12%. Стоимость реализации учитывая НДС можно рассчитать по следующей формуле:

$$Ц_р = Ц_д + Ц_д * НДС, \quad (4.14)$$

$$Ц_р = 628452{,}812 + 628452{,}812 * 0{,}12 = 703\,867{,}15 \text{ тенге}$$

Цена реализации с учетом НДС равна 703 867,15 тенге.
Себестоимость – 502762,25 тенге.
Прибыль – 125690,56 тенге.

Вывод

В этой главе были рассмотрены экономическая и техническая реализация проекта, расчет всех необходимых затрат на приобретение оборудования, различные затраты на энергопотребление и другие вещи, пенсионные взносы, уплата налогов и амортизационные отчисления.

Этот проект был задуман как малый. Не требует огромных инвестиций, многочисленных офисов и многочисленных сотрудников. Судя по масштабам организации, огромной прибыли не ожидается, но она также не требует больших затрат. Учитывая все факторы и тот факт, что на Казахстанском рынке нет конкурентов в этой области. Этот проект вполне претендует на успех. Также подробно показан процесс сокрытия информации, какие видеофайлы я использовал.

5 Безопасность жизнедеятельности

5.1 Анализ условий труда обслуживающего персонала при эксплуатации технического оборудования

Главной целью работы является установка и защита ОС на базе Linux на которой будет работать сервер. Серверное помещение – это телекоммуникационное помещение, в котором размещаются распределительные устройства и большое количество активного телекоммуникационного оборудования.

В процессе трудовой деятельности на рабочем месте могут воздействовать опасные (вызывающие травмы) и вредные (вызывающие заболевания) производственные факторы:

- повышенная температура поверхностей оргтехники;
- повышенная или пониженная температура воздуха рабочей зоны;
- повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека;
- отсутствие или недостаток естественного света;
- недостаточная искусственная освещенность рабочей зоны.

Работа на персональном компьютере сопровождается постоянной и значительной интенсивностью функций зрительного анализатора. Одной из основных особенностей является другой принцип чтения информации, чем при обычном чтении. При обычном чтении текста на бумаге, расположенной горизонтально на столе, рабочий со склоненной головой читает вслух, падая легким потоком на текст. Работая на персональном компьютере, оператор читает текст вслух, почти не наклоняя головы, глаза смотрят прямо или почти прямо, текст (источником света является вещество на экране) формируется на другой стороне экрана, поэтому пользователь не читает вслух отраженный текст и смотрит прямо на источник света, который надолго заставляет глаза и орган зрения работать в целом напряженным, непривычным для него образом.

Нарушение работы органов зрения резко усиливается при работе более четырех часов в сутки. Всемирная организация здравоохранения (ВОЗ) представила концепцию «компьютерного зрительного синдрома» (KZS), при которой стандартные признаки жжения в глазах, окрашивания век и конъюнктивы в красный цвет, боли в глазнице и лбу, затуманивание, замедленная рефокусировка от близких объектов до отдаленных.

Физические вредные и опасные факторы рассматриваются: повышенный уровень электромагнитного, рентгеновского, ультрафиолетового и инфракрасного излучения; повышенный уровень статического электричества и запыленности воздуха рабочей зоны; увеличение содержания положительных утюгов и уменьшение содержания отрицательных утюгов в воздухе рабочей зоны.

Психофизиологические вредные и опасные факторы: интенсивность внешности и внимания; интеллектуальная, эмоциональная и длительная статическая нагрузка; однообразие работы; большое количество информации,

обрабатываемой в единицу времени; нерациональная организация рабочего места.

Типичные ощущения, которые операторы ПК испытывают к концу рабочего дня: напряжение глаз, головная боль, ноющие боли в мышцах шеи, рук и спины уменьшаются в концентрации.

Технический персонал состоит из двух сотрудников: главного технического директора и ИТ-специалиста с навыками информационной безопасности.

Работа сотрудников связана с компьютерами, соответственно с вредным дополнительным воздействием целой группы факторов, что существенно снижает производительность их труда.

К таким факторам можно отнести:

1) неправильная освещенность;
2) нарушение микроклимата;
3) шумового давления.

