

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Вычислительные системы и информационная безопасность»

|  |  |  |
| --- | --- | --- |
| Зав. кафедрой | | «ВСиИБ» |
| \_\_\_\_\_\_\_\_\_\_\_\_ | | А.Р. Газизов |
| подпись | | И.О. Фамилия |
| «\_\_\_» | \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. | |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Тема «КОНТЕНТНАЯ ФИЛЬТРАЦИЯ ТРАФИКА С ПОМОЩЬЮ ПРОКСИ-СЕРВЕРА SQUID»

Направление подготовки 10.05.02 Информационная безопасность телекоммуникационных систем

Специализация Защита информации в системах связи и управления

Обозначение ВКР 10.05.02.860000.000 группа ВИБТ62

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Г.А. Муратов

(подпись, дата) И.О.Ф.

Руководитель ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_ доц., к.т.н. В.В. Галушка

(подпись, дата) должность, И.О.Ф.

Консультанты по разделам:

Безопасность и экологичность проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ доц., к.т.н.

(подпись, дата) (должность, И.О.Ф.)

Технико-экономическое обоснование \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ проф., д.э.н.

(подпись, дата) (должность, И.О.Ф.)

Нормоконтроль \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ст. преп. М.А. Ганжур

(подпись, дата) (должность, И.О.Ф.)

Ростов-на-Дону

2023



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Вычислительные системы и информационная безопасность»

|  |  |  |
| --- | --- | --- |
| Зав. кафедрой | | «ВСиИБ» |
| \_\_\_\_\_\_\_\_\_\_\_\_ | | А.Р. Газизов |
| подпись | | И.О. Фамилия |
| «\_\_\_» | \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. | |

**ЗАДАНИЕ**

на выполнение выпускной квалификационной работы

Тема «КОНТЕНТНАЯ ФИЛЬТРАЦИЯ ТРАФИКА С ПОМОЩЬЮ ПРОКСИ-СЕРВЕРА SQUID»

Обучающийся Муратов Григорий Александрович

Обозначение ВКР 10.05.00.860000.000 группа ВИБТ62

Тема утверждена приказом по ДГТУ от «23» июня 2022 г. № 2831-ЛС-О

Срок представления ВКР к защите «23» января 2023 г.

Исходные данные для выполнения выпускной квалификационной работы:

Методы и средства обмена информацией в сети интернет: протоколы HTTP, HTTPS, TCP; наиболее актуальные угрозы информационной безопасности в сети интернет; требования к защите данных в автоматизированных системах; прокси-сервер на основе операционной системы Linux; принципы функционирования средств фильтрации веб-контента.

Содержание выпускной квалификационной работы

Введение: Во введении необходимо изложить актуальность выбранной темы, обозначить объект и предмет исследования, цель и задачи выпускной квалификационной работы, теоретическую и практическую значимость работы, структуру работы.

Наименование и краткое содержание разделов:

1. Задача использования прокси-серверов для фильтрации трафика. Технологии организации общего доступа в интернет из локальной сети. Понятие и функции прокси-сервера. Задачи фильтрации сетевого трафика. Существующие средства фильтрации сетевого трафика. Kerio winroute firewall. TrustAccess. Fortinet FortiGate. Cisco Firepower NGFW. Squid.

2. Методы фильтрации трафика прокcи-сервером squid. Возможности Squid по управлению сетевыми соединениями. Понятие контентной фильтрации. Фильтрация на стороне сервера. Фильтрация на стороне клиента. Фильтрация на уровне провайдера. Поисковая фильтрация. Проблема фильтрации шифрованного трафика. Технологии анализа шифрованного трафика в прокси-сервер Squid.

3. Практическая реализация системы защиты. Програмно-аппаратная платформа для реализации системы фильтрации. Генерация сертификатов. Распределение прав доступа между сервисами. Правила фильтрации содержимого web-страниц. Пример работы системы.

4. Безопасность и экологичность проекта. Эргономические основы оформления помещений. Организация освещения и его соответствие нормативным требованиям. Расчет общего или местного освещения. Экологичность работы. Организация и обеспечение пожарной безопасности на предприятии. Выводы.

5. Технико-экономическое обоснование. Спецификация проекта. План-график проектирования и разработки системы. Расчет затрат на разработку проекта.

Заключение: Заключение должно содержать обобщенные результаты проведенной работы в соответствии с поставленной целью и задачами, необходимо указать чем завершается работа – усовершенствованием, модернизацией, дать свои предложения.

Перечень графического и иллюстративного материалов:

1. Схемы работы прокси

2. Схема выполнения процедуры «рукопожатия» в протоколе TLS

3. Структура сертификата файла squid.pem

4 Результат настройки прокси-севрера

Руководитель ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_ доц., к.т.н. В.В. Галушка

(подпись, дата)

Задание к исполнению принял \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Г.А. Муратов

(подпись, дата)

**Аннотация**

Выпускная квалификационная работа посвящена проблеме обеспечения внутрисетевой безопасности информационных систем. В работе проведен анализ возможных способов внедрения прокси-серверов и особенностей фильтрации сетевого трафика.

На основе полученных данных предложен проект по внедрению прокси-сервера Squid с использованием операционной системы Linux дистрибутива Ubuntu.

Объем текстового материала 92 листа (А4), количество иллюстраций 35, таблиц — 12, использованных источников — 20.

**Abstract**

The final qualifying work is devoted to the problem of ensuring the intra-network security of information systems. The paper analyzes possible ways of implementing proxy servers and features of filtering network traffic.

Based on the data obtained, a project is proposed to implement a Squid proxy server using the Linux operating system of the Ubuntu distribution.

The volume of text material is 92 sheets (A4), the number of illustrations is 35, tables — 12, sources used — 20.

**Содержание**

[Введение 6](#_Toc124859761)

[1 Задача использования прокси-серверов для фильтрации трафика 9](#_Toc124859762)

[1.1 Технологии организации общего доступа в интернет из локальной сети 9](#_Toc124859763)

[1.2 Понятие и функции прокси-сервера 13](#_Toc124859764)

[1.3 Задачи фильтрации сетевого трафика 17](#_Toc124859765)

[1.4 Существующие средства фильтрации сетевого трафика 18](#_Toc124859766)

[1.4.1 Kerio winroute firewall 19](#_Toc124859767)

[1.4.2 TrustAccess 20](#_Toc124859768)

[1.4.3 Fortinet FortiGate 20](#_Toc124859769)

[1.4.4 Cisco Firepower NGFW 21](#_Toc124859770)

[1.4.5 Squid 22](#_Toc124859771)

[2 Методы фильтрации трафика прокcи-сервером squid 23](#_Toc124859772)

[2.1 Возможности Squid по управлению сетевыми соединениями 23](#_Toc124859773)

[2.2 Понятие контентной фильтрации 24](#_Toc124859774)

[2.2.1 Фильтрация на стороне сервера 25](#_Toc124859775)

[2.2.2 Фильтрация на стороне клиента 26](#_Toc124859776)

[2.2.3 Фильтрация на уровне провайдера 26](#_Toc124859777)

[2.2.4 Поисковая фильтрация 27](#_Toc124859778)

[2.3 Проблема фильтрации шифрованного трафика 28](#_Toc124859779)

[2.4 Технологии анализа шифрованного трафика в прокси-сервер Squid 34](#_Toc124859780)

[3 Практическая реализация контентной фильтрации трафика 38](#_Toc124859781)

[3.1 Програмно-аппаратная платформа для реализации системы фильтрации 38](#_Toc124859782)

[3.2 Генерация сертификатов 43](#_Toc124859783)

[3.3 Правила фильтрации содержимого web-страниц 52](#_Toc124859784)

[3.4 Пример работы системы 58](#_Toc124859785)

[4 Безопасность и экологичность проекта 66](#_Toc124859786)

[4.1 Эргономические основы оформления помещений 66](#_Toc124859787)

[4.2 Экологичность работы 70](#_Toc124859788)

[4.3 Организация и обеспечение пожарной безопасности на предприятии 75](#_Toc124859789)

[4.4 Выводы 78](#_Toc124859790)

[5 Технико-экономическое обоснование 80](#_Toc124859791)

[5.1 Спецификация проекта 80](#_Toc124859792)

[5.2 План-график проектирования и разработки системы 81](#_Toc124859793)

[5.3 Расчет затрат на разработку проекта 86](#_Toc124859794)

[Заключение 90](#_Toc124859795)

[Перечень использованных информационных ресурсов 91](#_Toc124859796)

# Введение

В настоящее время интернет-атаки становятся более популярными среди злоумышленников, пытающихся украсть, изменить или уничтожить данные,   
а также вывести из строя информационную систему Компании. Прибегая   
к методам фильтрации входящего и исходящего сетевого трафика, минимизируется не только риск взаимодействия злоумышленниками   
с ресурсами предприятия, но и позволяет контролировать обращения сотрудников к потенциально опасным ресурсам в глобальной сети Интернет.

Кибер-злоумышленники знают, как манипулировать данными, проходящими через интернет-соединение. Они могут перенаправить   
интернет-трафик или проникнуть в него, чтобы заразить компьютер или сеть   
из компьютеров вредоносным программным обеспечением (ПО). Не стоит забывать о том, что и антивирусное ПО не всегда может обнаружить определенные вредоносные программы, такие как программы-вымогатели.

Существует множество типов кибератак, каждая из которых может быть нацелена на конкретную компьютерную систему для различных целей.

В зависимости от контекста, кибератаки могут быть частью кибервойны или кибертерроризма. Кибератака может быть использована суверенными государствами, отдельными лицами, группами, обществом или организациями, и она может исходить из анонимного источника. Продукт, облегчающий кибератаку, иногда называют кибероружием.

Растущая озабоченность корпораций по поводу конфиденциальности данных, производительности труда сотрудников и юридической ответственности в совокупности способствовала внедрению решений   
для веб-безопасности. Их способность обеспечивать сетевую безопасность путем ограничения доступа к веб-сайтам, подверженным вирусам   
или загруженным вредоносным содержимым, еще больше способствовала более широкому внедрению решений для фильтрации веб-контента среди корпоративных пользователей.

Таким образом, решения для фильтрации веб-контента стали неотъемлемым компонентом бизнес-организаций. В 2021 году сегмент   
бизнес-организаций внес наибольший вклад в рынок фильтрации веб-контента, на его долю пришлось более 35% выручки мирового рынка. Хотя ожидается, что этот сегмент останется основным рынком конечного использования,   
в сегменте школ и учреждений ожидается максимальный рост в течение прогнозируемого периода с 2022 по 2030 год. Чтобы ограничить доступ учащихся к информации, не связанной с учебной программой, и лучше управлять выделенного потребления полосы пропускания, школы   
и учреждения по всему миру все чаще внедряют решения для фильтрации   
веб-контента в более широком масштабе.

В 2021 году фильтрация URL-адресов была основным методом фильтрации веб-контента, используемым бизнес-организациями   
и федеральными агентствами, на долю которого приходилось около трети мирового рынка фильтрации веб-контента. Ежедневно во Всемирную паутину WWW (World Wide Web) добавляется большое количество контента,   
что приводит к увеличению количества URL-адресов, что делает фильтрацию на основе списков (URL-фильтрацию) неустойчивой. Ожидается, что в течение прогнозируемого периода фильтрация по ключевым словам и типам файлов получит широкое распространение. Рост методов фильтрации типов файлов можно объяснить растущим спросом со стороны коммерческих организаций, школ и федеральных агентств. Ожидается, что из-за технических сложностей и меньшей безопасности методы фильтрации Интернет-протокола (IP) и DNS (системы доменных имен) будут демонстрировать более медленный рост в течение прогнозируемого периода 2022 и 2030 годов.

С появлением технологий решения веб-фильтрации стали более гибкими и масштабируемыми, способными решать сложные проблемы безопасности в сетевом деловом мире. Сегодня фильтры веб-контента обеспечивают более точную классификацию и более безопасную архитектуру. Кроме того, фильтры веб-контента стали лучше оценивать контент и предлагают простую установку. Поставщики решений для обеспечения безопасности по всему миру сосредоточены на разработке многофункциональных решений, которые помимо функции фильтрации веб-контента также могут функционировать в качестве межсетевых экранов, решений для предотвращения вторжений и антивирусных решений, среди прочего. Кроме того, такие функции, как ведение журналов, аудит, прокси-сервер и защита от потери данных, постепенно находят применение в решениях веб-безопасности.

Целью дипломной работы является повышения уровня защиты сети организации путем контроля трафика с помощью средств фильтрации контента глобальной сети Интернет.

Для достижения поставленной цели были решены следующие задачи:

— выполнен анализ возможных способов внедрения прокси-серверов;

— выполнен анализ особенностей фильтрации сетевого трафика;

— предложены способы практической реализации фильтрации веб-контента;

— проведена оценка эффективности предложенного метода защиты.

# 1 Задача использования прокси-серверов для фильтрации трафика

## 1.1 Технологии организации общего доступа в интернет из локальной сети

Сеть — это любое количество независимых компьютерных систем, связанных между собой так, чтобы был возможен обмен данными. Сетевые системы должны быть связаны не только физически, но и логически. Даже два компьютера, соединенных друг с другом, можно классифицировать как сеть.

Сети настраиваются для передачи данных из одной системы в другую или для совместного использования ресурсов, таких как серверы, базы данных и принтеры в сети. В зависимости от размера и диапазона компьютерной сети можно различать разные размеры сети. К наиболее важным типам сетей относятся:

Персональные сети (PAN) – подразумевает соединение электронных устройств в непосредственной близости от пользователя (рисунок 1.1);

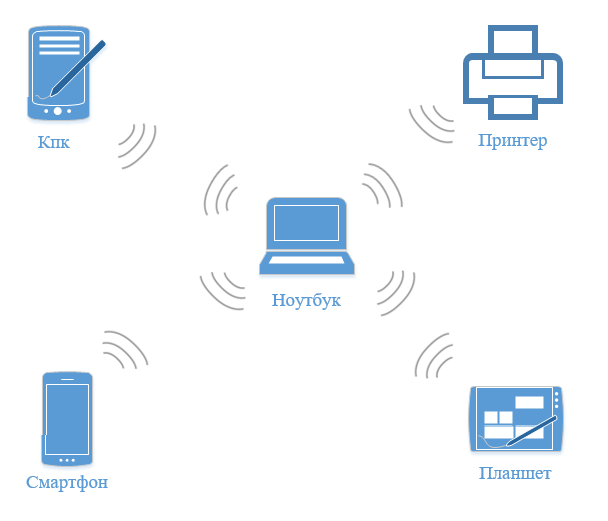


Рисунок 1.1 – Персональные сети (PAN)

Локальные сети (LAN) – группа компьютеров и других устройств, соединенных вместе по сети и находящихся в одном месте (рисунок 1.2);

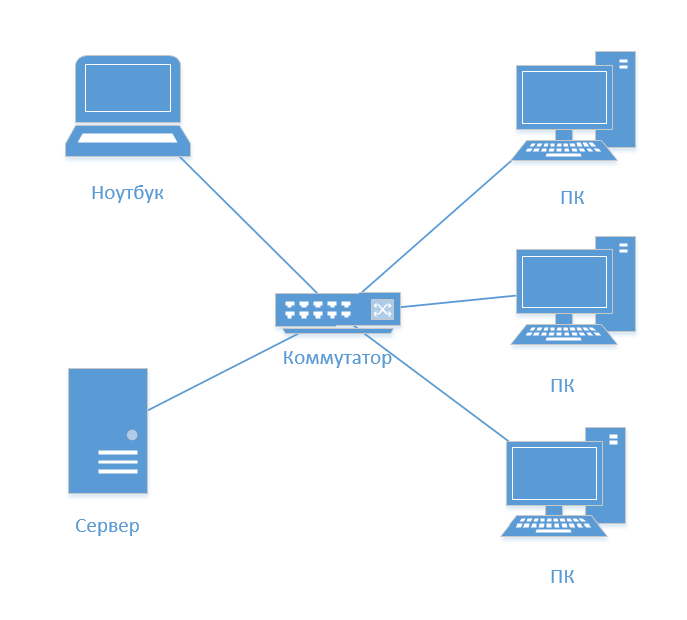


Рисунок 1.2 – Локальные сети (LAN)

Городские сети (MAN) – это широкополосная телекоммуникационная сеть, которая соединяет несколько локальных сетей в непосредственной близости (рисунок 1.3). MAN больше, чем локальная сеть (LAN), но меньше, чем глобальная сеть (WAN);

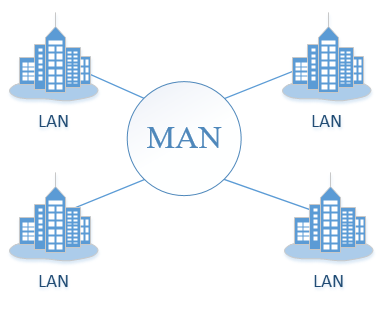


Рисунок 1.3 – Городские сети (MAN)

Глобальные сети (WAN) – сеть, определяемая как компьютерная сеть, которая соединяет более мелкие сети. Глобальная сеть включает в себя связанный набор телекоммуникационных сетей, которые распределены по большой географической области, охватывающей несколько городов, территорий или стран (рисунок 1.4);

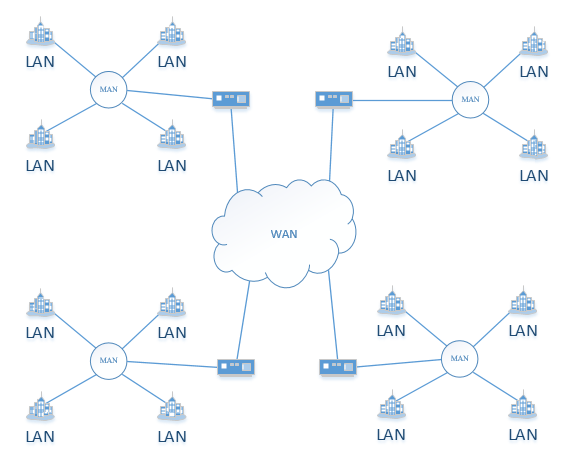


Рисунок 1.4 – Глобальные сети (WAN)

Глобальные сети (GAN) – относится к любой сети, которая состоит из различных взаимосвязанных компьютерных сетей (WAN) и также охватывает неограниченную географическую область (рисунок 1.5).