Вентиляция помещения хорошая так как установлены центральные кондиционеры, которые выполняют такие функции: вентилирование помещения здания, кондиционирование внутреннего пространства, очищение воздухопотока, увлажнение, подогрев, рециркуляция воздушных масс. Шум помещения в пределах нормы так как используется сервера и компьютеры нового поколения.

Помещение имеет размеры: длина (L) = 5 метров, ширина (B) = 4 метра, высота (H) = 3 метра. Помещение находиться в здании на 2-м этаже, рассчитано на 2 рабочих места.

План помещения выбранного для размещения оборудования и технического персонала изображен на рисунке 5.1.

Рисунок 5.1 – План рабочего помещения

5.2 Расчет естественного освещения

Геометрические размеры помещения: длина помещения L=5; ширина B=4; высота H=3. Коэффициенты отражения потолка, стен и пола: 50%, 30%, 10%. Противостоящее здание находится на расстоянии $P_{зд}$=30 м, $H_{зд}$=30 м.

Используется светильники типа TLPL228.2x36, в котором применяются люминесцентные лампы типа TL-D 58w/865. Боковое, одностороннее (т.к. B<12 м) естественное освещение. Площадь окна 1,43 м^2(L=1,3, B=1,1)

Нормируемое значение КЕО определяется по формуле (5.1):

$$e_N = e_H * m_N \quad (5.1)$$

e_N = 1*0.75=0.75
где e_H = 1;
e_H = 0.75 для Алматы при ориентации окон на север,
Кз = 1,5 коэффициент запаса по таблице 10.

Находим световую характеристику световых проемов η_o= 10,5 по таблице 2 при l = B – 1 м = 4 – 1 = 3 м; $\frac{L}{l} = \frac{5}{3} = 1,67$; $\frac{l}{h_1} = \frac{3}{2} = 1,5$

Световая характеристика - взаимозависимость фототока с значимостью световой струи постоянного спектрального состава - характеризует нелинейность фотоприемника [11].

Таблица 5.1 – Значение цветовой характеристики η_o окон при боковом освещении

Отношение длины помещения к его глубине	Значение световой характеристики при отношении глубины помещения к его высоте от уровня условной рабочей поверхности до верха окна							
	1	1,5	2	3	4	5	7,5	10
4 и больше	6,5	7	7,5	8	9	10	11	12,5
3	7,5	6	8,5	9,6	10	11	12,5	14
2	8,5	9	9,5	10,5	11,35	15	17	17
1,5	9,5	10,5	13	15	17	19	21	23
1	11	15	16	18	21	23	26,5	29
0,5	18	23	31	37	45	54	66	-

Коэффициент запаса – это подход некоторого определенного максимального усилия к наибольшему усилию, возникаемому в конструкции.

Максимальное напряжённость в системы никак не обязано быть выше дозволяемого усилия с целью этого использованного материала конкретного с учетом коэффициента запаса с целью установленных условий работы.

Коэффициент запаса – количество наибольшее единицы.

Таблица 5.2 - Коэффициенты отражения потолка и стен в зависимости от характера отражающей поверхности

Характер отражающей поверхности	Коэффициент отражения ρ, %
Побеленный потолок; побеленные стены с окнами, закрытыми белыми шторами	70
Побеленные стены при незавешенных окнах; побеленный потолок в сырых помещениях, чистый бетонный и светлый деревянный потолок	50
Бетонный потолок в грязных помещениях; деревянный потолок, бетонные стены с окнами; стены, оклеенные светлыми обоями	30
Стены и потолки в помещениях с большим количеством темной пыли; сплошное остекление без штор; красный кирпич неоштукатуренный; стены с темными обоями	10

Таблица 5.3 - Значения светового потока люминесцентных ламп

Тип лампы	Световой поток, лм
ЛДЦ 20	820
ЛД 20	920
ЛБ 20	1180
ЛДЦ 30	1450
ЛД 30	1640
ЛБ 30	2100
ЛДЦ 40	2100
ЛД 40	2340
ЛБ 40	3000
ЛДЦ 80	3560

Таблица 5.4 – Значение коэффициента запаса

Помещения	Примеры помещений	Коэффициент запаса k		
		Газоразрядные лампы	Лампы накаливания	Светодиодные светильники УСС
Запыленность свыше 5 мг/м3	Цементные заводы, литейные цеха и т. п.	2	1,7	1,5
Дым, копоть 1-5 мг/м3	Кузнечные, сварочные цеха и т. п.	1,8	1,5	1,3
Менее 1 мг/м3 Значительная концентрация паров кислот и щелочей	Инструментальные, сборочные цеха Цеха химических заводов, гальванические цеха	1,5 1,8	1,3	1,1 1,5
Запыленность значительно менее 1 мг/м3, отсутствие паров кислот и щелочей	Жилые, административные и офисные и т.п. помещения	1,4	1,5	1

Определяем общий коэффициент светопропускания по формуле (5.2):

$$\tau_o = \tau_1 * \tau_2 * \tau_3 * \tau_4 * \tau_5 \ldots = 0,8 * 0,7 * 0,8 * 1 = 0,45 \qquad 5.2)$$

где τ_1=0,8 по таблице (4);
τ_2=0,7 по таблице (5);
τ_3=0,8 по таблице (6);
τ_4=1 по таблице (7).

где τ_1- коэффициент светопропускания материала;

τ_2- коэффициент, учитывающий потери света в переплетах светопроема;

τ_3- коэффициент, учитывающий потери света в несущих конструкциях, при боковом освещении равен 1;

τ_4 - коэффициент, учитывающий потери света в солнцезащитных устройствах;

τ_5 - коэффициент, учитывающий потери света в защитной сетке, устанавливаемой под фонарями.

Таблица 5.5 – Значения коэффициента t_1

Вид светопропускающего материала	t_1
Стекло оконное листовое: -одинарное -двойное -тройное	0,9 0,8 0,75
Стекло витринное толщиной 6 - 8 мм	0,8
Стекло листовое армированное	0,6
Стекло листовое узорчатое	0,65
Стекло листовое со специальными свойствами: -солнцезащитное -контрастное	0,65 0,75
Органическое стекло: -прозрачное -молочное	0,9 0,6
Пустотелые стеклянные блоки светопрозрачные	0,5 0,55
Стеклопакеты	0,8

Таблица 5.6 – Значения коэффициента t_2

Вид переплета для окон промышленных зданий	t_2
Переплеты деревянные: одинарные спаренные двойные раздельные	 0,75 0,7 0,6
Переплеты стальные: одинарные открывающиеся одинарные глухие двойные открывающиеся двойные глухие	 0,75 0,9 0,6 0,8

Таблица 5.7 – Значения коэффициента t_3

Несущие конструкции покрытий	t_3
Стальные формы	0,9
Железобетонные и деревянные формы и арки	0,8
Балки и рамы сплошные при высоте сечения: 50 см и более Менее 50 см	 0,8 0,9

Таблица 5.8 – Значения коэффициента t_4

Солнце защитные устройства, изделия и материалы	t_4
Убирающиеся регулируемые жалюзи и шторы (межстекольные, внутренние, наружные)	1
Стационарные жалюзи и экраны с защитным углом не более 45° при расположении пластин жалюзи или экранов под углом 90° к плоскости окна: горизонтальные вертикальные	 0,65 0,75

Для этого находим:

отношение глубины помещения к высоте от уровня условной рабочей поверхности до верха окна - $\frac{l}{h_1} = \frac{3}{2} = 1{,}5$,

отношение глубины помещения к ширине помещения - $\frac{l}{B} = \frac{3}{4} = 0{,}75$,

отношение длины помещения к его глубине - $\frac{l}{B} = \frac{5}{3} = 1{,}67$,

величину средневзвешенного коэффициента отражения $p_{ср}$ потолка, стен и пола, при площади потолка $S_1 = 5 * 4 = 20м^2$,

площади стен $S_2 = 2 * 3 * 4 + 5 * 3 = 39м^2$,

площади пола $S_3 = 5 * 4 = 20м^2$,

$$p_{ср} = \frac{(50*20+30*39+10*20)}{(20+39+20)} = 30\%$$

r_1=1,15.