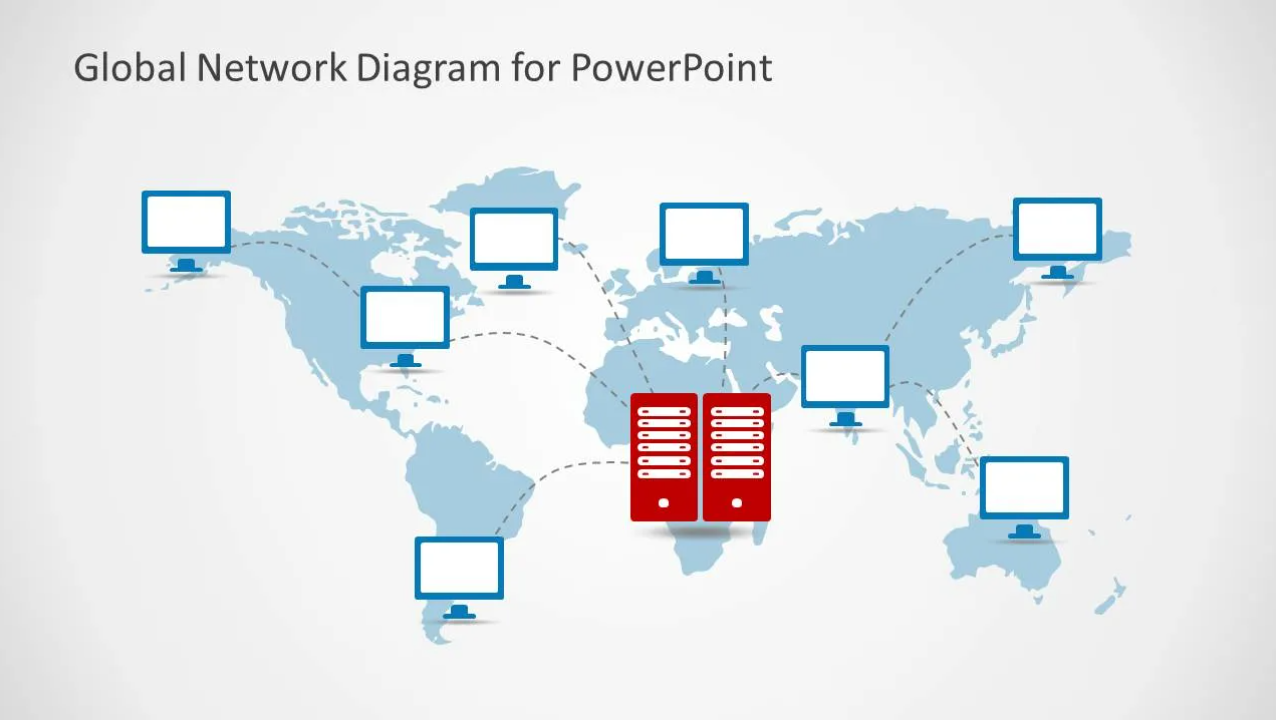


Рисунок 1.5 – Глобальные сети (GAN)

Когда компьютеру необходимо подключиться к другому компьютеру, ему требуется IP-адрес. Этот IP-адрес является уникальным номером, идентифицирующим местоположение компьютера в сети. Ранее, когда впервые появилось понятие IP-адресации, предполагалось, что будет более чем достаточно адресов, чтобы охватить всех. Поскольку количество узлов в Интернете растет и продолжает расти каждый день, в следствие чего IP-адресов не будет достаточно для развития глобальной сети Интернет. Используя технологию трансляции адресов NAT, решаются проблемы с подключением слишком большого количества устройств к Интернету. Основная цель NAT — взять любое количество устройств, которые принадлежат одной локальной сети, и позволить им использовать один и уникальный IP-адрес в Интернете.

Прокси — это сервер, расположенный между клиентом (ищущим ресурс) и другим сервером, выступающим в роли посредника. Клиент, запрашивающий ресурс, подключается к прокси-серверу, и прокси-сервер оценивает запрос в соответствии со своими правилами фильтрации. Если запрос подтвержден, прокси подключается к серверу и предоставляет запрошенный ресурс клиенту. Кроме того, прокси можно использовать для фильтрации запросов и предотвращения доступа к определенным веб-сайтам.

NAT изменяет IP-адрес в заголовке IP-пакета, когда он проходит через устройство маршрутизации, и позволяет использовать другой набор IP-адресов для трафика внутри локальной сети. это IP-адреса для внешнего трафика, а сервер, расположенный между клиентом и другим сервером и выступающий в качестве посредника. Для работы NAT не требуется никакого специального прикладного программного обеспечения, в то время как приложения за прокси-сервером должны поддерживать прокси-сервисы и должны быть настроены для использования прокси-сервера.

## 1.2 Понятие и функции прокси-сервера

Прокси-сервер является промежуточным звеном между компьютером, используемым клиентом, и системой интернет-серверов. Его основная задача — транслировать запросы пользователей в Сеть и отправлять обратно полученные ответы. Обращаясь к внешним ресурсам в Интернете, веб-сайты могут получить следующую информацию о клиенте:

* данные о местоположении;
* загруженные файлы;
* время, проведенное на ресурсе;
* данные об используемом интернет-провайдере;
* имя модели компьютера и т.д.

Прокси-серверы маскируют источник запросов, чтобы сделать соединения безопасными. Каждый раз при обращении к веб-сайту создается и отправляете веб-запрос, который перенаправляется на прокси-сервер. Затем прокси-сервер изменяет и шифрует передаваемые данные, такие как IP-адрес, и отправляет их на удаленный веб-ресурс.

После этого веб-сайт отправляет свои данные обратно на прокси-сервер, который их обрабатывает, прежде чем передать информацию клиенту о веб-странице. Таким образом, прокси-сервер работает как брандмауэр или веб-фильтр. По принципу работы разделяют прямые и обратные прокси-сервера.

Прямые прокси-серверы обычно используются внутренними сетями. Как только один из клиентов отправляет запрос на подключение к определенному веб-сайту, он сначала должен пройти через прямой прокси-сервер, который определяет, разрешено ли клиенту обращаться к этому ресурсу. Если да, запрос на подключение отправляется на внешний сервер, который не видит IP-адрес клиента, а видит только запрос на подключение, отправленный с прямым прокси-сервером.

Прямой прокси-сервер обеспечивает полный административный контроль над локальными сетевыми подключениями. Он действует как щит или брандмауэр, который позволяет администраторам ограничивать доступ клиентов внутренней сети к нежелательным веб-ресурсам (рисунок 1.6).

В отличие от прямого прокси-сервера, обратный прокси-сервер работает на стороне веб-сайта (или веб-службы), скрывая IP-адреса во внутренней сети от внешних пользователей (рисунок 1.7). Обратный прокси-сервер решает, могут ли веб-клиенты видеть содержимое веб-сайта или использовать веб-службу или нет.

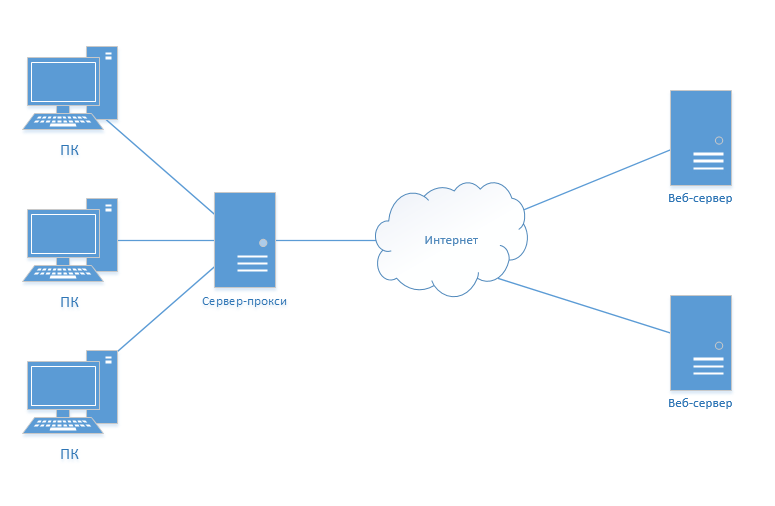


Рисунок 1.6 – схема работы прямого прокси

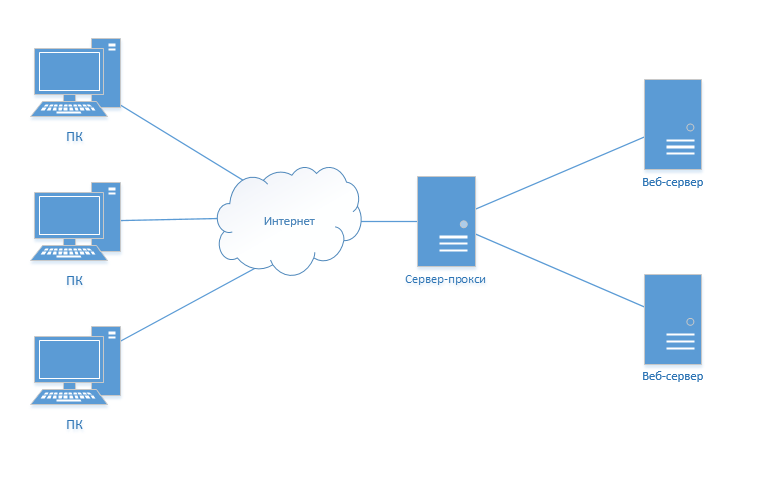


Рисунок 1.7 – схема работы обратного прокси

Что касается преимуществ, то обратные прокси-серверы очень усложняют злоумышленникам атаки на внутренние серверы. Кроме того,   
он работает как балансировщик нагрузки, чтобы распределить данные между внутренними серверами, что предотвращает перегрузку многочиселнными запросами на подключение. Прокси данного типа используются провайдерами веб-сервисов.

В настоящее время существует много типов прокси, некоторые из них являются базовыми.

Прокси-сервер HTTP находится между веб-сервером (HTTP-сервером)   
и веб-клиентом. Он используется для обработки протокола HTTP и проверки любого потенциально опасного содержимого перед его отправкой веб-клиенту. Прокси-сервер также действует как буфер между веб-сервером и потенциально опасными веб-клиентами. Это делается путем обеспечения соответствия HTTP RFC и предотвращения потенциальных атак переполнения буфера. HTTP-прокси специально создан для HTTP-соединений и работает по той же модели клиент-сервер.

HTTPS также используется для передачи данных по сети. Разница между прокси-серверами HTTP и HTTPS заключается в том, что прокси-серверы HTTPS шифруют соединение между пользователем и прокси-сервером.

Основным преимуществом использования HTTPS-прокси является защита пользователей от атак типа «человек посередине». Хотя HTTP не имеет встроенных функций безопасности, HTTPS обеспечивает шифрование для защиты связи между браузерами и серверами. Это означает, что все данные, передаваемые между браузером и сервером, зашифрованы, что затрудняет перехват информации хакерами.

SOCKS прокси-серверы работают по сетевому коммуникационному протоколу SOCKS (сокр. SOCKet Secure). Этот протокол позволяет клиенту, находящемуся за брандмауэром, установить TCP соединение с целевым сервером. Таким образом, обмен пакетами данных между сервером и клиентом будет более стабильным.

SOCKS предназначен для маршрутизации любого типа трафика, генерируемого любым протоколом или программой. Таким образом, SOCKS довольно универсален, что делает его идеальным решением для таких процессов, как потоковая передача и одноранговый обмен.

Протокол SOCKS разработан таким образом, что его прокси-сервер не читает пакеты данных, которыми обмениваются пользователь и сервер, он просто отправляет и получает данные. Это сильная сторона, а также ахиллесова пята прокси-серверов SOCKS.

Прокси-сервер SOCKS эволюционировал от оригинального SOCKS до SOCKS4 и самого последнего усовершенствования, SOCKS5.

## 1.3 Задачи фильтрации сетевого трафика

Веб-фильтрация — это способ предостеречь пользователей от определенных веб-сайтов и URL-адресов, не позволяя их интернет-браузерам загружать определенные страницы с этих веб-сайтов. Они могут быть адаптированы по-разному для разных организаций, в зависимости от потребностей каждого сайта. Это означает дополнительную безопасность в сети, позволяющую пользователям пользоваться чистым и безопасным Интернетом.

Одной из очень важных функций устройства веб-фильтрации является проверка TLS/SSL-сертификатов. Эта функция расшифровывает содержимое зашифрованных веб-сайтов, сканирует содержимое на наличие вирусов и обеспечивает его соответствие допустимым политикам пользователя перед повторным шифрованием веб-сайта для доставки пользователю сети.

Поскольку более половины самых посещаемых веб-сайтов в мире зашифрованы, проверка TLS/SSL является жизненно важным процессом. Однако это может истощить ресурсы ЦП и привести к снижению производительности сети, низкой скорости Интернета и временной недоступности веб-утилит, таких как электронная почта.

Некоторые программные решения и все облачные решения учитывают потребление ресурсов ЦП. С облачными приложениями веб-фильтрации проверка TLS/SSL выполняется на серверах поставщика услуг, высвобождая ресурсы ЦП и повышая производительность сети.

В связи с ростом киберпреступлений и различного рода интернет-мошенничеств, становится все более важным защитить пользователей от злоумышленников в Интернете. Интернет-мошенники часто встречаются в чатах, на сайтах социальных сетей и в других местах. Интернет-фильтр может заблокировать все небезопасные ресурсы интернета и загрузки потенциально вредоносных файлов или программ. Кроме того, при необходимости, есть возможность ограничить время посещения, либо полностью заблокировать заданные веб-ресурсы.

## 1.4 Существующие средства фильтрации сетевого трафика

В настоящее время существует неограниченное количество фаерволов, программного обеспечение прокси и других средств фильтрации контента.

Наиболее популярные средства защиты:

* Kerio winroute firewall;
* TrustAccess;
* Fortinet FortiGate;
* Cisco Firepower NGFW;
* Squid.

### 1.4.1 Kerio winroute firewall

Kerio Control — это сертифицированное ICSA Labs комплексное решение для управления угрозами, обеспечивающее всестороннюю защиту сети. Kerio winroute firewall предоставляет возможности брандмауэра следующего поколения, включая сетевой брандмауэр и маршрутизатор, систему предотвращения вторжений, антивирусный шлюз, VPN, балансировку нагрузки на канал, создание отчетов, а также фильтрацию веб-контента и приложений.

Продукт представляет собой не просто SSL VPN, а полноценный пакет брандмауэра, разработанный для Microsoft Windows. Функции включают брандмауэр с проверкой пакетов с отслеживанием состояния, антивирусный шлюз и фильтрацию веб-контента. Доступ к VPN можно получить либо с помощью клиентского приложения, либо без клиента через веб-браузер.

Этот инструмент можно установить непосредственно на машину под управлением Windows 2000 или более поздней версии с двумя сетевыми адаптерами. Один сетевой адаптер подключен к сети, а другой к Интернету. WinRoute станет шлюзом после установки.

Брандмауэр WinRoute предназначен для прямого подключения к среде Windows. Этот продукт может взаимодействовать с Active Directory и легко интегрироваться в существующую инфраструктуру. Сама VPN может быть настроена как VPN типа «сеть-сеть, сервер-сервер», чтобы связать филиалы с ресурсами головного офиса, или ее можно использовать как VPN «клиент-сервер», чтобы пользователи могли получать доступ к ресурсам удаленно. Однако, здесь надо указать один из минусов данного программного обеспечения, ограниченная функциональность VPN.

### 1.4.2 TrustAccess

Средство защиты информации TrustAccess состоит из системы распределенных межсетевых экранов с центральным управлением. Что не мало важно, так это наличие сертификата ФСТЭК России по уровню МЭ 2 и НДВ 4, что и влечет за собой возможность строить защиту для ИСПДн до класса К1 включительно. Возможные сетевые атаки, от которых TrustAccess гарантирует защиту:

* Man in the Middle;
* Подмена защищаемого объекта;
* Replay-атака;
* IP-спуффинг;
* Перехват сетевых пакетов;
* Подмена сетевых пакетов;
* Отказ в обслуживании и т.д.

Также из достоинств можно выделить возможность настройки разграничения доступа как для авторизованных пользователей, так и для узлов сети.

### 1.4.3 Fortinet FortiGate

Компания Fortinet была основана в 2000 году. Компания выпустила свой первый брандмауэр FortiGate в 2002 году и завершила IPO в 2009 году. Компания предлагает устройства сетевой безопасности для операторов связи, центров обработки данных, предприятий и распределенных офисов.

Брандмауэры нового поколения FortiGate обеспечивают высокую производительность, многоуровневую безопасность и глубокую видимость для сквозной защиты корпоративной сети. Его специализированные процессоры безопасности (SPU) обеспечивают масштабируемую производительность и низкую задержку. Службы безопасности от FortiGuard Labs предоставляют обновления информации об угрозах и автоматические меры по их устранению. Брандмауэры интегрируются с другими продуктами безопасности Fortinet для сети, конечных точек, приложений, центров обработки данных, облака и уровня доступа, а также с решениями сторонних производителей. Они работают на операционной системе FortiOS.

Из недостатков данного продукта можно выделить необходимость в наличии необходимых знаний, прохождении обучения для управления соответствующими брандмауэрами Компании.

### 1.4.4 Cisco Firepower NGFW

Cisco ASA с сервисами FirePOWER расширяет возможности межсетевых экранов нового поколения серии Cisco ASA 5500-X и устройств адаптивной защиты Cisco ASA 5585-X, обеспечивая непрерывный мониторинг и защиту. Этот продукт обеспечивает интегрированную защиту от угроз для всего спектра атак — до, во время и после атаки — за счет объединения возможностей безопасности брандмауэра Cisco ASA с лучшими в отрасли функциями защиты от угроз Sourcefire и расширенными функциями защиты от вредоносных программ в одном устройстве.

Возможности Cisco NGFW включают в себя брандмауэр с отслеживанием состояния, непрерывающую оперативную настройку с подключением по сети, трансляцию сетевых адресов , последовательный периферийный интерфейс , VPN , кластеризацию и высокую доступность. Брандмауэры Cisco следующего поколения также обеспечивают динамическую маршрутизацию, расширенную защиту от вредоносного ПО, фильтрацию URL-адресов и аналитику безопасности, индикацию компрометации и осведомленность о приложениях. Брандмауэры ASA также имеют встроенный механизм IPS на основе сигнатур, полную видимость стека и детальный контроль, возможность включения информации из-за пределов брандмауэра, расшифровку SSL для выявления нежелательных зашифрованных приложений и т. д.

При использовании данного продукта, есть свои проблемы с процессом настройки правил на самих оборудованиях, а именно нестабильную работу программного обеспечения. Также отметим, достаточно сложный технический процесс для настройки сети при расширении информационной системы предприятия, добавления дополнительных сетей и т.д.

### 1.4.5 Squid

Squid Proxy — это прокси-сервер на базе Unix, который может выполнять как кэширование, так и проксирование. Он может кэшировать любой веб-контент из точки хранения данных, ближайшей к пользователю или запрашивающей стороне. Он часто кэширует большие мультимедийные файлы и веб-страницы, чтобы уменьшить перегрузку полосы пропускания. Таким образом, это помогает уменьшить разрешение веб-сайта и время загрузки данных.

Прокси-серверы Squid используются контент-провайдерами и медиакомпаниями для балансировки нагрузки и управления всплесками трафика для популярных материалов, что в конечном итоге повышает удобство работы зрителей, которым нужны программы.

Squid доступен для использования бесплатно и без ограничений благодаря Стандартной общественной лицензии (GPL) Фонда свободного программного обеспечения (FSF). Squid изначально разрабатывался для систем на базе Unix, но с тех пор был перенесен на Windows.

У Squid Proxies нет ассортимента продуктов или функций лучших конкурентов, это простой частный или общий прокси, который славится простотой настройки.

**2 Методы фильтрации трафика прокcи-сервером squid**

## 2.1 Возможности Squid по управлению сетевыми соединениями

Прокси-сервер можно понимать как промежуточный сервер, который отвечает за перенаправление запросов от клиента к ресурсам в Интернете, обеспечивая различные функции, безопасность и конфиденциальность в зависимости от ваших потребностей или политики вашей компании.

Прокси-серверы действуют как брандмауэры или веб-фильтры, предоставляя общие сетевые соединения и кэшируя данные для ускорения общих запросов. Хороший прокси-сервер защитит пользователей и внутренние сети от нежелательных вещей из Интернета.

Возможности фильтрации Squid могут быть описаны как сложными, так и легкими способами настройки. Все зависит от требований к сети. Он может легко фильтровать на основе заданного списка URL-адресов. Это может быть список «только разрешенных» или, наоборот, «не разрешенных». Он также может обрабатывать SSL-трафик, сопоставление URL-адресов с регулярными выражениями, упорядоченную оценку правил, фильтрацию на основе порта или протокола и даже может применять определенные списки фильтрации на основе MAC-адреса или IP-адреса исходного трафика.

Squid позволяет интернет-провайдерам экономить на своей пропускной способности за счет кэширования контента. Кэшированный контент означает, что данные обслуживаются локально, и пользователи увидят это благодаря более высокой скорости загрузки часто используемого контента. К примеру, 100 пользователей отправляют один и тот же запрос на сервер Squid, Squid извлекает данные на сервер только один раз и кэширует эти данные, а затем обслуживает 100 пользователей (рисунок 2.1).