Таблица 5.9 – Определяем коэффициент r_1 – для бокового освещения

Отношение глубины помещения В к высоте от уровня условной рабочей поверхности до верха окна h_1	Отношение расстояния l расчетной точки от наружной стены к глубине помещения, В	Средневзвешенный коэффициент отражения потолка, стен и пола, $p_{ср}$								
		0,5			0,4			0,3		
		Отношение длины помещения l_n к его глубине В								
		0,5	1,0	2,0	0,5	1,0	2,0	0,5	1,0	2,0
1	2	3	4	5	6	7	8	9	10	11
1,00	0,10	1,02	1,02	1,02	1,12	1,01	1,01	1,00	1,00	1,00
1,00	0,50	1,47	1,42	1,33	1,28	1,25	1,20	1,09	1,08	1,07
1,00	1,00	2,59	2,43	2,11	1,95	1,86	1,67	1,32	1,29	1,22
3,00	0,10	1,07	1,06	1,05	1,04	1,04	1,03	1,01	1,01	1,01
3,00	0,20	1,23	1,20	1,16	1,14	1,12	1,10	1,05	1,04	1,03
3,00	0,30	1,51	1,46	1,36	1,31	1,28	1,21	1,10	1,09	1,07

Находим коэффициент $К_{зд}$, учитывающий затенение противостоящим зданием таблице 9 при отношении $Р_{зд}/Н_{зд}$=30/30=1 и $p_{ср}$=30%-$К_{зд}$=1,47.

Определяем площадь световых проемов по формуле (5.3):

$$S_0 = \frac{S_n * e_n * K_з * \mu_0 * К_{зд}}{100 * t_0 * r_1} = \frac{20 * 0{,}75 * 1{,}5 * 10{,}5 * 1{,}4}{100 * 0{,}45 * 1{,}15} = 6{,}8 \qquad (5.3)$$

Вывод: фактически оконный проем не соответствует нормированным значениям. Необходимо проанализировать искусственное освещение и провести его реконструкцию.

5.3 Расчет системы искусственного освещения помещения

Помещения имеет естественное освещение через одно боковое окно, и искусственное освещение, которое позволяют вести работы в темное время суток и днем в местах, где показатель КЕО не соответствует нормативам.

Определим расчетную высоту подвеса по формуле (5.4):

$$h_р = Н - (h_{раб} + h_{св}) \qquad (5.4)$$

где Н – 3 м высота помещения;

$h_{раб}$ – уровень рабочей поверхности, $h_{раб}$ = 0,8

$h_{св}$– расстояние между светильником и потолком, $h_{св}$= 0,3

Тогда $h_р$ = 3 – (0,3 + 0,8) = 1,9м.

L–расстояние между соседними светильниками или рядами (если по длине (А) и ширине (В) помещения расстояние различные то они

обозначаются L_Aи L_B, *l*–расстояние от крайних светильников или рядов до стены.

Оптимальное решение *l* крайнего ряда светильников до стены рекомендуется принимать равным (0,3-0,5)L.

Таблица 5.10 – Значения λ для разных типов светильников

Типовая кривая	Значения λ	
	Рекомендуемое	Наибольшее допустимое
Концентрированная (К)	0,4-0,7	0,9
Глубокая (Г)	0,8-1,2	1,4
Косинусная (Д)	1,2-1,6	2,1
Равномерная(М)	1,8-2,6	3,4
Полуширокая(Л)	1,4-2,0	2,3

Расстояние между светильниками L определяется как $L = \lambda * h_p$,
$L_A = 1,2 * 1,9 = 2,28$, $l_a = 1,36$, $\lambda_{1/2} = 1,2 - 0,8$
$L_B = 0,8 * 1,9 = 1,52$, $l_b = 1,24$.