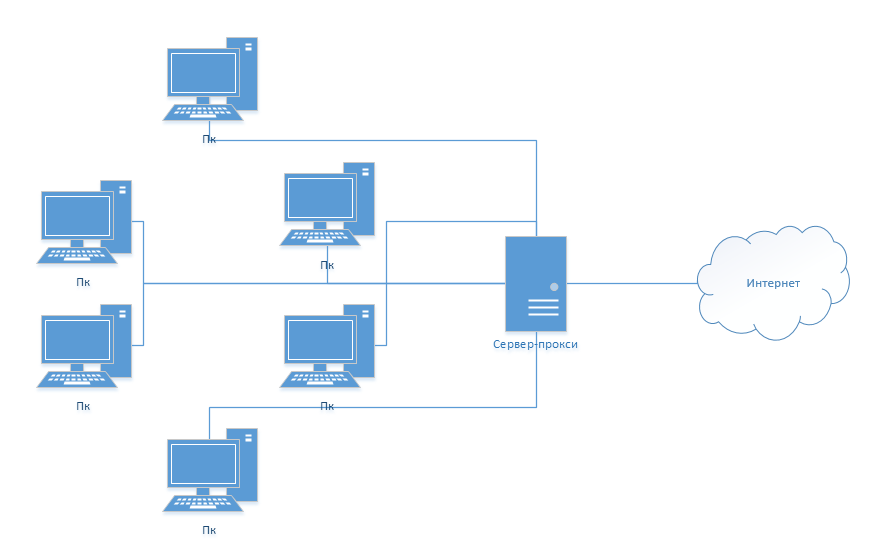


Рисунок 2.1 – схема прокси-сервера

Также есть возможность ограничить пропускную способность. Squid был выпущен под лицензией GNU (General Public License) и находится в свободном доступе. Будучи прокси-сервером, он хранит часто используемые страницы в локальном кеше, чтобы сократить время отклика при загрузке этих веб-страниц и сохранить пропускную способность в процессе.

## 2.2 Понятие контентной фильтрации

Поскольку все соединения проходят через прокси, его реализация позволяет фильтровать их. Используя прокси, сетевые администраторы могут внести в черный список запрещенные веб-сайты или приложения; хотя это, и не основная цель Squid, он полезен для усиления контроля над сетевой активностью.

Фильтрация контента — это процесс, который управляет или блокирует доступ к определенным электронным письмам или веб-страницам. Цель состоит в том, чтобы заблокировать контент, содержащий вредоносную информацию. Дело в том, что в данном случае предстоит фильтровать не сами соединения, а их содержимое. Другими словами, фильтрация веб-контента не означает запрет на установку TCP-соединения клиента с внешними ресурсами по 80 порту, а подразумевает запрет на доступ к веб-ресурсам, содержащим потенциально вредоносный контент. Правда, содержимое веб-ресурса, можно проверить только при установке TCP-соединения клиента с сервером, когда веб-страница будет загружена непосредственно на хосте, инициирующем запрос. Здесь и возникает сложность в проверке содержимого веб-ресурса перед тем, как будет установлено TCP-соединения клиента с сервером.

Фильтрация контента может быть реализована аппаратно или программно и часто встроена в интернет-брандмауэры для блокировки доступа к информации. Компании используют инструменты фильтрации контента для повышения безопасности и применения корпоративных политик в отношении управления информационными системами, например, при фильтрации сайтов социальных сетей.

Обычно в этом сценарии прокси-устройство имеет два сетевых интерфейса. Сетевая карта используется для локальной сети, а вторая используется для доступа в Интернет. Все запросы на подключение, запрашиваемые локальными сетевыми устройствами, проходят через внутреннюю сетевую карту и перенаправляются на внешнюю сетевую карту, если трафик разрешен.

Из-за широкой области применения существуют различные типы контент-фильтров, большинство из которых можно настроить для удовлетворения потребностей конкретных пользователей.

### 2.2.1 Фильтрация на стороне сервера

Фильтры содержимого на стороне сервера управляют веб-трафиком для каждого пользователя определенной сети. Поставщик или администратор установит серверные фильтры содержимого на центральном сервере, который связан со всеми компьютерами в сети. Такие решения для фильтрации контента полезны для мониторинга использования Интернета в крупных организациях.

Поскольку эти решения основаны на серверах, они позволяют администраторам создавать один набор правил для фильтрации контента и применять его ко всем пользователям в сети. В отличие от фильтрации на основе интернет-провайдера, полный контроль остается за компанией, что делает настройку динамичной и простой.

Назначение разных уровней доступа разным классам пользователей становится очень простым в фильтрах содержимого на стороне сервера. Многие поставщики решений предлагают готовые фильтры контента, которые требуют лишь незначительных настроек на организационном уровне.

### 2.2.2 Фильтрация на стороне клиента

Фильтр содержимого на стороне клиента устанавливается в системе конечного пользователя. Однако конечный пользователь может не иметь возможности изменить или удалить его. Это ограничение обычно достигается путем защиты программы паролем или ограничения возможностей системы конечного пользователя с помощью прав администратора.

Ограничивая работу клиентского фильтра, администратор может быть уверен, что клиент не сможет обойти его ограничения. Клиентские фильтры хорошо подходят для домашнего использования.

Они также подходят для бизнес-сред, требующих фильтрации для определенных компьютеров. Однако управление фильтрами содержимого на стороне клиента становится утомительным (а в некоторых случаях почти невозможным) по мере увеличения размера организации.

### 2.2.3 Фильтрация на уровне провайдера

Для масштабной фильтрации контента поставщик интернет-услуг может ограничить веб-трафик на основе правил, установленных организацией или правительством. Когда интернет-провайдер реализует фильтрацию контента, эти ограничения распространяются на каждого пользователя.

Интернет-провайдер может фильтровать нежелательный веб-трафик, а также отслеживать чат и электронную почту. Как и другие типы фильтров контента, интернет-провайдеры также могут динамически блокировать или разблокировать услуги по мере обновления правил. В соответствии с действующей политикой поставщика услуг интернет-провайдеры могут разрешать клиентам выбирать типы контента для фильтрации для определенных классов пользователей.

Однако следует отметить, что делегирование функции фильтрации контента поставщику услуг Интернета устраняет прямой контроль и усложняет процесс модификации в случае необходимости внесения изменений.

### 2.2.4 Поисковая фильтрация

Поскольку удаленная работа становится нормой во время пандемии COVID-19, такие решения, как Google Workspace и Microsoft 365 для бизнеса, позволяют организациям контролировать действия пользователей без необходимости использования VPN или клиентского программного обеспечения. Взяв под контроль веб-трафик сотрудников, работодатели могут удаленно внедрять дополнительные фильтры безопасности.

Когда эти фильтры включены, поисковые системы будут отображать только те результаты, которые соответствуют политике компании. Кроме того, может быть заблокировано неприемлемое содержимое, которое не заблокировано организацией явно, но считается вредоносным поставщиком решения.

Однако стоит отметить, что фильтрация поисковыми системами — это не то же самое, что фильтрация URL-адресов. Хотя между ними может быть определенный уровень совпадения, сотрудники, которые уже знают URL-адрес веб-сайта, который специально не заблокирован, могут по-прежнему иметь доступ к нему без использования поисковой системы.

На домашнем уровне некоторые поисковые системы, такие как Kiddle и Wacky Safe, предлагают фильтры, ориентированные на детей. Результаты поиска этих поставщиков предварительно фильтруются, чтобы гарантировать, что они подходят для детей.

Такие фильтры контента полезны, поскольку они позволяют детям и другим иждивенцам просматривать Интернет без присмотра, гарантируя, что они будут держаться подальше от веб-сайтов, которые, как известно, содержат небезопасный или неприемлемый контент.

## 2.3 Проблема фильтрации шифрованного трафика

В глобальной сети Интернет данные передаются по протоколу прикладного уровня передачи данных HTTP. Однако сетевой трафик передается в незашифрованном виде, что не безопасно. Поэтому стали применять криптографический протокол SSL/TLS, который подразумевает более безопасную связь. Благодаря протоколам шифрования и появилась аббревиатура HTTPS.

SSL (и его преемник, TLS) — это протокол, который работает непосредственно поверх TCP (хотя существуют также реализации для протоколов на основе дейтаграмм, таких как UDP). Таким образом, протоколы более высоких уровней (например, HTTP) можно оставить без изменений, но при этом обеспечить безопасное соединение. Под слоем SSL HTTP идентичен HTTPS.

Что касается метода шифрования, SSL-сертификат шифрует данные с использованием метода асимметричного шифрования. Асимметричное шифрование включает в себя два разных, но математически связанных ключа. Один из этих ключей называется открытым ключом, а другой называется закрытым ключом.

Открытый ключ, как следует из названия, общедоступен и используется для шифрования данных. Закрытый ключ, с другой стороны, держится в секрете и хранится на веб-сервере. Он используется для расшифровки данных.

Процесс установления безопасного соединения называется рукопожатием SSL/TLS (рисунок 2.2). Это виртуальное рукопожатие происходит между двумя сторонами, участвующими в передаче данных: клиентом и сервером.

Процесс рукопожатия SSL/TLS состоит из следующих пунктов:

1. Приветствие клиента: клиент инициирует рукопожатие и отправляет сообщение «Hello» на сервер. Сообщение будет включать поддерживаемую версию TLS, наборы шифров и строку случайных байтов, известную как случайный клиент.
2. Приветствие сервера: сервер отвечает на инициацию клиента, отправляя сообщение, которое состоит из сертификата SSL/TLS, поддерживаемых наборов шифров и случайного выбора сервера.
3. Аутентификация и предварительный мастер-ключ: клиент аутентифицирует сертификат сервера и создает предварительный мастер-ключ для сеанса, шифрует с помощью открытого ключа сервера и отправляет зашифрованный предварительный мастер-ключ на сервер.
4. Расшифровка и главный секрет: сервер расшифровывает предварительный главный ключ, используя свой закрытый ключ, а затем и сервер, и клиент выполняют определенные шаги для создания главного секрета с согласованным шифром.
5. Шифрование с помощью сеансового ключа: и клиент, и сервер обмениваются шифрованием и дешифрованием информации с использованием общего ключа. Этот ключ называется сеансовым ключом, а этот метод шифрования называется симметричным шифрованием.



Рисунок 2.2 – Схема выполнения процедуры «рукопожатия» в протоколе TLS

При правильном использовании SSL/TLS все, что злоумышленник может видеть в кабеле, — это IP-адрес и порт, к которому подключен узел, примерное количество данных, которое отправляется, а также используемое шифрование и сжатие. Он также может разорвать соединение, но обе стороны будут знать, что соединение было прервано третьей стороной.

В интересах каждого, как клиента, так и внешнего веб-сайта иметь безопасное соединения. Для того, чтобы подтвердить личность клиента и убедиться, что это не злоумышленник, на узле клиента должен быть открытый ключ сервера. Однако, хранить все ключи со всех внешних веб-сайтов что вообще существуют, не очень практично, так как база данных будет огромной, а обновления должны запускаться каждый час.

Решением данной проблемы являются центры сертификации (ЦС). Когда на любой компьютер устанавливается операционная система или любой браузер, то появился список доверенных ЦС. Список сертификатов находится в свойствах браузера. В операционной системе Windows 10 необходимо открыть панель управления, далее открыть свойства браузера и там же перейти во вкладку «промежуточные центры сертификации» (рисунок 2.3):

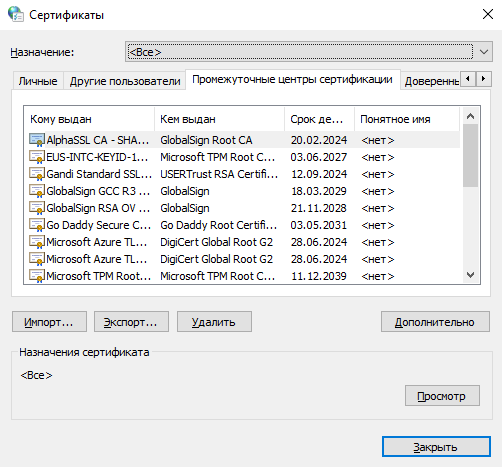


Рисунок 2.3 – Список сертификатов в ОС Windows

В операционной системе Linux дистрибутива Ubuntu корневые сертификаты ЦС хранятся в файле /etc/ssl/certs/ (рисунок 2.4).

Эти списки могут быть изменены, например, можно удалить или добавить сертификаты, а также есть возможность создать вручную собственный ЦС (однако в этом случае узел, на котором был создан и добавлен ЦС, будет является единственным, кто доверяет этому ЦС). В этом списке ЦС также хранится открытый ключ ЦС.

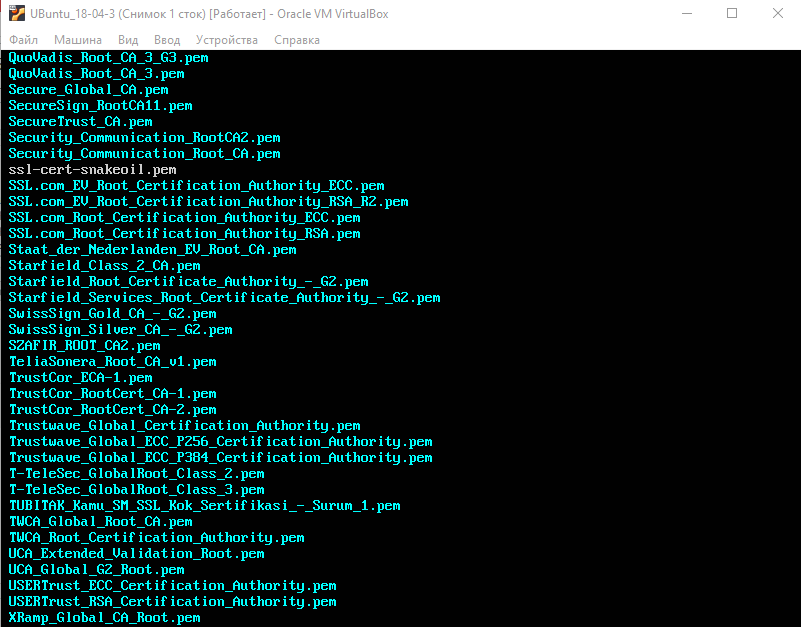


Рисунок 2.4 – Список сертификатов в ОС Windows

Например, при попытке соединения с серверами Google, они отправляют клиенту свой сертификат, который также упоминает, что подписан GeoTrust (GeoTrust — поставщик цифровых сертификатов.). Если клиент доверяет GeoTrust, можно проверить (используя открытый ключ GeoTrust), что GeoTrust действительно подписал сертификат сервера. Чтобы подписать сертификат самостоятельно, необходим закрытый ключ, который известен только GeoTrust. Таким образом, злоумышленник не сможет сам подписать сертификат и ошибочно объявить себя Google.com. Когда сертификат будет изменен хотя бы на один бит, знак будет неправильным, и клиент отклонит его.

Итак, если клиент знает открытый ключ, сервер сможет подтвердить его подлинность. Открытый ключ шифрует, а закрытый ключ расшифровывает. Зашифровав сообщение с помощью открытого ключа сервера, затем отправив его, сервер пытается повторить исходное сообщение. Если у сервера получилось повторить исходное сообщение, отправленное клиентом, то он просто докажет, что получил закрытый ключ, не раскрывая ключ.

Важно доверять открытому ключу: любой может сгенерировать пару закрытый/открытый ключ, в том числе и злоумышленник.

Если один из ЦС, которому доверяет клиент, скомпрометирован, злоумышленник может использовать украденный закрытый ключ, чтобы подписать сертификат для любого веб-сайта, который ему нравится. Когда злоумышленник может отправить клиенту поддельный сертификат, подписанный им самим закрытым ключом от ЦС (этому ЦС доверяет клиент), инициатор соединения с сервером не знает, что открытый ключ является поддельным, подписанным украденным закрытым ключом.

Здесь есть важная деталь: клиент должен доверять ЦС, чтобы не создавать сертификаты при любых попытках соединения с внешними серверами или при любых других случаях. Когда такие организации, как Microsoft, Apple и Mozilla, доверяют центру сертификации, центр сертификации должен проводить проверки; другая организация периодически проверяет их, чтобы убедиться, что все работает в соответствии с правилами. Выдача сертификата осуществляется тогда и только тогда, когда владелец домена может доказать, что он владеет доменом, для которого выдан сертификат.

Когда клиент с сервером устанавливает HTTPS-соединение, то только им известно какими данными они обмениваются между собой. Другими словами, если трафик от клиента до сервера будет проходить через промежуточный сервер или какой-либо узел сети, то конфиденциальность данных при их передаче не нарушиться. Узел сети, через который будет проходить соединение, не сможет видеть какие данные, передаваемые по защищенному протоколу HTTPS, передаются между клиентом и сервером.

В случае с фильтрацией контента с помощью прокси-сервера, необходимо произвести его настройку таким образом, чтобы прокси Squid, находясь посредине между клиентом и сервером, мог расшифровывать и обратно зашифровывать данные проходящие мимо него.

## 2.4 Технологии анализа шифрованного трафика в прокси-сервер Squid

В первую очередь, для реализации работы фильтрации прокси-сервером, необходимо, чтобы squid, выполняя роль «человека по середине», расшифровывал трафик между источником и целевым сайтом самозвареяющим сертификатом для проверки. HTTPS-трафик зашифрован и использует SSL протокол (Secure Sockets Layer) для защиты передаваемой информации от посторонних лиц. Использование HTTPS способствует некой защите от расшифровки траффика Компании злоумышленниками, но не надо забывать о том, что злонамеренные действия могут совершать и сами сотрудники предприятия. Зашифрованный трафик может не только скрыть незаконную деятельность сотрудников, но и потенциально опасный контент. Для решения данной проблемы используется функция Squid SSL Bump.

“SSL Bumping” в Squid расшифровывает и записывает в лог-файл HTTPS-трафик. SSL Bump в свою очередь должен включать в себя набор настроек «Peek and Splice».

«Peek and Splice» - это набор новых действий SslBump и связанных функций. SslBump – режим работы squid, используемый для перехвата содержимого зашифрованных HTTPS-сеансов. Для работы SSL Bumping требуется SSL-сертификат и приватный ключ в формате PEM. Можно использовать самоподписанный SSL-сертификат или выданный центром сертификации.

Функция Peek and Splice просматривает приветственное сообщение TLS клиента и при имеющейся информацию SNI (Server Name Indication — расширение компьютерного протокола TLS, которое позволяет клиентам сообщать имя хоста, с которым он желает соединиться во время процесса «рукопожатия»), отправляет идентичное или аналогичное сообщение клиента серверу, а затем просматривает приветственное сообщение TLS сервера.

Этапы обработки.

Обработка squid\_bump проходит в несколько этапов «рукопожатий» TCP и TLS. Шаги просмотра дают squid больше информации о клиенте или сервере, но часто ограничивают действия, которые squid может выполнять в будущем. При этом поток данных между клиентом и сервером дублируется, где первый находится между клиентом и прокси, а второй - между прокси и сервером.

Этап 1 включает в себя:

* Получение информации уровня TCP от клиента.
* Выполнение последовательности проверок и корректировок (Callout Sequence) с указанным выше запросом на подключение.
* Оценка всех ssl\_bump правил и выполнение первого совпавшего действия (splice, bump, peek, stare, terminate).

Этап 1 - единственный этап обработки, который выполняется всегда.

Запрос на подключение CONNECT регистрируется в access.log. Следует отметить, что для перехваченного HTTPS-трафика на данном этапе недоступно «доменное имя». Запись в журнале будет содержать только IP-адрес и порт назначения.

Этап 2 включает в себя:

* Получение информации из TLS сообщения ClientHello клиента, включая SNI.
* Выполнение последовательности проверок и корректировок (Callout Sequence) с уже установленным запросом на подключение.
* Оценка всех ssl\_bump правил и выполнение первого совпавшего действия (splice, bump, peek, stare, terminate).

Шаг 2 выполняется только в том случае, если правило peek/stare совпало на предыдущем шаге.

Установленный запрос на подключение CONNECT регистрируется в access.log на этом шаге, если этот шаг является окончательным (т.е. шаг 3 отсутствует).

Этап 3:

* Получение информации из TLS сообщения ServerHello сервера, включая его сертификат.
* Проверка сертификата сервера.
* Оценка всех ssl\_bump правил и выполнение первого совпавшего действия (splice, bump, peek, stare, terminate) для соединения.

Шаг 3 выполняется только в том случае, если правило peek/stare совпало на предыдущем шаге.