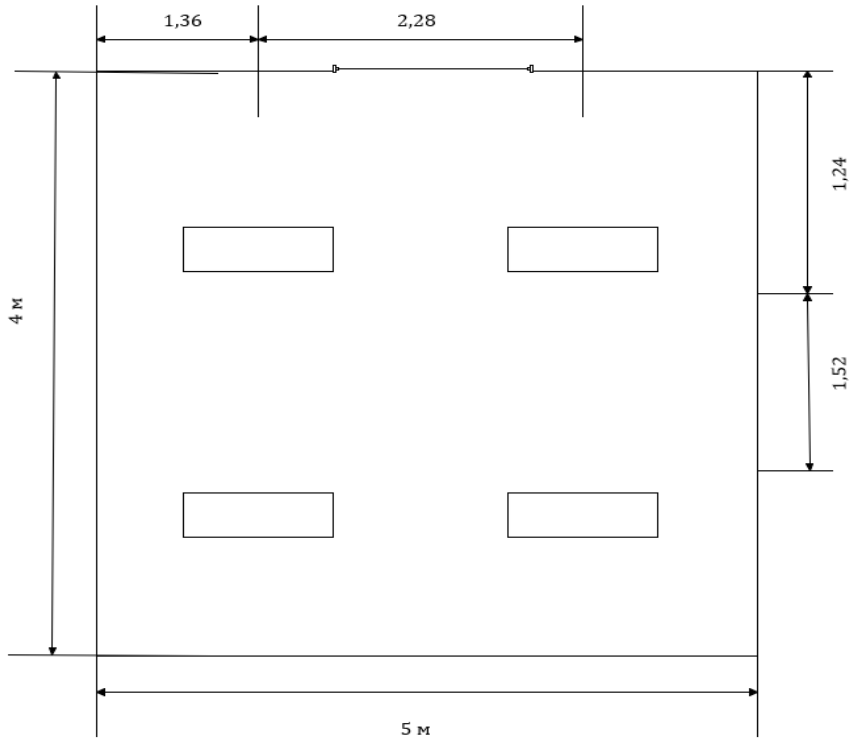

Рисунок 5.2 – Схема расстояния между светильниками

Намечаем контрольную точку А. Находим проекцию расстояния на потолок от точки А до светильника – d. Далее определяем угол между

потолком и прямой d. По этому углу находим силу света от каждого источника и освещенность помещения относительно расчетной точки.

Горизонтальная освещенность в точке А от одного светильника определяется формулой (5.5):

$$e_i = \frac{l_\alpha * cos^3(\alpha)}{h^2} \quad (5.5)$$

где α = arctg($\frac{d_i}{h}$);

рассчитаем необходимые параметры для каждого светильника:

Лампы 1,2,3,4:

$$d_1 = \sqrt{(2.28)^2+(1.51)^2} = 2.74 \alpha = \text{arctg}\ (\frac{2.74}{1.9}) = 55.2$$

По рассчитанным углам определяем силу света и освещенность:

$$e_1 = \frac{83.44 * cos^3(55.2)}{(1.9)^2} = 4.3 \text{ лк};\ l_\alpha = 83.44 \text{ кд при } 55{,}2^\circ$$

Суммарная условная освещенность равна:

$$\sum E = e_1 * 4 = 4{,}3 * 4 = 17{,}2 \text{ лк}$$

Расчетная формула освещенности в точке А от нескольких светильников принимает следующий вид:

$$E_{АГ} = \frac{\mu * F_л}{100 * К_з} \sum_1^n e_{АГn},$$

где μ – коэффициент, учитывающий освещенность от удаленных светильников и отраженный световой поток от стен, потолка и расчетной поверхности. Этот коэффициент вводится как поправочный, чтобы избежать завышения мощности ламп.

При эмалированных светильниках прямого света μ = 1.1 – 1.2.

$$E_{АГ} = \frac{5000 * 2 * 1{,}2 * 17{,}2}{1000 * 1{,}5} = 137{,}6 \text{ лк}$$

Освещенность на рабочем месте считается не достаточной, следовательно, производим реконструкцию освещенности.

5.4 Расчёт освещение помещение по методу коэффициента использования

Определим индекс помещения (*i*) по формуле (5.6):

$$i = \frac{S}{h * (A+B)}, \quad (5.6)$$

где S, A, B – соответственно длина, ширина и площадь помещения.

$$i = \frac{4 * 5}{1{,}9 * (4+5)} = 1{,}169$$

Расчетный поток ИС определяется по формуле (5.7):

$$Ф = \frac{E_{min} * S * z * K}{N * \mu}, \quad (5.7)$$

где N – число ИС;
K – коэффициент запаса;
z – коэффициент минимальной освещенности (отношение средней и минимальной освещенности).
E_{min}= 300 лк III, б разряд зрительных работ по таблице 5.11;