В большинстве случаев единственный значимый выбор на шаге 3 - разорвать ли соединение. Данное решение выбирается на предыдущем шаге.

Набор действий, применяемых в ssl\_bump, приведен в таблице 2.1.

Таблица 2.1 – Действия ssl\_bump

| Действие | Применяется на шаге | Описание |
| --- | --- | --- |
| Peek | 1, 2 | Когда правило просмотра совпадает с шагом 1, squid переходит к шагу 2, где он анализирует сообщение TLS Client Hello и извлекает SNI (если есть). Когда на шаге 2 соблюдается правило peek, Squid переходит к шагу 3, где он анализирует TLS Server Hello и извлекает сертификат сервера, сохраняя при этом возможность объединения клиентских и серверных подключений; просмотр сертификата сервера обычно исключает возможность перехвата содержимого зашифрованных сенасов. |
| Splice | 1, 2, иногда 3 | Устанавливается TCP-туннель без расшифровки соединения. Клиент и сервер обмениваются данными, как будто между ними нет прокси. |
| Stare | 1, 2 | Когда правило stare совпадает на шаге 1, squid переходит к шагу 2, где он анализирует сообщение TLS Client Hello и извлекает SNI (если есть). Когда правило проверки совпадает во время шага 2, squid переходит к шагу 3, где он анализирует TLS Server Hello и извлекает сертификат сервера, сохраняя при этом возможность перехвата клиентских и серверных соединений; просмотр сертификата сервера обычно исключает возможность последующего соединения |
| Bump | 1, 2, иногда 3 | Устанавливается TLS соединение с сервером (используя клиентский SNI, если он есть) и клиентом (используя имитированный сертификат сервера). |
| Terminate | 1, 2, 3 | Разрывает соединение клиент-сервер |

**3 Практическая реализация контентной фильтрации трафика**

## 3.1 Програмно-аппаратная платформа для реализации системы фильтрации

Для создания модели сети с клиентом, где будет обычный пользователь ПК и прокси-сервером используем ПО для виртуализации VirtualBox.

В роли клиента будет выступать виртуальная машина под управлением ОС Windows 10.

При создании виртуальных машин необходимо задать настройки их сетевым адаптерам таким образом, чтобы они образовали внутреннюю виртуальную сеть, то есть могли передавать данные друг другу и не могли передавать их во вне, в том числе хостовой системе (рисунок 3.1).

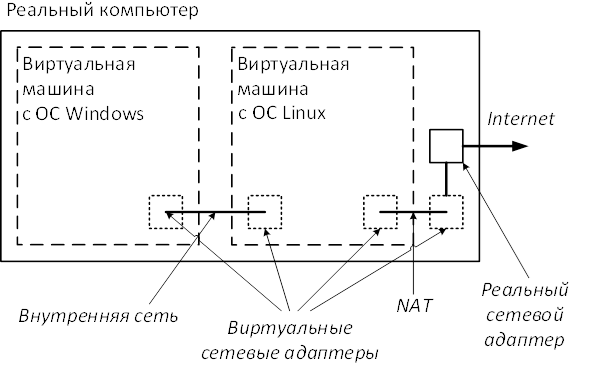


Рисунок 3.1 – схема соединения между виртуальными машинами и реальным компьютер компьютером

В роли прокси-сервера будет выступать виртуальная машина под управлением ОС Linux дистрибутива Ubuntu версии 18.04, версия ядра 4.15.0-20-generic. Ubuntu Linux — самая популярная операционная система с открытым исходным кодом. Как и в случае с Windows, установка Ubuntu Linux очень проста, и любой человек, обладающий базовыми знаниями о компьютерах, может настроить свою систему. По сравнению с Windows, для которой требуется антивирус, риски вредоносных программ, связанные с Ubuntu Linux, незначительны. Также стоит отметить, низкие системыные требования. Рекомендуемые аппаратные требования: процессор 700 МГц, 512 МБ ОЗУ и 5 ГБ на жестком диске.

При создании виртуальных машин необходимо задать настройки их сетевым адаптерам таким образом, чтобы они образовали внутреннюю виртуальную сеть, то есть могли передавать данные друг другу и не могли передавать их во вне, в том числе хостовой системе. Для этого в VirtualBox используется тип подключения «Внутренняя сеть» (рисунок 3.2).

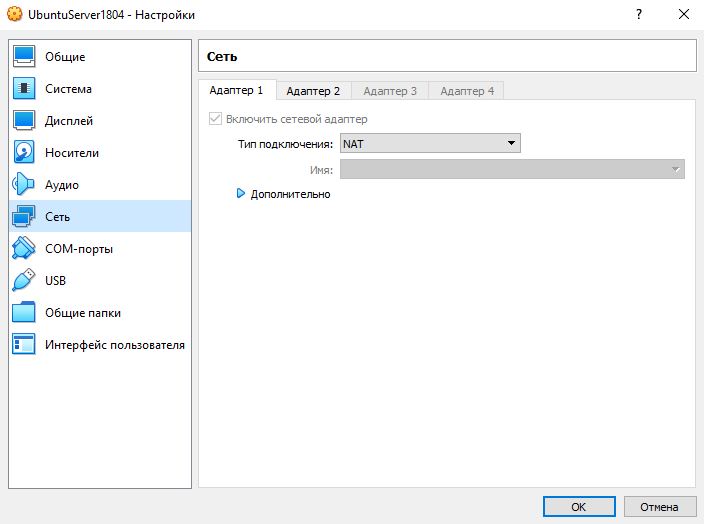


Рисунок 3.2 – настройки сетевых адаптеров для прокси-сервера

После задания данной настройки на обоих виртуальных машинах производится дополнительная настройка сетевых интерфейсов.

На виртуальной машине Windows задается сетевому интерфейсу в качестве шлюза по умолчанию IP-адрес виртуальной машины с Linux (рисунок 3.3).

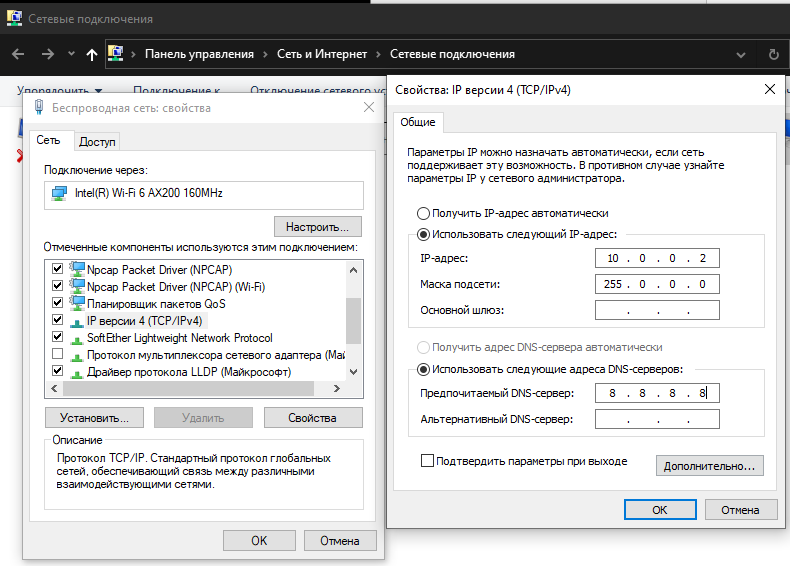


Рисунок 3.3 – сетевые настройки клиентского хоста под управлением ОС Windows

В дистрибутиве Ubuntu Server 18.04 настройки сетевых интерфейсов хранятся в папке /etc/netplan. Настройка производится путем внесения изменений в файл /etc/netplan 01-netcfg.yaml c помощью текстового редактора nano от имени суперпользователя. Всего у виртуальной машины Linux выделено 2 виртуальных сетевых интерфейса. Один сетевой интерфейс будет являться шлюзом по умолчанию для виртуальной машины Windows, которая находиться во внутренней сети, а второй интерфейс будет являться выходом в глобальную сеть Интернет.

В итоге содержимое файла должно быть следующим:

network:

ethernets:

enp0s3:

dhcp4: true

enp0s8:

addresses: [10.0.0.1/8]

Применяются настройки следующей командой:

sudo netplan apply

Но в ОС Ubuntu Linux не предустановлено ПО для выполнений функций прокси-сервера. Для этого необходимо произвести ряд настроек.

Прежде чем будет выполнятся настройка HTTPS squid, необходимо установить ряд пакетов, библиотек в Linux для сборки Squid:

sudo apt install openssl devscripts build-essential dpkg-dev libssl-dev

OpenSSL ­– это криптографическая библиотека общего назначения с помощью которой будет производится генерация сертификатов и другая работа с Secure Sockets Layer (SSL).

Devscripts включает в себя две оболочки (вышеупомянутую "debuild" и "debpkg"), которые призваны упростить процесс сборки Squid.

Пакет build-essential является справочником по всем пакетам, необходимым для компиляции Squid. Обычно он включает компиляторы и библиотеки GCC/g++ и некоторые другие утилиты.

dpkg-dev — это основной менеджер пакетов для Debian и систем на основе Debian, таких как Ubuntu. Инструмент устанавливает, создает, удаляет, настраивает и извлекает информацию для пакетов Debian. Команда работает с пакетами в формате .deb.

libssl-dev – является библиотека утилиты OpenSSL.

Версия Squid, которая находится в репозиториях Linux не поддерживает работу с SSL. Поэтому необходимо произвести ручную установку прокси Squid.

В Linux Ubuntu перед установкой ПО важным этапом является добавление репозиториев исходного кода в системные репозитории с исходным кодом следующей командой:

sudo apt build-dep squid

Репозитории исходного кода содержат метаданные, необходимые для определения зависимостей пакетов во время сборки.

В папке squid\_build будет находится все необходимые ресурсы для непосредственной сборки squid:

mkdir ~/squid\_build && cd ~/squid\_build

Следующей командой производится скачивание исходников squid:

sudo apt source squid

Затем необходимо выполнить редактирование файла debian/rules и добавить в него следующие флаги компиляции:

--enable-ssl

--enable-ssl-crtd

--with-openssl

Эти опции надо вставить в переменную BUILD\_CXX. Результат редактирования представлен на рисунке 3.4

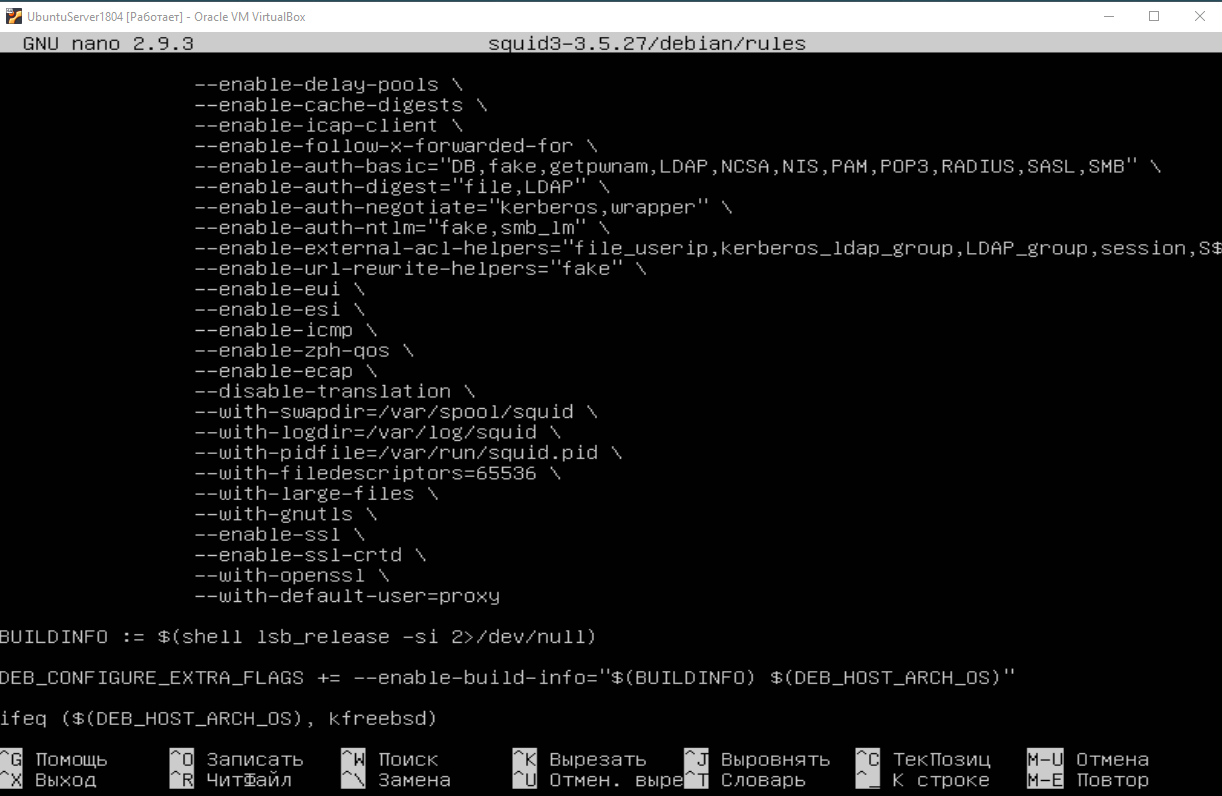


Рисунок 3.4 – содержимое файла debian/rules

Squid версии 2.5 и более поздние могут завершать соединения TLS или SSL. Поэтому необходимо добавить в файл debian/rules «--enable-ssl» . Squid-3.5 и более поздние версии автоматически определяют доступность библиотеки GnuTLS и включают функциональные возможности, если они доступны. OpenSSL должен быть включен явным образом с помощью параметра конфигурации «--with-openssl». Конфигурация «enable-ssl-crtd» необходима для работы Squid с сертификатами.

Заключительным этапом установки Squid будет произведение сборки и установки непосредственно самого прокси:

sudo debuild -d -uc -us

sudo dpkg -i ../squid\*.deb

Установленная версия Squid теперь поддерживает SSL (рисунок 3.5).

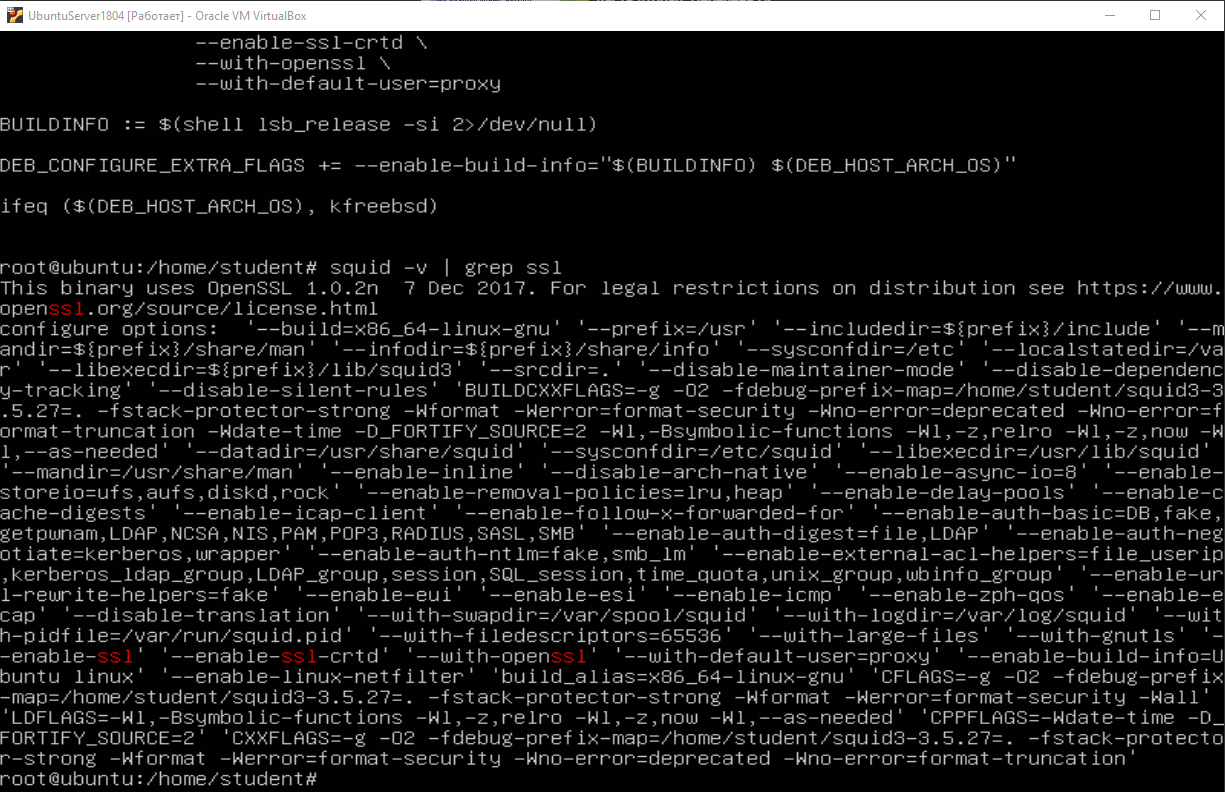


Рисунок 3.5 – результат ввода команды «squid –v | grep ssl»

## 3.2 Генерация сертификатов

Сертификат SSL/TLS — это лишь часть процесса шифрования данных, который работает на основе инфраструктуры открытого ключа (PKI) и шифрования с открытым ключом. Это цифровой файл, содержащий информацию, которая помогает вашему веб-серверу проверять и устанавливать безопасное зашифрованное соединение.

Криптографии ключ — это строка символов, используемая в алгоритме шифрования для изменения данных, чтобы они выглядели случайными. Подобно физическому ключу, он блокирует (шифрует) данные, так что только тот, у кого есть правильный ключ, может разблокировать (расшифровать) их.

Для этого каждый сертификат SSL/TLS содержит два асимметричных криптографических ключа — открытый ключ и закрытый ключ, — которые используются для создания третьего уникального симметричного ключа, известного как сеансовый ключ. Поскольку симметричное шифрование быстрее, чем асимметричное шифрование, оно делает связь между двумя сторонами быстрее и эффективнее. («Быстрее» — понятие относительное, когда речь идет о миллисекундах, но, тем не менее, это быстрее.)

Веб-сайту необходим SSL-сертификат, чтобы обеспечить безопасность пользовательских данных, подтвердить право собственности на веб-сайт, предотвратить создание злоумышленниками поддельной версии сайта

Чтобы SSL-сертификат был действительным, домены должны получить его от центра сертификации (ЦС). Технически любой может создать свой собственный SSL-сертификат, создав пару открытого и закрытого ключей. Такие сертификаты называются самозаверяющими сертификатами, потому что используемая цифровая подпись вместо ЦС будет собственным закрытым ключом веб-сайта.

Для создания сертификата SSL необходимы как файл закрытого ключа, так и запрос на подпись сертификата (CSR). Процесс начинается с запроса закрытого ключа от ЦС. Закрытые ключи — это криптографические ключи, сгенерированные с использованием таких алгоритмов, как RSA и ECC. Ключи RSA являются одной из старейших криптографических систем для этого процесса и поддерживаются многими платформами.

Алгоритмы шифрования RSA, DSA и ECC являются основными алгоритмами, используемыми для генерации ключей в инфраструктуре открытых ключей.

Инфраструктура открытых ключей (PKI) используется для управления идентификацией и безопасностью в интернет-коммуникациях и компьютерных сетях. Основной технологией, позволяющей использовать PKI, является криптография с открытым ключом, механизм шифрования, основанный на использовании двух связанных ключей: открытого ключа и закрытого ключа.

Эта пара открытого и закрытого ключей вместе шифрует и расшифровывает сообщения. Такое сопряжение двух криптографических ключей также известно как асимметричное шифрование, которое отличается от симметричного шифрования, при котором один ключ используется как для шифрования, так и для дешифрования.

Преимущество асимметричного шифрования заключается в том, что открытый ключ может быть опубликован для всеобщего обозрения, в то время как закрытый ключ хранится в безопасности на устройстве пользователя, что делает его намного более безопасным, чем симметричное шифрование.