Таблица 5.11 – Требования к освещению помещении промышленных предприятий

Характеристика зрительной работы	Наименьший или эквив. размер объекта различения, мм	Разряд зрительной работы	Подразряд зрительной работы	Характеристика фона	Искусственное освещение				
					Освещенность, лк			Сочетание нормируемых величин показателя ослепленности и коэффициента пульсации	
					При системе комбинированного освещения		при системе общего освещения		
					Всего	В том числе от общего		Р	Кп, %
Высокой точности	От 0,30 До 0,50	III	а	Темный	2000 1500	200 200	500 400	40 20	15 15
			б	Средний Темный	1000 750	200 200	300 200	40 20	15 15
			в	Светлый Средний Темный	750 600	200 200	300 200	40 20	15 15
			Г	Светлый << Средний	400	200	200	40	15

$$Ф = \frac{300*20*1,1*1,5}{4*0,7} = 3535 \text{ лм}$$

Принимаем лампу TL-D36W/840 и проводим расчет необходимое количество светильников по формуле (5.8):

$$Ф = \frac{E_{min}*S*z*К_з}{Ф*\mu}, \quad 5.8)$$

$Ф_л$= 3450 лм для лампы типа TL-D36W/840;

$$N = \frac{300*20*1.1*1.5}{3450*0.7} = 4$$

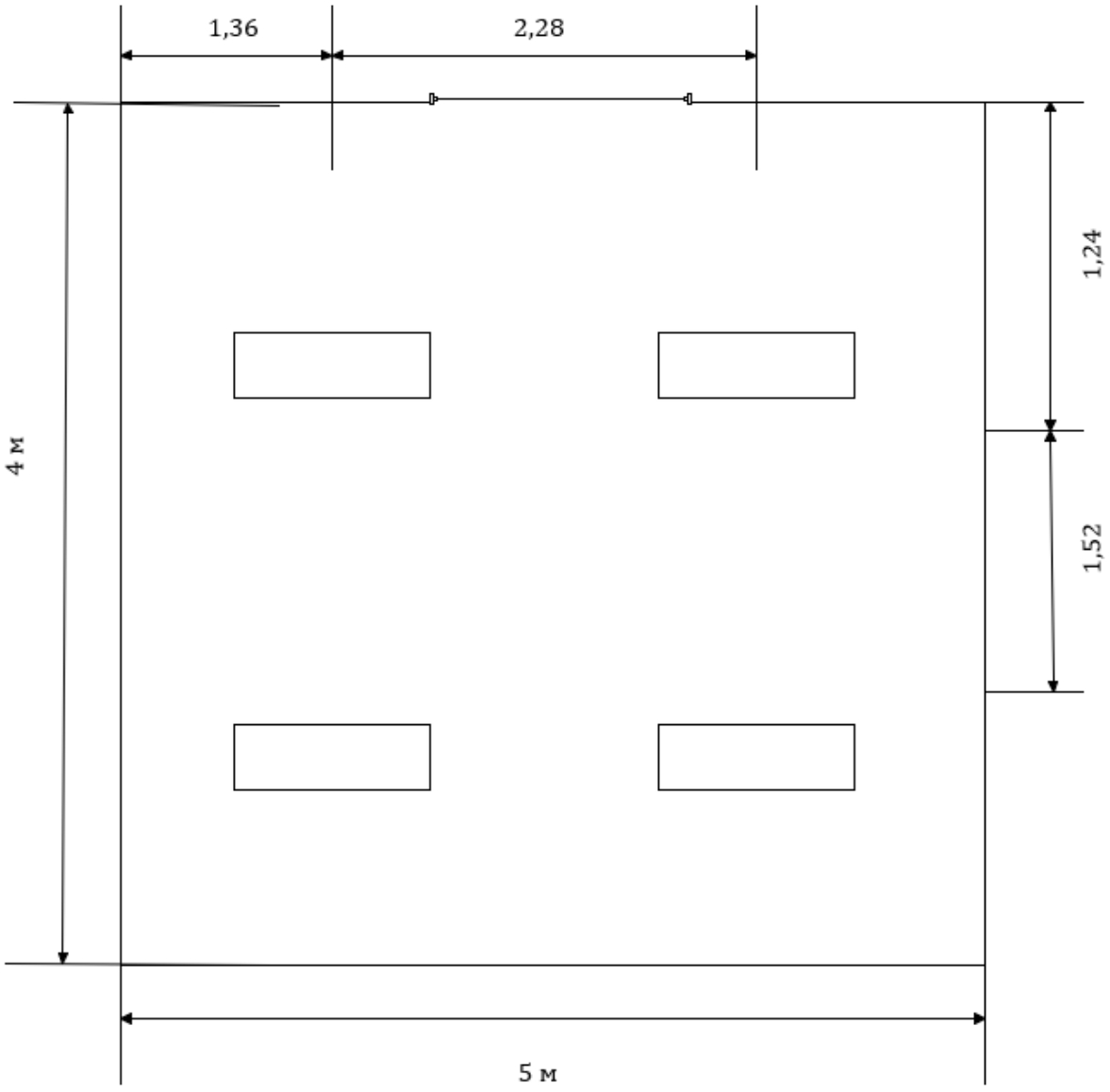

Рисунок 5.3 – Схема расстояния между светильниками

Вывод

На основе полученных расчетов приходим к выводу, что в данном помещении при заданных условиях работы необходимо установить 4 светильников типа TLPL36/41-827.

Спроектирована система освещения для рабочей поверхности. В качестве источника света был выбран тип светильников с учетом нормализованного освещения помещений, а также было выбрано и рассчитано их расположение и количество.

Заключение

В ходе работы был произведен анализ существующих подходов к обеспечению безопасности корпоративных серверов. Был рассмотрен ряд передового ПО для защиты в данной области. Был проработан ряд существенных моментов:

1) обеспечение безопасного доступа к устройствам;
2) обеспечение безопасности поднятых серверов;
3) обеспечение безопасности ПК.

Проведя анализ предметной области для защиты USB-портов я пришел к выводу, что блокировка USB-портов является необходимой мерой для защиты ноутбуков, персональных данных компьютеров и рабочих станций корпоративной сети с целью предотвращения утечки информации, порчи и кражи личных данных, других вредоносных действий злоумышленников. И поэтому в проекте реализовал практическую защиту от данной уязвимости.