Криптография с открытым ключом использует математические алгоритмы для генерации ключей. Открытый ключ состоит из строки случайных чисел, которые можно использовать для шифрования сообщения. Только предполагаемый получатель может расшифровать и прочитать это зашифрованное сообщение, и оно может быть расшифровано и прочитано только с использованием соответствующего закрытого ключа, который является секретным и известен только получателю.

Открытые ключи создаются с использованием сложного криптографического алгоритма, чтобы связать их со связанным с ними закрытым ключом, чтобы их нельзя было использовать с помощью атаки грубой силы.

Размер ключа или длина битов открытых ключей определяет силу защиты. Например, 2048-битные ключи RSA часто используются в сертификатах SSL, цифровых подписях и других цифровых сертификатах. Эта длина ключа обеспечивает достаточную криптографическую защиту, чтобы злоумышленники не взломали алгоритм. Организации по стандартизации, такие как CA/Browser Forum (Добровольный консорциум центров сертификации, поставщиков программного обеспечения для интернет-браузеров и защищенной электронной почты, операционных систем и других приложений с поддержкой PKI, который обнародует отраслевые руководящие принципы, регулирующие выдачу и управление цифровыми сертификатами X. 509 v. 3, которые привязаны к якорю доверия, встроенному в такие приложения. Его руководящие принципы охватывают сертификаты, используемые для протокола SSL/TLS и подписи кода, а также системную и сетевую безопасность центров сертификации.), определяют базовые требования к поддерживаемым размерам ключей.

Одним из наиболее распространенных вариантов использования PKI является TLS/SSL на основе X.509. Это основа протокола HTTPS, обеспечивающего безопасный просмотр веб-страниц. Но цифровые сертификаты также применяются в самых разных случаях, включая подписание кода приложений, цифровые подписи и другие аспекты цифровой идентификации и безопасности.

Алгоритм RSA был разработан в 1977 году Роном Ривестом, Ади Шамиром и Леонардом Адлеманом. Он основан на том факте, что для факторизации больших простых чисел требуется значительная вычислительная мощность, и это был первый алгоритм, использующий преимущества парадигмы открытого/закрытого ключа. Существуют различные длины ключей, связанные с RSA, при этом длина ключа RSA 2048 бит является стандартом для большинства веб-сайтов сегодня.

DSA использует другой алгоритм, чем RSA, для создания открытых/закрытых ключей, основанный на модульном возведении в степень и проблеме дискретного логарифма. Он обеспечивает тот же уровень безопасности, что и RSA, для ключей эквивалентного размера. DSA был предложен Национальным институтом стандартов и технологий (NIST) в 1991 году и принят Федеральным стандартом обработки информации (FIPS) в 1993 году.

ECC основан на математических алгоритмах, управляющих алгебраической структурой эллиптических кривых над конечными полями. Он обеспечивает такие же уровни криптографической стойкости, как RSA и DSA, с более короткими ключами. ECC был самым последним разработанным методом шифрования из трех: Алгоритм цифровой подписи на основе эллиптических кривых (ECDSA) был аккредитован в 1999 году, а соглашение о ключах и передача ключей с использованием криптографии на основе эллиптических кривых последовали за ним в 2001 году. Как и DSA, ECC сертифицирован FIPS.

Для правильной работы прокси-сервера, на виртуальной машине Linux необходимо сгенерировать корневой сертификат собственного ЦС на основе которого будут подписываться сертификаты для сайтов. Сертификаты будут хранится в отдельной папке /etc/squid/ssl.

mkdir /etc/squid/ssl

cd /etc/squid/ssl

Нижеописанной командой создается файл закрытого ключа squid.key.

sudo openssl genrsa -out /etc/squid/ssl/squid.key

Содержание файла закрытого ключа squid.key представлено на рисунке 3.6.



Рисунок 3.6 – cодержание файла закрытого ключа squid.key

Закрытый ключ squid.key, будет использоваться для создания запроса на подпись сертификата (CSR), так как процесс установки TLS/SSL-соединения начинается с запроса закрытого ключа от ЦС на узле. Следующим шагом будет создание CSR-файла для самозаверяющего сертификата.

sudo openssl req -new -key /etc/squid/ssl/squid.key -out /etc/squid/ssl/squid.csr

Содержание squid.csr файла запроса на подпись сертификата представлено на рисунке 3.7.



Рисунок 3.7 – cодержание запроса на подпись сертификата файла squid.csr

Наконец, из файлов squid.key (файл закрытого ключа) и squid.csr создается корневой сертификат squid.pem собственного центра сертификации.

sudo openssl x509 -req -days 3650 -in /etc/squid/ssl/squid.csr -signkey /etc/squid/ssl/squid.key -out /etc/squid/ssl/squid.pem

Сертификат хранит техническую информацию, которая необходима для правильной работы.

На рисунках 3.8-3.9 указана структура сертификата файла squid.pem

В поле Version содержится информация о версии закодированного сертификата. Это говорит о том, что поле версии сертификата может иметь одно из трех значений: 0 означает версию v1, 1 означает версию v2 и 2 означает версию v3. По умолчанию это v1, что равно 0. Это объясняет, почему в выводе openssl есть 1 и 0x0, он показывает номер версии (1) и реальное значение поля (0) на рисунке 3.8.

Поле SerialNumber (рисунок 3.8) указывает на уникальное целое число, присвоенное сертификату. Это значение уникально для всех сертификатов, выпущенных одним и тем же ЦС. Это означает, что два сертификата из разных ЦС потенциально могут иметь один и тот же серийный номер.

Поле SignatureAlgorithm содержит идентификатор криптографического алгоритма, используемого ЦС для подписи сертификата.

Поле Issuer является уникальным идентификатором центра сертификации, выдавшего сертификат.

Поле Validity указывает временной интервал, в течение которого сертификат действителен.

Поле Subject Public Key Info используется для переноса открытого ключа и идентификации алгоритма, с которым используется ключ (например, RSA, DSA или Диффи-Хеллмана).

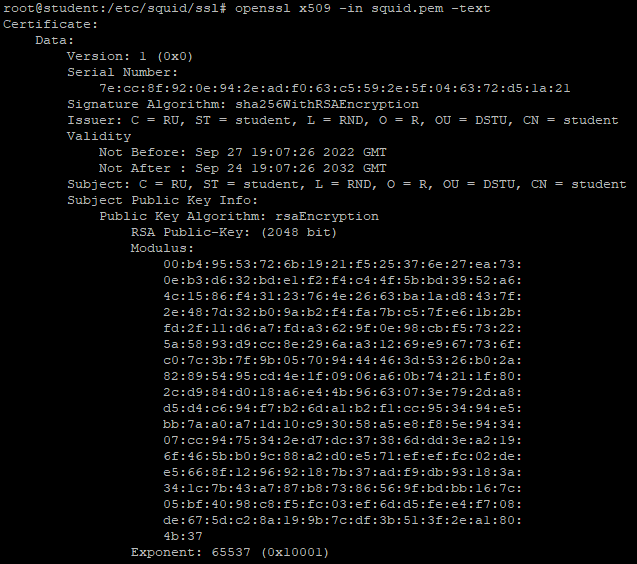


Рисунок 3.8 – структура сертификата файла squid.pem



Рисунок 3.9 – структура сертификата файла squid.pem

Примечательно, что сам сертификат определен между строками Begin Certificate и End Certificate. Строкой Begin Certificate обозначается начало сертификата, а строка End Certificate обозначает его конец. Так как сертификат с расширением .pem, то содержимое файла сертификата хранится в кодировке base64 (рисунок 3.9).

## 3.3 Правила фильтрации содержимого web-страниц

Для заключительных шагов к настройке прокси-сервера, в рамках фильтрации контента, необходимо отредактировать файл конфигурации /etc/squid/squid.conf. В Ubuntu Linux для редактирования данного файла необходимо выполнить команду:

sudo nano /etc/squid/squid.conf

Однако, изначально данный файл содержит очень много текста, как правило, закомментированных примеров параметров и пояснения к ним. Для того, чтобы извлечь из этого файла только активные параметры можно выполнить команды:

cp /etc/squid/squid.conf /etc/squid/backup.conf

cat /etc/squid/backup.conf | grep -v '^\(#\|$\)' > /home/student/squid.conf

cp /home/student/squid.conf /etc/squid/squid.conf

В полученном файле squid.conf будут некоторые стандартные настройки. Важной из них является

http\_port число intercept — порт, на котором прокси-сервер принимает соединения (обычно 3128). Этот порт понадобится для настройки клиентских браузеров. “intercept” указывает на то, что прокси будет прозрачным.

https\_port число intercept ssl-bump cert=/etc/squid/ssl/squid.pem ­– здесь помимо указания номера порта и и настройки прозрачности прокси, указывается директива ssl-bump и путь к сертификату.

ssl\_bump peek all

ssl\_bump splice all

sslcrtd\_program /usr/lib/squid/ssl\_crtd –s /var/lib/ssl\_db –M 4MB – данные строчки указывают на насртоку прокси-сервреа на расшифровку трафика. «sslcrtd\_program» данная программа будет генерировать SSL-сертификаты. Генерацией сертификатов будет заниматься не прокси сервер, а отдельный процесс «/usr/lib/squid/ssl\_crtd», что обеспечит скорость работы самого Squid.

Для того, чтобы изменения конфигурации Squid вступили в силу, используется команда

sudo squid -k reconfigure

Представленная выше команда не перезапускает процесс, а только применяет новые настройки. Если требуется полностью перехапустить прокси-сервер, можно использовать команду

sudo /etc/init.d/squid restart

Основной инструмент настройки Squid — это списки контроля доступа или ACL (Access Control List). ACL объявляются директивой, имеющей следующий синтаксис:

acl имя параметр элементы\_списка

Следующая важная директива http\_access, имеющая формат:

http\_access действие имя\_acl

определяет действие (разрешить или запретить) с элементами указанного ACL.

В начальной конфигурации Squid содержаться несколько строк со стандартными директивами. Изменять их не рекомендуется. Среди таких директив есть, например, следующие:

acl SSL\_ports port 443

acl Safe\_ports port 80 # http

acl Safe\_ports port 21 # ftp

acl Safe\_ports port 443 # https

Они определяют список портов, часто используемых стандартными приложениями, которые считаются безопасными.

Следующая стандартная директива запрещает доступ для всех небезопасных портов:

http\_access deny !Safe\_ports

Это равносильно разрешению доступа для всех безопасных портов.

В ACL можно указывать не только IP-адреса сетей или узлов, но и доменные имена сайтов. В качестве параметра указывается «domain», а в качестве элементов списка — URL сайтов. Если не поставить точку перед «domain», то правило будет распространяться только на доменное имя. При наличии точки оно также распространяется и на все поддомены данного домена.

Важной особенностью прокси-сервера Squid является возможность аутентификации пользователей. Её настройка включает в себя несколько этапов. Для начала необходимо установить в Linux Ubuntu утилиту для хеширования паролей следующей командой:

sudo apt-get install apache2-utils

Далее создается файл с паролями:

htpasswd -c /etc/squid/users имя\_пользователя

Ключ -с указывает, что необходимо создать новый файл с паролями. Если файл с таким именем уже существует, то он будет перезаписан. Для демонстрации работы будут созданы три пользователя admin, user\_vip, user.

Чтобы активировать функцию авторизации необходимо в файл конфигурации вписать следующие параметры:

auth\_param basic program /usr/lib/squid/basic\_ncsa\_auth /etc/squid/users

auth\_param basic children 5

auth\_param basic credentialsttl 8 hours

acl auth\_users proxy\_auth REQUIRED

Для пользователей, которым необходимы различные правила доступа, создаются отдельные ACL командой:

acl название proxy\_auth имя\_пользователя

После этого созданные ACL можно будет использовать в директиве http\_access.

Следующей функцией, имеющейся в Squid, является управление скоростью сетевых соединений. Для этого используется механизм Delay Pools (пулы задержек). Пулы задержек позволяют ограничить скорость получаемого прокси-сервером интернет-трафика в зависимости от разных условий, например в зависимости от того, какой пользователь запрашивает этот трафик, в какой подсети он находится, с какого сайта и т.д. В конфигурации Squid по умолчанию механизм пулов задержки выключен, поэтому при доступе к прокси-серверу между клиентами используется простая конкуренция, в результате которой некоторые пользователи могут занимать значительную часть пропускной способности интернет-канала, а некоторые практически не получать её. При этом, объекты, находящиеся в кэше, всегда отдаются клиенту с максимальной скоростью.

Для реализации ограничения скорости в squid используются следующие параметры конфигурационного файла: acl, http\_access, delay\_pools, delay\_access, delay\_parameters, delay\_class.

delay\_pools количество – эта команда задает количество delay pools.

delay\_class номер класс – эта команда определяет класс для каждого из delay\_pools. Существует несколько классов, у каждого класса существуют соответствующие ему настройки. Для каждого delay\_pools задается единственный класс.

delay\_access номер действие имя\_acl – здесь задается попадание (allow) или не попадание (deny) пользователей в заданный пул (номер\_пула).

Примечательно, что в данном случае слова allow и deny обозначают не действия (разрешить/запретить), а принадлежность пользователя к пулу задержек. При этом можно задавать не только пользователей, а любые сущности, доступные в acl, например адреса сетей, доменные имена сайтов и др. При этом, проверка принадлежности к данному классу происходит до первого совпадения. Если пользователь по какому-либо параметру совпадает с acl, и для него указано правило allow, то для него применяются параметры текущего пула, если задано правило deny, то на пользователя правила пула не применяются. Если запрос не попал ни в один delay pools то информация отправляется напрямую клиенту без задержек. Синтаксис использования данного параметра аналогичен http\_access.

delay\_parameters номер параметры – эта команда устанавливает параметры для указанного номера пула. При этом, в зависимости от класса заданного номера пула, параметры различаются. Для одного delay pools может использоваться только один delay\_parameters.

Существуют следующие классы delay pools:

— Класс 1 содержит ограничения для всех запросов, попадающих в пул.

— Класс 2 имеет ограничения: общие для всех (аналогично первому классу), плюс индивидуальные ограничения для отдельных хостов (биты 25 – 32 сетевого адреса IPv4).

— Класс 3 имеет ограничения: общие, индивидуальные для отдельных хостов (аналогично 2 классу). Плюс ограничения для сетей подсетей (биты 17 – 32 сетевого адреса IPv4).

— Класс 4 имеет ограничения: всё, что определено в классе 3, плюс добавлены ограничения для конкретного авторизованного пользователя.

Команда задания ограничений скорости для пула 4 класса выглядит следующим образом:

delay\_parameters pool total\_rest/total\_max net\_rest/net\_max host\_rest/host\_max user\_rest/user\_max

где:

* pool — номер пула, для которого определяются параметры;
* total — ширина канала на всех;
* net — ширина канала на подсеть;
* host — ширина канала на отдельный адрес;
* user — ширина канала на отдельного авторизованного пользователя;
* rest — скорость заполнения пула в байтах в секунду (значение –1 определяет отсутствие ограничений);
* max — объем пула в байтах.

В процессе обеспечения доступа в интернет прокси-сервер Sqiud может собирать и накапливать довольно информацию об устанавливаемых пользователями сетевых соединениях. Такая информация записывается в log-файл:

/var/log/squid/access.log

Однако, в том виде, в котором данные представлены в этом файле, очень сложно анализировать вручную. Поэтому существуют утилиты, автоматизирующие процесс анализа логов Squid и генерирующие на их основе удобную для воcприятия статистику.

Одной из таких утилит является sarg (Squid Analysis Report Generator). Несмотря на то, что sqrg является самостоятельной утилитой и способен генерировать отчёты в виде html-файлов без дополнительных программ, для удобного просмотра полученной статистки в графическом интерфейсе требуется наличие веб-сервера.

Для Linux существует множество программ, способных выполнять функции веб-сервера, однако для простого просмотра отчётов sarg достаточно будет использовать простой веб-сервер lighthttpd. Установка данного сервиса производится следующей командой:

sudo apt install lighttpd

Настройка sarg осуществляется также с помощью конфигурационного файла /etc/sarg/sarg.conf.

В нём необходимо задать перечисленные ниже параметры:

graphs yes

graph\_days\_bytes\_bar\_color orange

output\_dir /var/www/html/squid-reports

charset UTF-8

Результат правки в файле /etc/sarg/sqrg.conf представлены на рисунке 3.10.

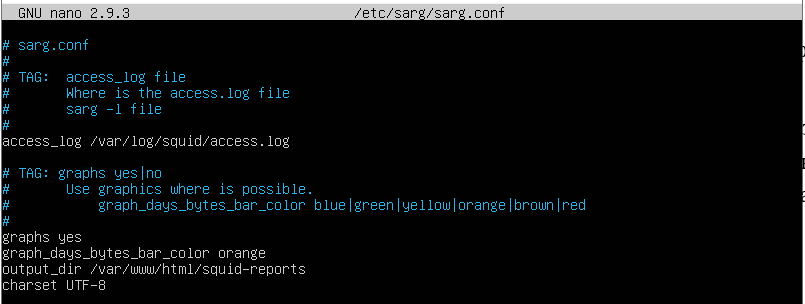


Рисунок 3.10 – содержимое файла sqrg.conf

После выполнения всех настроек необходимо запустить генерацию отчётов с помощью sarg командой:

sudo /usr/bin/sarg

Затем проверку сетевой статистики в sarg осуществляется посредством ввода в браузер IP-адрес узла (или виртуальной машины), на котором установлен sarg и squid.

В приложении А представлено содержимое файла конфигурации /etc/squid/squid.conf.

В рамках демонстрации возможностей работы squid были созданы три пользователя admin, user\_vip, user, где для пользователя user\_vip установим недоступность развлекательного контента. Для пользователя user помимо развлекательного контента запретом будут также социальные сети. Пользователю admin ограничений не выставлено. Также для наглядности установлена скорость соединения для разных пользователей, находящихся в разных сетях, а также к конкретным ресурсам.

## 3.4 Пример работы системы

Сеть между виртуальной машиной Windows и Ubuntu Linux остается неизменной (рисунок 3.1). Однако, в рамах дипломной работы для проверки работоспособности прокси-сервера, в настройках сетевых адаптеров двух виртуальных машин менялись настройки сетевой карты.

В ОС windows необходимо открыть «Сетевые подключения», далее в свойствах сетевого интерфейса во вкладке «Дополнительно» или «Доступ» установить галочку «Разрешить другим пользователям сети использовать подключение к интернету данного компьютера». Далее необходимо задать ip-адрес виртуальной машине и указать шлюз по умолчанию в командной строке windows:

netsh interface ip set address name=“Подключение по локальной сети” source=static addr=10.0.0.2 mask=255.0.0.0 gateway=10.0.0.1 gwmetric=1

Данной командой можно задавать любой ip-адрес виртуальной машине windows и шлюз по умолчанию. Но для полной настройки виртуальной машины windows, также необходимо в настройках браузера указать прокси сервер, через который будут проходить соединения и порт.

Заключетльным этапом будет проверка работоспособности прокси-сервреа squid.

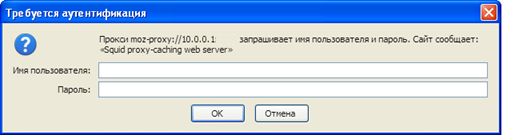


Рисунок 3.11 – окно аутентификации пользователей

На рисунке 3.11 представлено окно аутентификации, появляющееся при открытии браузера на узле под управлением виртуальной машины ОС Windows.

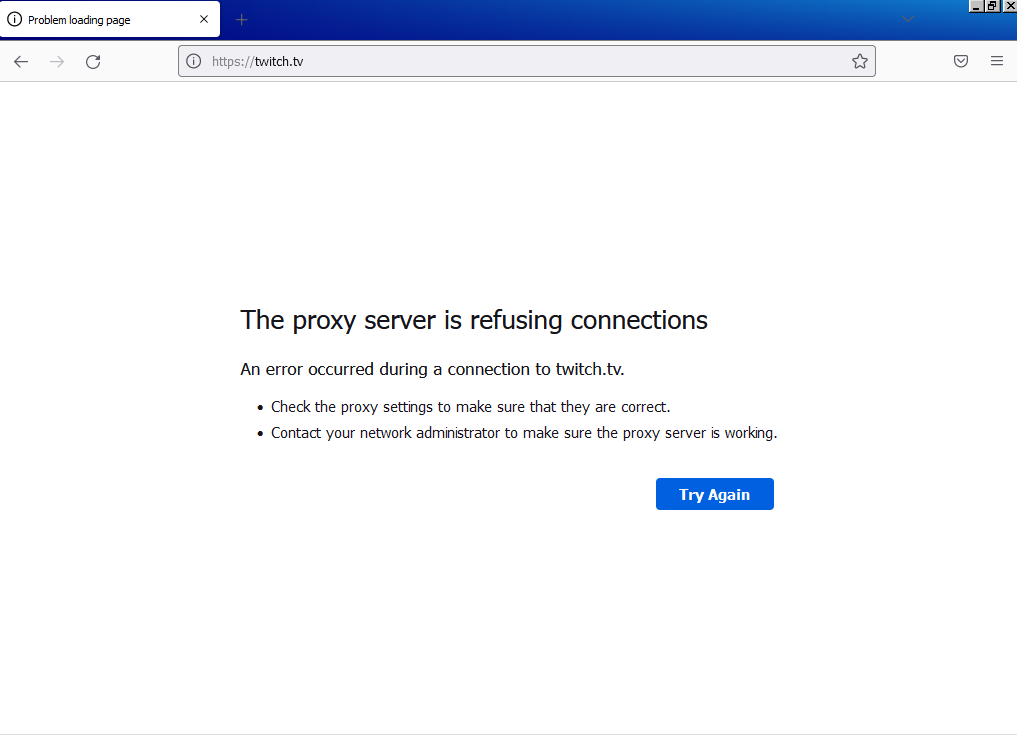


Рисунок 3.12 – недоступность развлекательного контента для пользователя user\_vip

На рисунке 3.12 представлен запрет на подключение к ресусру развлекательного контента для пользователя user\_vip. А на рисунках 3.13-3.14 показана недоступность как развлекательного контента, так и социальных сетей для пользователя user.

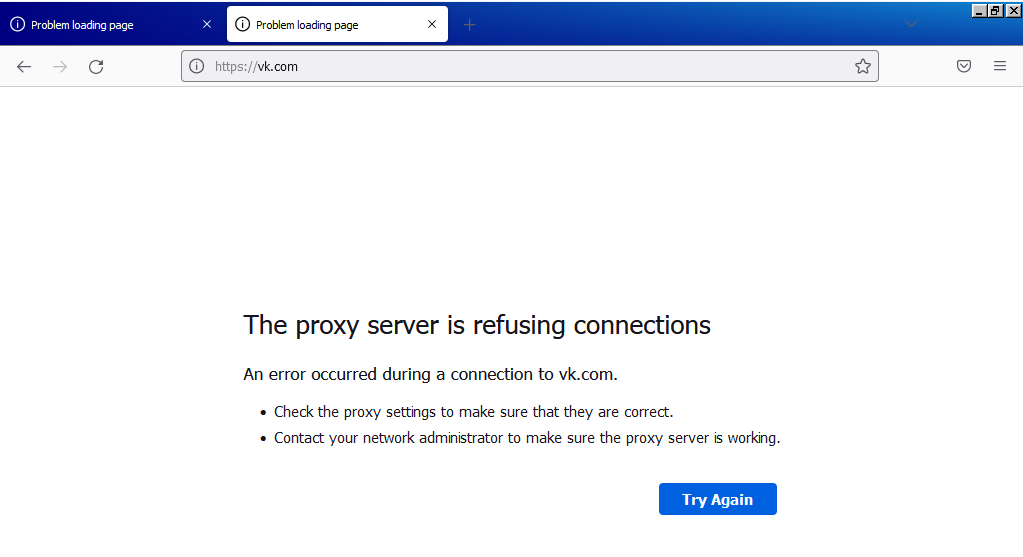


Рисунок 3.13 – недоступность ресурса социальных сетей для пользователя user

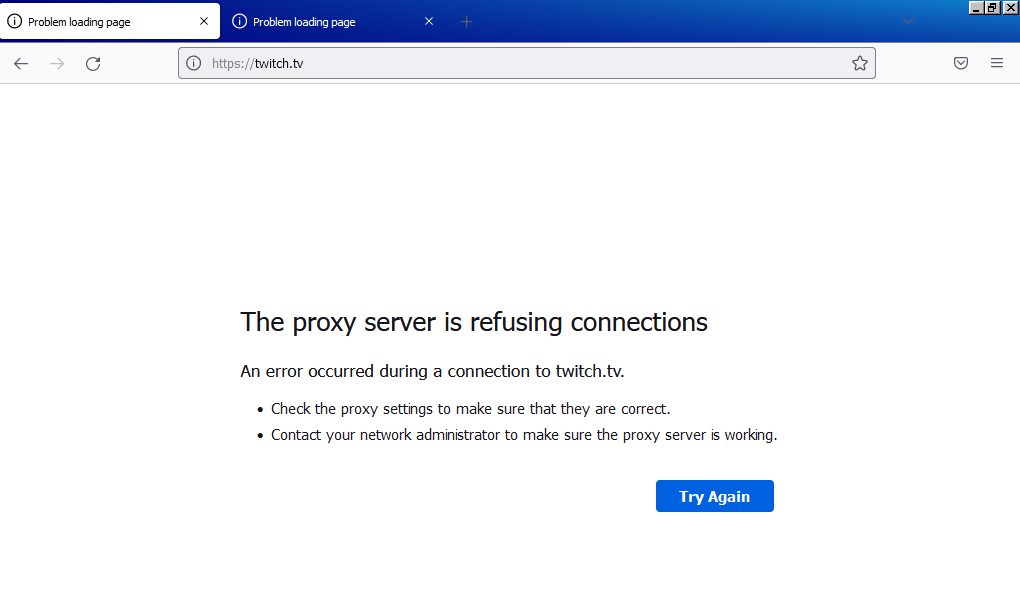


Рисунок 3.14 – недоступность развлекательного контента для пользователя user

На рисунках 3.15-3.16 показано ограничение скорости сетевого соединения для пользователя user и user\_vip соответственно.

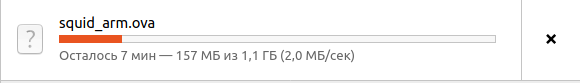


Рисунок 3.15 – результат настройки ограничения сетевого соединения 2Мб/с

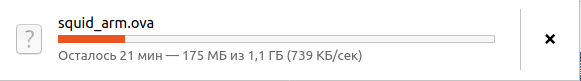


Рисунок 3.16 – результат настройки ограничения сетевого соединения 1Мб/с

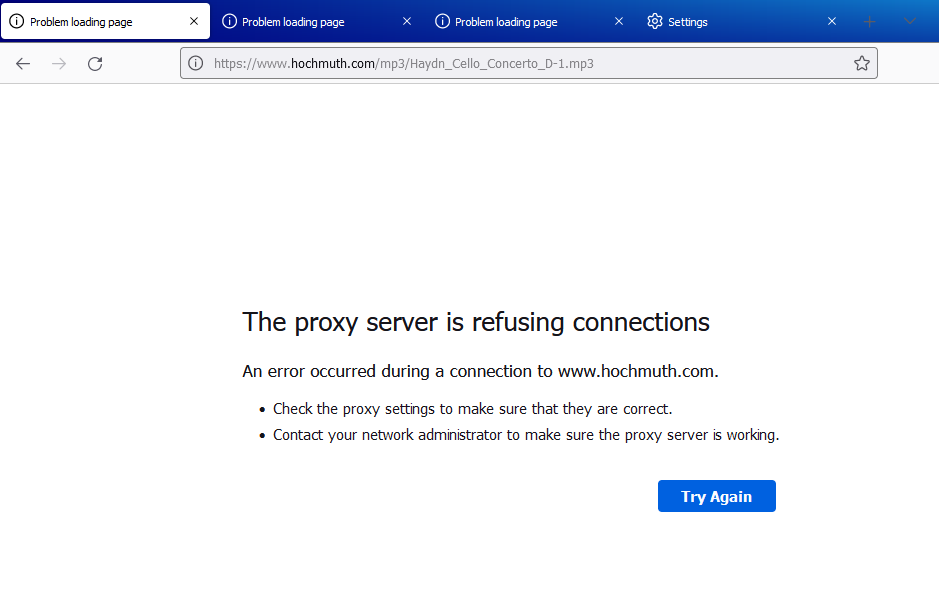


Рисунок 3.17 – результат запрета на скачивание mp3-файла

Также в конфигурационном файле указан запрет на скачивание mp3-файлов всем пользователям (рисунок 3.17).

На рисунках 3.18-3.21представлена сетевая статистика пользователей admin, user и user\_vip в сервисе sarg.

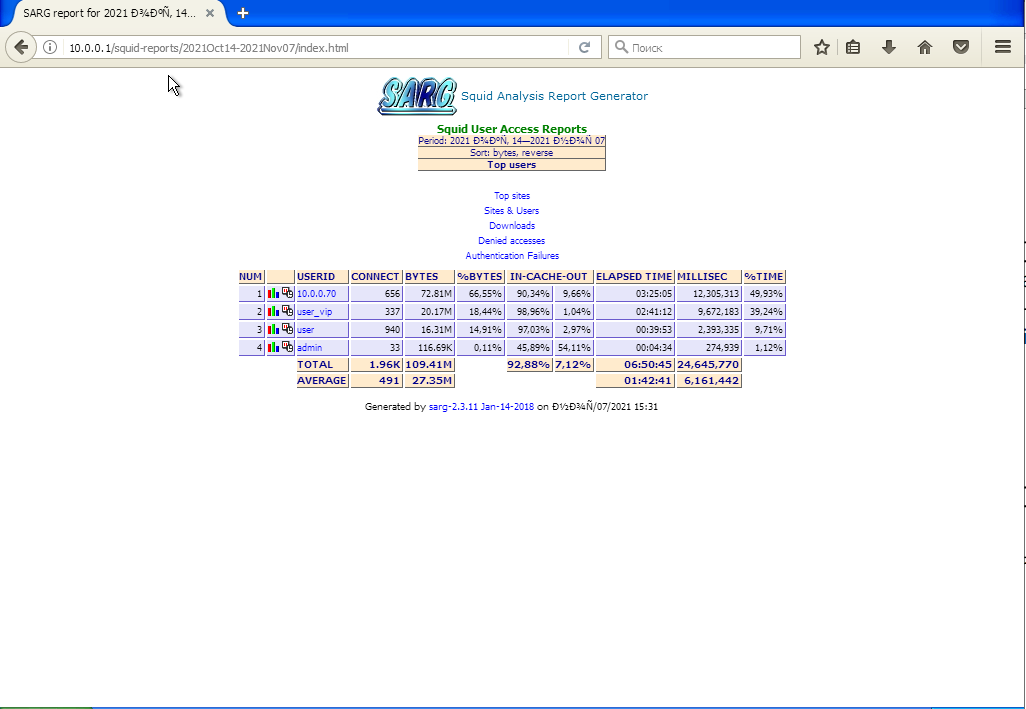


Рисунок 3.18 – главная страница Sarg

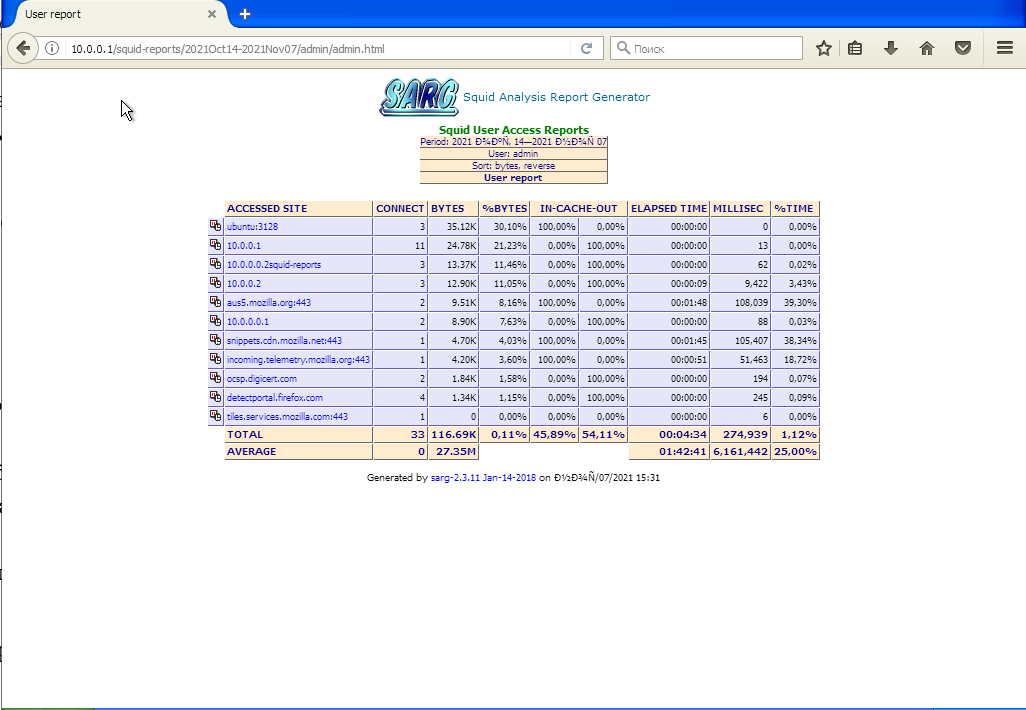


Рисунок 3.19 – страница статистики пользователя admin

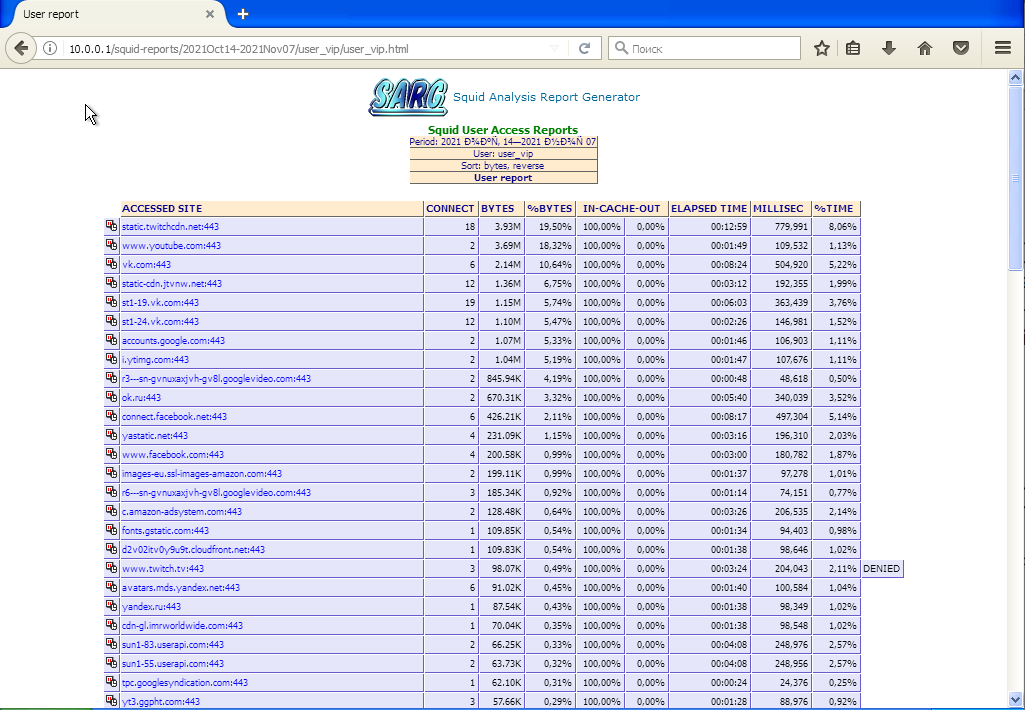


Рисунок 3.20 – страница статистики пользователя user\_vip

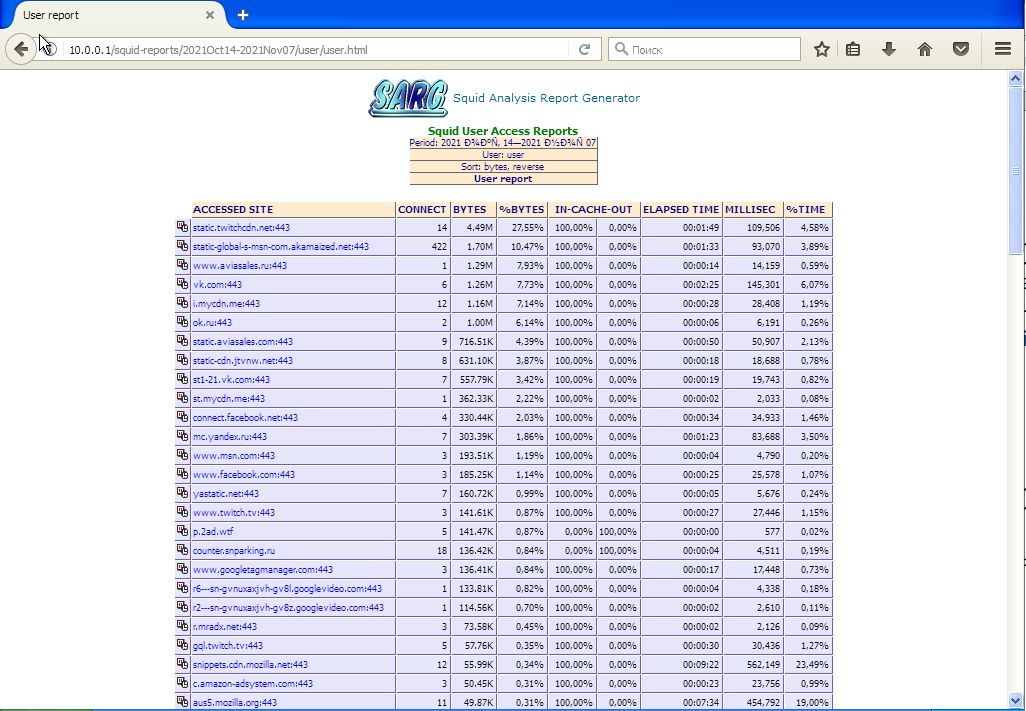


Рисунок 3.21 – страница статистики пользователя user

# 4 Безопасность и экологичность проекта

## 4.1 Эргономические основы оформления помещений

Проектирование рабочих мест, снабженных видеотерминалами, относится к числу важных проблем эргономического проектирования в области вычислительной техники.

Рабочее место с дисплеем должно обеспечивать оператору возможность удобного выполнения работ в положении сидя и не создавать перегрузки костно-мышечной системы. Основными элементами рабочего места являютcя: рабочий стол, рабочий стул (кресло), дисплей, клавиатура, мышь; вспомогательными – пюпитр, подставка для ног.

Конструкция и размеры рабочего стола должны обеспечивать возможность размещения на его рабочей поверхности необходимого комплекта оборудования и документов с учетом характера выполняемой работы.

Согласно ГОСТ 12.2.049-80 МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ, ОБОРУДОВАНИЕ ПРОИЗВОДСТВЕННОЕ, общим эргономическим требованиям, а также требованиям технического регламента Таможенного союза "О безопасности низковольтного оборудования" (ТР ТС 004/2011), утвержденный Решением Комиссии Таможенного союза от 16 августа 2011 года N 768 и ГОСТ Р 50948-2001 Средства отображения информации индивидуального пользования Общие эргономические требования и требования безопасности, стол может быть как регулируемым, так и нерегулируемым по высоте. Регулируемая высота рабочей поверхности должна изменяться в пределах от 680 до 800 мм, а нерегулируемого составлять 725 мм.

Размеры рабочей поверхности должны быть: глубина – не менее 600 (800) мм, ширина – не менее 1200 (1600) мм (в скобках указаны предпочтительные величины).

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм.

Рабочая поверхность стола не должна иметь острых углов и краев, а покрытие рабочей поверхности должно быть выполнено из диффузно отражающего материала с коэффициентом отражения 0,45 – 0,50.