Процесс реализации был разбит на несколько этапов. Я составил функциональную схему работы программы. Затем я приступил к созданию приложения для защиты USB-портов на языке Python под названием «kinza», которое выполняет защитные функции предотвращения утечки информации по USB-портам в операционной системе Linux Ubuntu:

1) включение USB-портов;
2) выключение USB-портов;
3) включение режима «только для чтения»;
4) поиск подключенного USB-устройства;
5) очистка терминала;
6) вывод всех блочных устройств, подключенных к системе;
7) вывод всего подключенного оборудования;
8) список подключенного USB-флеш;
9) отмонтирования USB-флеш;
10) форматирование USB-флеш.

Мною разработанное приложение «kinza» для применения на корпоративном сервере являет собой реализацию на актуальную и важную тематику.

Программа для блокировки USB-порта может полностью выполнять свои функции, отлажена и работоспособна. Блокировка USB-портов – одна из функций ПО и может быть использована для предотвращения подключений неавторизованных USB носителей, копирования информации. Протестировав функционал приложения для защиты USB-портов можно сказать, что данное ПО успешно справляется с поставленной задачей.

Какова бы ни была система защиты информации в конкретной организации, главное – помнить два основных правила. Первое: комплексная защита информации – это совокупность принятых в компании мер по защите, а не набор продуктов. Нельзя списывать со счетов и то, что в обеспечении информационной безопасности участвует каждый сотрудник компании.

Второе: основа любой системы защиты – люди. От того, насколько грамотно персонал настроит эксплуатируемые системы, как он готов реагировать на инциденты в области безопасности, зависит защищенность предприятия в целом.

Что же касается технологий и конкретных средств защиты информации, то использование антивирусов, межсетевых экранов и механизмов разграничения доступа обеспечивает лишь минимально необходимый уровень защищенности.

Список литературы

1 Алексеенко К. Web-сервер глазами хакера. – СПб.: БХВ-Петербург, 2019.

2 Романов П.Ю, Лисьев Г.А Программное обеспечение компьютерных сетей и web-серверов. – М.: Инфра-М, 2015.

3 Жуков Ю. Основы WEB-хакинга. Нападение и защита. – Воронеж: 2012.

4 Гайкович В.Ю., Ершов Д.В. «Основы безопасности информационных технологий», – М.: МИФИ, 2014.

5 А. Астахов «Анализ защищенности корпоративных автоматизированных систем», М.: Информационный бюллетень Jet Info N7-2014.