Рабочий стул (кресло) должны обеспечивать поддержание физиологически рациональной рабочей позы оператора в процессе трудовой деятельности, создавать условия для изменения позы с целью снижения статического напряжения мышц шейно-плечевой области и спины, а также для исключения нарушения циркуляции крови в нижних конечностях.

Рабочий стул должен быть подъемно- поворотным и регулируемым по высоте, углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сидения.

В целях снижения статического напряжения мышц рук необходимо использовать стационарные или съемные подлокотники, регулирующиеся по высоте над сиденьем и внутреннему расстоянию между подлокотниками.

Поверхность сиденья должна иметь ширину и глубину не менее 400 мм, обеспечивать возможность изменения наклона от 15° вперед до 5° назад, и быть регулируемой по высоте в пределах от 400 до 550 мм…

Естественное и искусственное освещение в помещении регламентируется СП 52.13330.2016 Естественное и искусственное освещение (ред. 29.01.22). Актуализированная редакция СНиП 23-05-95 в зависимости от степени зрительного напряжения.

Искусственное освещение в рабочих помещениях осуществляется системой общего и местного равномерного освещения.

В производственных и административно-общественных помещениях, в случаях преимущественной работы с документами, разрешено применение системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).

В качестве источников света при искусственном освещении должны применяться преимущественно люминесцентные лампы (ЛЛ) типа ЛБ и ЛД.

Общее освещение в организации выполнено в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения работника при рядном расположении рабочих мест.

Расчет освещения приводится для комнаты площадью 15м2, ширина которой 5м, высота – 3 м; коэффициенты отражения потолка и стен *Р*П=70%; *Р*С=50%.

Для расчета количества светильников определим световой поток, падающий на поверхность по формуле:

,

где *F* – рассчитываемый световой поток, Лм;

*Е* – нормированная минимальная освещенность, лк (определяется по таблице). Работу программиста, в соответствии с этой таблицей, можно отнести к разряду точных работ, следовательно, минимальная освещенность будет

*Е* = 300лк;

*S* – площадь освещаемого помещения (в нашем случае *S* = 15м2);

*Z* – отношение средней освещенности к минимальной (обычно принимается равным 1,1…1,2 , примем Z = 1,1);

*К* – коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (его значение зависит от типа помещения и характера проводимых в нем работ и в нашем случае *К* = 1,5);

*n* – коэффициент использования светового потока, (выражается отношением светового потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и исчисляется в долях единицы; зависит от характеристик светильника, размеров помещения, окраски стен и потолка, характеризуемых коэффициентами отражения от стен (РС) и потолка (РП). Значение n определим по таблице коэффициентов использования различных светильников (табл. 4.1).

Таблица 4.1 – Коэффициент использования светового потока n,%

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Коэффици-ент  отражения | Марка светильника | | | | | | | | | | | | | | |
| НСП09 | | | ВЗГ20 | | | ЛСП02 | | | ПВЛМ | | | РСП05 | | |
| *Рп*, % | 30 | 50 | 70 | 30 | 50 | 70 | 30 | 50 | 70 | 30 | 50 | 70 | 30 | 50 | 70 |
| *Рс*, % | 10 | 30 | 50 | 10 | 30 | 50 | 10 | 30 | 50 | 10 | 30 | 50 | 10 | 30 | 50 |
| Индекс  помещения I | Коэффициент использования с ветового потока n, % | | | | | | | | | | | | | | |
| 0,5 | 14 | 16 | 22 | 12 | 14 | 17 | 23 | 26 | 31 | 11 | 13 | 18 | 19 | 22 | 26 |
| 0,6 | 19 | 21 | 27 | 16 | 18 | 21 | 30 | 33 | 37 | 14 | 17 | 23 | 24 | 27 | 32 |
| 0,7 | 23 | 24 | 29 | 19 | 21 | 24 | 35 | 38 | 42 | 16 | 20 | 27 | 28 | 31 | 36 |
| 0,8 | 25 | 26 | 33 | 21 | 24 | 26 | 39 | 41 | 45 | 19 | 23 | 29 | 31 | 34 | 40 |
| 0,9 | 27 | 29 | 35 | 23 | 25 | 28 | 42 | 44 | 48 | 21 | 27 | 32 | 34 | 37 | 43 |
| 2,0 | 38 | 41 | 48 | 32 | 33 | 35 | 55 | 57 | 60 | 35 | 40 | 46 | 52 | 55 | 59 |
| 3,0 | 44 | 47 | 54 | 35 | 37 | 39 | 60 | 62 | 66 | 41 | 45 | 52 | 58 | 61 | 64 |
| 4,0 | 46 | 50 | 59 | 37 | 39 | 41 | 63 | 65 | 68 | 44 | 48 | 54 | 61 | 64 | 67 |
| 5,0 | 48 | 52 | 61 | 38 | 40 | 42 | 64 | 66 | 70 | 48 | 51 | 57 | 63 | 66 | 69 |

Для этого вычислим индекс помещения по формуле:

 ,

где S – площадь помещения, S = 15 м2;

h – расчетная высота подвеса, h = Н – 0,8 м = 3 – 0,8 = 2,92 м;

A – ширина помещения, А = 3 м;

В – длина помещения, В = 5 м.

Подставив значения, получим:

Зная индекс помещения I, по табл. 1 находим n = 0,22.

Подставим все значения в формулу для определения светового потока *F*:

лм.

Для освещения выбираем люминесцентные лампы типа ЛД80, световой поток которых *F*л = 4070 лк, мощность одной лампы – 80 Вт.

Рассчитаем необходимое количество ламп по формуле:



*N* – определяемое число ламп;

*F* – световой поток, *F* = 33750 лм;

*Fл –* световой поток лампы.

 шт.

При выборе осветительных приборов используем светильники типа ВЗГ20. Каждый светильник комплектуется двумя лампами.

Электрическая мощность общей осветительной системы с учетом потерь в пускорегулирующих аппаратах равна:

*Р*общ= 80 ∙ 8 ∙ 1,2 = 768 Вт = 0,77 кВт.

Анализ показал, при эксплуатации осветительной системы в организации, необходимы замена ламп и очистка светильников от пыли и грязи.

## 4.2 Экологичность работы

Санитарные правила и нормы СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» устанавливают предельно-допустимые уровни (ПДУ) воздействия на людей электромагнитных излучений в диапазоне частот 30 кГц – 300 ГГц (таблица 4.2).

При работе радио и теле и иных передающих станций магнитная составляющая по своей величине не имеет существенного значения, поэтому интенсивность ЭМИ оценивается только по величине напряженности электрического поля (Е, в/м).

Таблица 4.2 – Предельно допустимые уровни ЭМИ, создаваемые передающими станциями

|  |  |
| --- | --- |
| Частота, МГц | ПДУ, в/м |
| 30-60 | 5 |
| 60-120 | 4 |
| 120-240 | 3 |
| 240-300 | 2,5 |

При одновременном облучении от нескольких источников, для которых установлены разные ПДУ, должно соблюдаться следующее условие:

где *Еi* — напряженность электрического поля, создаваемого *i*-источником,

ПДУ*i* — предельно-допустимый уровень для *i*-источника, в/м.

Для защиты населения от ЭМИ мощных передающих станций (свыше 100 кВт) КВ диапазона, они должны размещаться за пределами населенных мест, вдали от жилой застройки.

Вокруг передающих станций создают санитарно-защитные зоны, размеры которых должны обеспечивать предельно-допустимый уровень ЭМИ в населенных местах (таблица 4.3).

Таблица 4.3 – Размеры санитарных зон

|  |  |
| --- | --- |
| Суммарная мощность передатчика,  кВт | Размеры санитарной зоны, м |
| до 10 | в пределах технической территории |
| 10-75 | 200-300 |
| 75-160 | 400-500 |
| более 160 | 500-1000 |

Санитарная зона разделяется на зону строгого режима (50-100 м) и зону ограниченного пользования в зависимости от мощности передатчика. В зоне строгого режима допускается пребывание только работников передающей станции, и ограниченное время. В зоне ограниченного пользования можно располагать объекты, в которых граждане могли бы находиться менее 8 час (гаражи, хозяйственно-бытовые помещения и др.)

Определим напряженности электрического поля на разном расстоянии от передающей антенны.

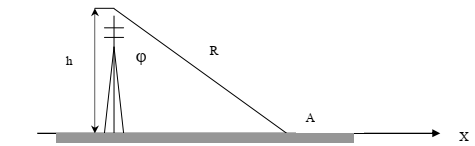


Рисунок 4.1 – Напряжённость электрического поля на расстоянии от антенны

Электрическая напряженность ЭМИ в расчетной точке А определяется по формуле:

где *Р* — мощность источника, Вт,

φ — коэффициент направленности антенны, рад.

где *R* — расстояние от антенны до расчетной точки, м,

*h* — высота антенны, м,

*x* — расстояние от основания антенны до расчетной точки, м.

Электрическая напряженность ЭМИ в жилом помещении определяется по формуле:

*Еж* = *k* · *Е*

где *k* — ослабление ЭМИ стенами здания.

*k* = 1 — для кирпичных стен;

*k* = 0,2 — для панельных стен.

Исходные данные для расчётов приведены в таблице 4.4.

Таблица 4.4 — Исходные данные для расчётов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *h*, м | 1 канал | | 2 канал | | 3 канал | |
| *f*1 | *P*1 | *f*2 | *P*2 | *f*3 | *P*3 |
| 140 | 47 | 1700 | 77 | 4700 | 147 | 7700 |

Где *h* — высота антенны;

*fi* — частота, МГц;

*рi* — мощность передатчика, Вт.

Определим ПДУ для каждого канала по таблице 4.1 и занесем в таблицу 4.4.

Определим электрическую напряженность в расчетных точках и результаты расчета сведем в таблицу 4.5 и рисунок 4.2.

Для *x* = 50 м:

Таблица 4.5 — Мощности электромагнитного излучения на разном расстоянии от передающей станции

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *x* | arctg(*x*/*h*) | *E*1 | *E*2 | *E*3 | *α* |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 50 | 0,34 | 0,88 | 1,47 | 1,89 | 0,56 |
| 100 | 0,62 | 1,03 | 1,72 | 2,199 | 0,76 |
| 150 | 0,82 | 0,99 | 1,66 | 2,12 | 0,71 |
| 200 | 0,96 | 0,91 | 1,51 | 1,93 | 0,59 |
| 250 | 1,06 | 0,81 | 1,35 | 1,73 | 0,47 |
| 300 | 1,13 | 0,73 | 1,21 | 1,54 | 0,38 |
| ПДУ | – | 5 | 4 | 3 | 1 |

Суммарная мощность передатчиков: 1700 + 4700 + 7700 = 14100 Вт = 14,1 кВт. Отсюда по таблице 4.3 определяем размер санитарной зоны — 200 м.

Находим по таблице 4.5 величину *Е* для *x* = 200 м и рассчитываем напряженность электрического поля в кирпичном и панельном домах.

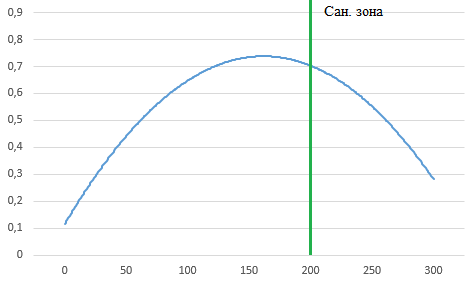


Рисунок 4.2 — Мощности электромагнитного излучения на разном расстоянии от передающей станции

Таблица 4.6 — Напряженность электрического поля в кирпичном и панельном домах

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *E*1 | *E*2 | *E*3 | *α* |
| *x* = 200 | 0.60 | 0.89 | 1.20 | 0.10 |
| Кирпичный  дом | 0.60 | 0.89 | 1.20 | 0.10 |
| Панельный  дом | 0.12 | 0.178 | 0.24 | 0.02 |
| ПДУ |  |  |  |  |

На границе санитарной зоны ЭМИ в кирпичных и панельных домах не превышает допустимые значения.

## 4.3 Организация и обеспечение пожарной безопасности на предприятии

Пожарная безопасность обеспечивается системой предотвращения пожара и системой пожарной защиты в соответствии с техническим регламентом о требованиях пожарной безопасности: федер. закон: [принят Гос. Думой Федер. собрания РФ 22 июля 2008 г. №123-ФЗ]; О противопожарном режиме [утв. и введен в действие постановления Правительства Российской Федерации от 25.04.2012 г. №390 с изм. на 30.12.2017 г.] – Введ. 2012–05–02; НПБ 105–03. Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности. Введ. 2003-08-01. Во всех служебных помещениях обязательно должен быть «План эвакуации людей при пожаре», регламентирующий действия персонала в случае возникновения очага возгорания и указывающий места расположения пожарной техники.

Пожар в рабочем помещении представляет особую опасность, так как сопряжен с большими материальными потерями. Как известно, пожар может возникнуть при взаимодействии горючих веществ, окисления и источников зажигания. На анализируемом рабочем месте присутствуют все три основные фактора, необходимые для возникновения пожара.

Как показал анализ, горючими компонентами в помещении являются: строительные материалы для акустической и эстетической отделки помещений, перегородки, двери, полы, изоляция кабелей и др.

Противопожарная защита — это комплекс организационных и технических мероприятий, направленных на обеспечение безопасности людей, на предотвращение пожара, ограничение его распространения, а также на создание условий для успешного тушения пожара.

Источниками зажигания могут быть электронные схемы, приборы, применяемые для технического обслуживания, устройства электропитания, кондиционирования воздуха, где в результате различных нарушений образуются перегретые элементы, электрические искры и дуги, способные вызвать загорания горючих материалов.

В современном электронном оборудовании очень высокая плотность размещения элементов электронных схем. В непосредственной близости друг от друга располагаются соединительные провода, кабели. При протекании по ним электрического тока выделяется значительное количество теплоты. При этом возможно оплавление изоляции. Для отвода избыточной теплоты служат системы вентиляции и кондиционирования воздуха. При постоянном действии эти системы представляют собой дополнительную пожарную опасность.

При проведении обслуживающих, ремонтных и профилактических работ используются различные смазочные вещества, легковоспламеняющиеся жидкости, прокладываются временные электропроводники, ведут пайку и чистку отдельных узлов. Возникает дополнительная пожарная опасность, требующая дополнительных мер пожарной защиты. В частности, при работе с паяльником следует использовать несгораемую подставку с несложными приспособлениями для уменьшения потребляемой мощности в нерабочем состоянии.

Для тушения пожаров на начальных стадиях широко применяются огнетушители. В настоящее время применяются главным образом углекислотные огнетушители, достоинством которых является высокая эффективность тушения пожара, сохранность электронного оборудования, диэлектрические свойства углекислого газа, что позволяет использовать эти огнетушители даже в том случае, когда не удается обесточить электроустановку сразу.

Рассмотрим аналитический прогноз горения зданий.

Расчет протяженности зон теплового воздействия *R*, м, при горении зданий и промышленных объектов производится по формуле:

где *q*соб –– плотность потока собственного излучения пламени пожара, кВт/м2, (таблица 4.7);

*q*кр –– критическая плотность потока излучения пламени пожара, падающего на облучаемую поверхность и приводящую к тем или иным последствиям, кВт/м2 (таблица 4.8);

*R*\* –– приведенный размер очага горения, м, равный:

–– для горящих зданий,

(1,75…2,0)·––для штабеля пиленого леса,

0,8·*D*рез –– для горения нефтепродуктов в резервуаре;

*l*, *h* –– длина и высота объекта горения, м;

*D*рез –– диаметр резервуара, м.

Таблица 4.7 — Теплотехнические характеристики материалов и веществ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Плотность потока пламени пожара, *qсоб*, кВт/м2 | | | | | | |
| Ацетон | Бензол | Бензин | Керосин | Мазут | Нефть | Древесина |
| 1200 | 2500 | 1780 —1220 | 1520 | 1300 | 874 | 260 |

Задавая ту или иную степень поражения человека, сооружения и т.п., по формуле определяют искомое расстояние от очага пожара. Определим расстояние от очага пожара, возникшего в здании цеха до границы зоны появления ожогов II степени и возгорания горючих материалов.

Таблица 4.8 — Критические значения плотностей потока, падающего излучения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критические значения плотностей потока, *q*кр, кВт/м2 | | | | |
| Безопасное нахождение человека | возгорание древесины через 10 минут | возгорание древесины через 5 минут | возгорание ЛВЖ через 3 минуты | возгорание ГЖ через 3 минуты |
| 1,5 | 14,0 | 17,5 | 35,0 | 41,0 |

Рассчитаем протяженность зоны теплового воздействия *R*, м безопасного нахождения людей при горении деревянного здания и резервуара с керосином.

В случае возгорания в деревянном здании высотой 16 м и длиной 75 м, граница зоны возгорания древесины через 5 минут будет равняться:

м.

Определим расстояние от очага пожара, возникшего в резервуаре с керосином диаметром 16 м до границы зоны возгорания древесины через 5 минут:

м.

Рекомендовано оборудовать рассматриваемое помещение установками стационарного автоматического пожаротушения. Наиболее целесообразно применять установки газового тушения пожара, действие которых основано на быстром заполнении помещения огнетушащим газовым веществом с резким сжижением содержания в воздухе кислорода.

Пожарная безопасность обеспечивается системой предотвращения пожара и системой пожарной защиты. В служебных помещениях вывешены «Планы эвакуации людей при пожаре», регламентирующие действия персонала в случае возникновения очага возгорания и указывающий места расположения пожарной техники.

В необходимых местах размещены ручные огнетушители (углекислотные ОУ-8 в количестве 2 шт).

Средствами обнаружения и оповещения о пожаре являются автоматические датчики-сигнализаторы о пожаре типа ДТП, реагирующие на повышение температуры. Средством оповещения сотрудников о пожаре служит внутрифирменное радио.

## 4.4 Выводы

В соответствии с принятыми нормами в отделе информационных технологий обеспечивается необходимый микроклимат, минимальный уровень шума, созданы удобные и правильные с точки зрения эргономики рабочие места, соблюдены требования технической эстетики.

Для сотрудников отдела в процессе работы одним из важнейших факторов, влияющих на производительность труда при длительной зрительной работе, является достаточное освещение рабочего места. Это достигается правильным выбором и расположением осветительных приборов.

Специальные мероприятия обеспечивают электробезопасность и пожаробезопасность сотрудников.

В целом условия труда инженера соответствуют установленным нормам, сотрудникам обеспечены комфорт и благоприятные условия труда.

# 5 Технико-экономическое обоснование

С целью повышения безопасности передачи данных внутри локальной сети организации была реализована система контроля пакетов данных и контентной фильтрации. Для подсчета экономических показателей разработанной системы был составлен план-график разработки и внедрения системы, определены затраты на её разработку и внедрение, включая затраты на оплату труда, материалы, затраты на использование машинного времени, общехозяйственные затраты.

## 5.1 Спецификация проекта

Спецификация проекта является конструкторским документом в составе проектной документации, который содержит данные о материалах, изделиях и оборудовании. Эти данные необходимы для определения общей потребности, комплектации материальными ресурсами объекта строительства. Спецификация предназначена для:

— определения полного перечня требуемых материалов, изделий, оборудования с их подробным описанием;

— подсчета количества материальных ресурсов.

На основе данных единой ведомости можно определить потребность в материалах, рассчитать стоимость и грамотно составить график их приобретения для обеспечения непрерывности процессов разработки системы.

Спецификация разработанной системы оценки защищённости беспроводных устройств представлена в таблице 5.1.

Таблица 5.1 — Спецификация системы

| Материалы | Единица измерения | Требуемое количество | Цена за единицу руб. | Сумма, руб. |
| --- | --- | --- | --- | --- |
| Wi-Fi адаптер TP-LINK Archer T3U Plus | шт. | 1 | 1 800 | 1 800 |
| Huawei MateBook D 15 BoD-WDH9 | шт. | 1 | 45 990 | 45 990 |
| Итого |  | | | 47 790 |

## 5.2 План-график проектирования и разработки системы

Выбор комплекса работ по разработке проекта производится в соответствии со стандартом «ГОСТ Р ИСО/МЭК 12207-99 Информационная технология. Процессы жизненного цикла программных средств», устанавливающим стадии разработки программных продуктов, приведен в таблице 5.2.