6 А. В. Соколов, В. Ф. Шаньгин Защита информации в распределенных корпоративных сетях и системах, ДМК Пресс, 2013.

7 Wiki Википедия серверов и хостинга // VPS.UA: Настройка Fail2ban URL: https://vps.ua/wiki/install-linux-vps/security/configuring-fail2ban (дата обращения 30.04.2019).

8 Полежаев П.Н., Малахов А.К., Сагитов А.М. «Ахиллесова пята» USB-устройств: атака и защита // Философские проблемы информационных технологий и киберпространства. № 1(9), 2015.

9 Агуров П.В. Интерфейс USB. Практика использования и программирования. - СПб.: БХВ-Петербург, 2014.

10 Хакимжанов Т.Е. Расчет аспирационных систем. Дипломное проектирование. Для студентов всех форм обучения всех специальностей. – Алматы: АУЭС, 2014.

11 Бекишева А.И. Методические указания к выполнению экономической части дипломной работы для бакалавров специальности 5В0703 – Информационные системы – Алматы: АУЭС, 2013.

12 Vscale Communnity // COMMUNITY.VSCALE.IO: Настройка Nginx в качестве обратного прокси к Apache URL: https://community.vscale.io/hc/ru/community/posts/208808369 (дата обращения 30.04.2019).

13 А.В. Лукацкий Обнаружение атак. - СПб.: БХВ-Петербург, 2013.

14 А. Ю. Щеглов Защита компьютерной информации от несанкционированного доступа. Наука и Техника. № 1(4), 2013.

15 В. В. Домарев Безопасность информационных технологий. Методология создания систем защиты, ТИД "ДС", № 2(2), 2012.

Перечень сокращений

BSD – Berkeley Software Distribution
CD – Compact Disc
DDOS – Distributed Denial of Service
DHCP – Dynamic Host Configuration Protocol
FTP – File Transfer Protocol
HTTP – Hyper Text Transfer Protocol
HTTPS – Hypertext Transfer Protocol Secure
IIS – Internet Information Services
IP – Internet Protocol
ISP – Internet Service Provider
SSL – Secure Sockets Layer
TLS – transport layer security
UFW – Uncomplicated Firewall
USB – Universal Serial Bus
WEB – World Wide Web
ОС – Операционная система
ПО – Программное обеспечение

Приложение А

Программа по защите USB-портов

```
import subprocess
import os
import glob
import os.path
from tkinter import*
def button_off():
   subprocess.call(["pkexec", "chmod", "0700", "/media"])
def button_on():
   subprocess.call(["pkexec", "chmod", "0777", "/media"])
def button_read():
   subprocess.call(["pkexec", "chmod", "0755", "/media"])
def button_listUSB():
   subprocess.call(["pkexec", "df", "-h"])
def button_clearTerminal():
   subprocess.call(["pkexec", "clear"])
def button_blockDevice():
   subprocess.call(["pkexec", "lsblk"])
def button_connUSB():
   subprocess.call(["sudo", "dmesg"])
def button_USB():
   subprocess.call(["ls /dev/disk/by-id/usb*"], shell=True)
def button_unmount():
   subprocess.call(["sudo udisksctl unmount -b /dev/sdb4"], shell=True)
def button_format():
   subprocess.call(["sudo mkfs -t vfat -n FLASH /dev/sdb4*"], shell=True)
root = Tk()
root.title("kinza")
root.geometry("170x310")
root.configure(background='silver')
Button(root, text="on", command=button_on, height=1, width=9).pack()
Button(root, text="off", command=button_off, height=1, width=9).pack()
Button(root, text="read", command=button_read, height=1, width=9).pack()
Button(root, text="listUSB", command=button_listUSB, height=1, width=9).pack()
Button(root, text="clearTerminal", command=button_clearTerminal, height=1,
width=9).pack()
Button(root, text="blockDevice", command=button_blockDevice, height=1,
width=9).pack()
Button(root, text="connUSB", command=button_connUSB, height=1, width=9).pack()
Button(root, text="USB", command=button_USB,height=1, width=9).pack()
Button(root, text="unmount", command=button_unmount,height=1, width=9).pack()
Button(root, text="format", command=button_format,height=1, width=9).pack()
root.mainloop()
```