Таблица 5.2 — Комплекс работ по разработке системы анализа радиоэфира

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Содержание работ | Исполнители | Длитель-ность в календар-ных днях | Загрузка | |
| в днях | в % |
| 1 Разработка технического задания | | | | |
| 1.1 Исследование и обоснование разработки | | | | |
| 1.1.1 Постановка задачи | Руководитель | 2 | 2 | 100 |
| Программист | 2 | 100 |
| 1.1.2 Сбор исходных данных | Руководитель | 5 |  | 0 |
| Программист | 5 | 100 |
| 1.2 Поиск аналогов и прототипов | | | | |
| 1.2.1 Анализ существующих методов | Руководитель | 5 | 2 | 40 |
| Программист | 5 | 100 |
| 1.2.2 Обоснование необходимости разработки и внедрения | Руководитель | 5 | 3 | 60 |
| Программист | 5 | 100 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.3 Анализ требований | | | | | | | | |
| 1.3.1 Определение и анализ требований к разрабатываемой системе | | Руководитель | | 4 | | 2 | | 50 |
| Программист | | 4 | | 100 |
| 1.3.2 Определение структуры входных и выходных данных | | Руководитель | | 5 | | 2 | | 40 |
| Программист | | 5 | | 100 |
| 1.3.3 Согласование и утверждение технического задания | | Руководитель | | 4 | | 4 | | 100 |
| Программист | | 4 | | 100 |
| Итого по этапу 1 | | Руководитель | | 30 | | 15 | | 50 |
| Программист | | 30 | | 100 |
| 2 Проектирование | | | | | | | | |
| 2.1 Проектирование архитектуры системы | | Руководитель | | 14 | | 4 | | 29 |
| Программист | | 14 | | 100 |
| 2.2 Выбор аппаратных средств цифровой радиосвязи | | Руководитель | | 14 | | 5 | | 36 |
| Программист | | 14 | | 100 |
| 2.3 Сборка макета системы на основе аппаратных модулей для тестирования и отладки | | Руководитель | | 14 | | 2 | | 14 |
| Программист | | 14 | | 100 |
| Итого по этапу 2 | | Руководитель | | 42 | | 6 | | 14 |
| Программист | | 42 | | 100 |
| 3 Разработка и тестирование модулей системы | | | | | | | | |
| 3.1 Разработка программных модулей системы | | Руководитель | | 14 | | 2 | | 14 |
| Программист | | 14 | | 100 |
| 3.2 Тестирование программной части | | Руководитель | | 12 | | 6 | | 50 |
| Программист | | 12 | | 100 |
| 3.3 Сборка и испытание системы | | Руководитель | | 14 | | 4 | | 29 |
| Программист | | 14 | | 100 |
| 3.4 Анализ результатов экспериментов, формулирование выводов | | Руководитель | | 11 | | 4 | | 36 |
| Программист | | 11 | | 100 |
| Итого по этапу 3 | | Руководитель | | 51 | | 16 | | 31 |
| Программист | | 51 | | 100 |
| 4 Оформление рабочей документации | | | | | | | | |
| 4.1 Проведение расчетов показателей безопасности жизнедеятельности | Руководитель | | 8 | |  | | 0 | |
| Программист | | 8 | | 100 | |
| 4.2 Проведение экономических расчетов | Руководитель | | 8 | |  | | 0 | |
| Программист | | 8 | | 100 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4.3 Оформление пояснительной записки | Руководитель | 13 |  | 0 |
| Программист | 13 | 100 |
| Итого по 4 этапу | Руководитель | 29 | 0 | 0 |
| Программист | 29 | 100 |
| Итого по проекту | Руководитель | 152 | 37 | 24 |
| Программист | 152 | 100 |

На основании вышеописанных таблиц, был составлен план-график по проектированию и разработке системы оценки защищённости беспроводных устройств, показывающий последовательность и взаимосвязь выполнения комплекса работ.

Таблица 5.3 — Календарный график выполнения работ

| Содержание работ | Исполнители | Длитель-ность в календар-ных днях | График работ | |
| --- | --- | --- | --- | --- |
| начало 1 сентября | окончание 31 января |
| Постановка задачи | Руководитель | 2 | 1 сен | 3 сен |
| Программист | 2 | 1 сен | 3 сен |
| Сбор исходных данных | Руководитель |  | 3 сен | 3 сен |
| Программист | 5 | 3 сен | 8 сен |
| Анализ существующих методов | Руководитель | 2 | 8 сен | 10 сен |
| Программист | 5 | 8 сен | 13 сен |
| Обоснование необходимости разработки и внедрения | Руководитель | 3 | 13 сен | 16 сен |
| Программист | 5 | 13 сен | 18 сен |
| Определение и анализ требований к разрабатываемой системе | Руководитель | 2 | 18 сен | 20 сен |
| Программист | 4 | 18 сен | 22 сен |
| Определение структуры входных и выходных данных | Руководитель | 2 | 22 сен | 24 сен |
| Программист | 5 | 22 сен | 27 сен |
| Согласование и утверждение технического задания | Руководитель | 4 | 27 сен | 1 окт |
| Программист | 4 | 27 сен | 1 окт |
| Проектирование архитектуры системы | Руководитель | 4 | 1 окт | 5 окт |
| Программист | 14 | 1 окт | 15 окт |
| Выбор аппаратных средств цифровой радиосвязи | Руководитель | 5 | 15 окт | 20 окт |
| Программист | 14 | 15 окт | 29 окт |
| Сборка макета системы на основе аппаратных модулей для тестирования и отладки | Руководитель | 2 | 29 окт | 31 окт |
| Программист | 14 | 29 окт | 12 ноя |
| Разработка программных модулей системы | Руководитель | 2 | 12 ноя | 14 ноя |
| Программист | 14 | 12 ноя | 26 ноя |
| Тестирование программной части | Руководитель | 6 | 26 ноя | 2 дек |
| Программист | 12 | 26 ноя | 8 дек |
| Сборка и испытание системы | Руководитель | 4 | 8 дек | 12 дек |
| Программист | 14 | 8 дек | 22 дек |
| Анализ результатов экспериментов, формулирование выводов | Руководитель | 4 | 22 дек | 26 дек |
| Программист | 11 | 22 дек | 2 янв |
| Проведение расчетов показателей безопасности жизнедеятельности | Руководитель |  | 2 янв | 2 янв |
| Программист | 8 | 2 янв | 10 янв |
| Проведение экономических расчетов | Руководитель |  | 10 янв | 10 янв |
| Программист | 8 | 10 янв | 18 янв |
| Оформление пояснительной записки | Руководитель |  | 18 янв | 18 янв |
| Программист | 13 | 18 янв | 31 янв |

На основе данных, приведённых в таблице 5.3 сформирован график этапов работ, представленный на рисунке 5.1.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **январь** | **31** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **20** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **10** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **декабрь** | **31** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Рисунок 5.1 — План-график разработки проекта |
| **20** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **10** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **ноябрь** | **30** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **20** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **10** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **октябрь** | **31** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **20** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **10** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **сентябрь** | **30** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **20** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **10** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Содержание работы** | | Постановка задачи | Сбор исходных данных | Анализ существующих методов | Обоснование необходимости разработки и внедрения | Определение и анализ требований к разрабатываемой системе | Определение структуры входных и выходных данных | Согласование и утверждение технического задания | Проектирование архитектуры системы | Выбор аппаратных средств цифровой радиосвязи | Сборка макета системы на основе аппаратных модулей для тестирования и отладки | Разработка программных модулей системы | Тестирование программной части | Сборка и испытание системы | Анализ результатов экспериментов, формулирование выводов | Проведение расчетов показателей безопасности жизнедеятельности | Проведение экономических расчетов | Оформление пояснительной записки |

**5.3 Расчет затрат на разработку проекта**

Предпроизводственные затраты представляют собой единовременные расходы на разработку обеспечивающих или функциональных систем и элементов на всех этапах проектирования, а также затраты на обработку материалов исследования, разработку технического задания, проверки. Сюда включаются затраты на разработку алгоритмов и программ, разработку технического и рабочего проекта системы и её опытной проверки.

Основная заработная плата разработчиков определяется по формуле:

,

где *O* — должностной оклад, руб;

— число рабочих дней в году;

— затраты времени на разработку, рабочие дни.

Число рабочих дней на разработку определяется по формуле:

,

где — календарные дни;

— коэффициент перевода календарных дней в рабочие;

Данный коэффициент равен отношению рабочих дней в году к общему числу календарных дней.

Согласно данным из таблицы 5.1, рассчитаем трудозатраты руководителя:

Рассчитаем трудозатраты программиста:

.

В компании оклад руководителя равен 70 000 руб., а оклад программиста 50 000 руб.

Рассчитаем заработную плату программиста и руководителя по формуле:

Отсюда следует, что основная заработная плата на разработку:

Рассчитаем дополнительную зарплату:

где — коэффициент начисления на дополнительную зарплату.

Премия рассчитывается согласно следующей формуле:

где — коэффициент начисления на премию.

Рассчитаем начисления на единый страховой сбор:

Ввиду того, что проектируемая система должна быть разработана и отлажена с помощью компьютеров, к суммарным затратам на разработку добавляются затраты на их использование.

Доля амортизационных отчислений на компьютерное оборудование, приходящаяся на разработку проекта, определяется по формуле:

где — балансовая стоимость компьютерного оборудования;

— продолжительность использования компьютера программистом, час;

— срок службы компьютерного оборудования. = 6 лет;

число одновременно выполняемых проектов (*z* = 1…3).

 — действительный годовой фонд времени компьютерного оборудования;

Действительный годовой фонд компьютерного оборудования рассчитывается по формуле:

где — номинальный фонд времени работы компьютерного оборудования;

— коэффициент потерь времени; *y* = 3 – 5 %.

Рассчитаем амортизационные отчисления для компьютерного оборудования:

Используемое при конструкторской подготовке системы анализа радиоэфира относится к нематериальным активам предприятия. Амортизационные отчисления на программное обеспечение, приходящиеся на конструкторскую подготовку производства, определяются по формуле:

Где — балансовая стоимость программного обеспечения, руб.;

— срок службы программного обеспечения (при отсутствии фактических данных применяется равным 10 годам).

Рассчитаем затраты электроэнергии, необходимые на проектирование и разработку системы.

где — установленная мощность компьютера, кВт;

— коэффициент загрузки установленной мощности.

— цена электроэнергии, руб/кВт.

Итого затраты на энергию будут следующими:

руб.

Итого затраты на машинное время:

Общехозяйственные расходы:

В таблице 5.4 представлены затраты на разработку и материалы в соответствии со спецификацией, приведённой в п. 5.1.

Таблица 5.4 — Затраты на разработку

|  |  |
| --- | --- |
| Статьи затрат | Затраты |
| Основная заработная плата | 335 013,70 |
| Дополнительная зарплата | 50 252,05 |
| Премия | 134 005,48 |
| Отчисления на социальные нужды | 155 781,37 |
| Затраты на материалы | 47 790,00 |
| Затраты на машинное время | 5 477,23 |
| Общехозяйственные расходы | 904 536,99 |
| ИТОГО | 1 628 226,82 |

В результате проведенных расчетов определено, что себестоимость разработки составляет 1 628 226,82 руб. С учетом нормативной рентабельности *R* = 20% планируемая прибыль *П* составит:

*П* = *С* · *R* = 843 132,97 · 20% = 317 013,36 руб.

A цена продукта составит:

*Ц* = *С* + *П* = 1 585 066,82 + 317 013,36 = 1 945 240,18 руб.

Цена продукта с учётом НДС составит:

*ЦНДС* = 1,20·*Ц* = 1,20 · 1 945 240,18 = 2 334 288,22 руб.

Потенциальными покупателями разработанной системы являются компании, где находятся критически важная информация как о сотрудниках, так и о клиентах, а также отделы информационной безопасности организаций.

# Заключение

На основании изучения существующих способов фильтрации контента были рассмотрены возможные способы контроля входящего и исходящего трафика. Проанализировали возможные принципы и функции работы прокси-сервера, существующие средства фильтрации сетевого трафика. Как показала практика, даже сотрудники Компании могут производить обращения к потенциально опасным ресурсам. Причины могут быть разные как преднамеренные, так и непреднамеренные.

Исходя из проблем фильтрации зашифрованного трафика были разработаны и описаны методы по реализации фильтрации веб-ресурсов. В ходе работы была смоделирована виртуальная сеть из двух виртуальных машин. Одна виртуальная машина выступала в роли прокси-сервера, а другая в роли клиента, сетевые соединения которого проходили через прокси-сервер.

В ходе настройки прокси-сервера были описаны все тонкости установки пакетов и библиотек для правильной работы сервера. Благодаря внедрению прокси squid появилась возможность не только фильтровать входящий и исходящий трафик, но и вести аудит с помощью утилиты sarg.

Использование предложенных в работе методов и средств контентной фильтрации интернет-трафика позволит минимизировать количество потенциальных попыток нарушения информационной безопасности и в целом повысит защищенность локальной сети предприятия.

# Перечень использованных информационных ресурсов

1. Бузов Г.A. Защита от утечки информации по техническим каналам: Учебн. пособие / Бузов Г.A., Калинин C.B., Кондратьев A.B.- M.: Горячая линия - Телеком, 2005. - 416 c.
2. Запечников C.B. Информационная безопасность открытых систем. Часть 1: Учебник для вузов / Запечников C.B., Милославская H.Г., Толстой A.И.,
3. Федеральный закон "Об информации, информационных технологиях и о защите информации" от 27.07.2006 N 149-ФЗ (последняя редакция)
4. Галатенко, В.А. Основы информационной безопасности. - М.: Интуит, 2003.
5. Доктрина информационной безопасности от 09.09.2000 г.
6. Лапонина О.Р. Основы сетевой безопасности. Учебное пособие. — 2005. URL: http://master.cmc.msu.ru/files/Laponina-1.pdf.
7. Галатенко В.А. Стандарты информационной безопасности: курс лекций: учебное пособие / В.А. Галатенко Под ред. Академика РАН В.Б. Бетелина / – М.:ИНТУИТ.РУ «Интернет-университет Информационных Технологий», 2006. – 264 с.
8. Олифер В. Компьютерные сети. Принципы, технологии, протоколы: Учебник для ВУЗов / В. Олифер. - СПб.: Питер, 2012. - 944 c.
9. Попов И.И. Компьютерные сети / И.И. Попов, Н.В. Максимов. - М.: Форум, 2004. - 336 c.
10. Айвенс К. Компьютерные сети. Хитрости / Айвенс К. – СПб.: Питер, 2006. – 298 с.
11. Роббинс А. Bash. Карманный справочник системного администратора. 2-е издание. – М: Диалектика, 2017. – 153 с.
12. Тронкон П., Олбинг К. Bash и кибербезопасность: атака, защита и анализ из командной строки Linux. – СПб: Питер, 2020. – 288 с.
13. Немет Э. Unix и Linux. Руководство системного администратора. 5-e издание / Э. Немет, Г. Снайдер, Т. Хейн, Б. Уэйли, Д. Макни // – М: Диалектика, 2020. – 1168 с.
14. Бэндл Д. Защита и безопасность в сетях Linux. – СПб.: Питер, 2012. – 480 с.
15. Andreasson O. Руководство по iptables [Электронный ресурс] // Проект OpenNET. – URL: https://www.opennet.ru/docs/RUS/iptables/ (дата обращения 17.08.2022).
16. Руководство по работе Squid [Электронный ресурс] // URL: http://www.squid-cache.org// (дата обращения 11.12.2022).
17. Руководство по настройке прозрачного прокси [Электронный ресурс] // URL: https://losst.pro/prozrachnyj-proksi-dlya-https-v-squid// (дата обращения 21.08.2022).
18. Федеральный закон от 22.07.2008 № 123-ФЗ «Технический регламент о требованиях пожарной безопасности».
19. СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания».
20. ГОСТ 12.1.003-2014 МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ.
21. Трудовой кодекс Российской Федерации от 30.12.2001 № 197-ФЗ (ред. от 19.12.2022) [Электронный ресурс]: справочная правовая си-стема КонсультантПлюс. URL: https://www.consultant.ru/document/cons\_doc\_LAW\_346-83/ (дата обраще-ния: 24.12.2022).
22. Налоговый кодекс Российской Федерации (НК РФ) от 31 июля 1998 года № 146-ФЗ [Электронный ресурс]: справочная правовая система КонсультантПлюс. URL: https://www.consultant.ru/document/cons\_doc\_LAW\_196-71/ (дата обращения: 24.12.2022).

**Приложение А.**

**Листинг конфигурационного файла /etc/squid/squid.conf .**

auth\_param basic program /usr/lib/squid/basic\_ncsa\_auth /etc/squid/users

auth\_param basic children 5

auth\_param basic credentialsttl 8 hours

acl auth\_users proxy\_auth REQUIRED

acl localnet 10 src 10.0.0.0/8

acl localnet 16 src 172.16.0.0/16

ac1 localnet 17 src 172. 17.0.0/16

acl adm proxy\_auth admin

acl vip proxy\_auth user\_vip

acl user proxy\_auth user

ac1 vip\_domain dstdomain .twitch.tv

acl user\_domain dstdomain .twitch.tv .vk.com .ok.ru .facebook.com

acl music urlgraph\_regex –I \.mp3$

acl SSL\_ports port 443

ac1 Safe\_ports port 80 # http

acl Safe\_ports port 21 # ftp

acl Safe\_ports port 443 # https

ac1 Safe\_ports port 70 # gopher

acl Safe\_ports port 210 # wais

ac1 Safe\_ports port 1025-65535 # unregistered ports

ac1 Safe\_ports port 280 # http-mgmt

acl Safe\_ports port 488 # gss-http

acl Safe\_ports port 591 # filemaker

acl Safe\_ports port 777 # multiling http

ac1 CONNECT method CONNECT

http\_access deny ! Safe\_ports

http\_access deny CONNECT ISSL\_ports

http\_access allow adm

http\_access deny vip\_domain vip

http\_access deny user\_domain user

http\_access allow auth\_users

http\_access deny music

http\_access allow localnet10 localnet16 localnet17

delay\_pools 6

delay\_class 1 4

delay\_access 1 allow adm

delay\_parameters 1 -1/-1 -1/-1 -1/-1 10000000/10000000

delay\_class 2 4

delay\_access 2 allow user vip youtube

delay\_parameters 2 -1/-1 -1/-1 -1/-1 5000000/5000000

delay\_class 3 4

delay\_access 3 allow localnet16 user

delay\_parameters 3 -1/-1 -1/-1 -1/-1 1000000/1000000

delay\_class 4 4

delay\_access 4 allow localnet17 user

delay\_parameters 4 -1/-1 -1/-1 -1/-1 512000/512000

delay\_class 5 4

delay\_access 5 allow localnet16 vip

delay\_parameters 5 -1/-1 -1/-1 -1/-1 2000000/2000000

delay\_class 6 4

delay\_access 6 allow localnet17 vip

delay\_parameters 6 -1/-1 -1/-1 -1/-1 1000000/1000000

http\_access allow localhost manager

http\_access deny manager

http\_access allow localhost

http\_access deny all

http\_port 10.0.0.1:3128 intercept

https\_port 10.0.0.1:3129 intercept ssl-bump cert=/etc/squid/squid CA.pen

ss1\_bump peek all

ssl\_bump splice all

ssl crtd\_program/usr/lib/squid/ssl\_crt ds /var/lib/ss 1\_dbM 4 MB

sslcrtd\_children 5

coredump\_dir /var/spool/squid

refresh\_pattern ^ftp: 1440 20% 10080

refresh\_pattern ^gopher: 1440 0% 1440

refresh\_pattern -i (/cgi-bin/|\?) 0 0% 0

refresh\_pattern (Release | Packages(.gz)+)$ 0 20% 2880

refresh pattern . 0 20% 4320