Титулка

ЗМІСТ

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

СЛОВАРЬ ТЕРМИНОВ

Открытый (исходный) текст — данные (не обязательно текстовые), передаваемые без использования криптографии.

Шифротекст, шифрованный (закрытый) текст — данные, полученные после применения криптосистемы.

Шифр, криптосистема — совокупность заранее оговоренных способов преобразования исходного секретного сообщения с целью его защиты.

Символ — это любой знак, в том числе буква, цифра или знак препинания.

Алфавит — конечное множество используемых для кодирования информации символов. Стандартный алфавит может быть изменён или дополнен символами. **Ключ** — параметр шифра, определяющий выбор конкретного преобразования данного текста. В современных шифрах криптографическая стойкость шифра целиком определяется секретностью ключа (принцип Керкгоффса).

Шифрование — процесс нормального применения криптографического преобразования открытого текста на основе алгоритма и ключа, в результате которого возникает шифрованный текст.

Расшифровывание — процесс нормального применения криптографического преобразования шифрованного текста в открытый.

Асимметричный шифр, двухключевой шифр, шифр с открытым ключом — шифр, в котором используются два ключа, шифрующий и расшифровывающий. При этом, зная лишь ключ зашифровывания, нельзя расшифровать сообщение, и наоборот.

Открытый ключ — тот из двух ключей асимметричной системы, который свободно распространяется. Шифрующий для секретной переписки и расшифровывающий — для электронной подписи.

Секретный ключ, закрытый ключ — тот из двух ключей асимметричной системы, который хранится в секрете. Криптоанализ — наука, изучающая

математические методы нарушения конфиденциальности и целостности информации.

Система шифрования (шифрсистема) — это любая система, которую можно использовать для обратимого изменения текста сообщения с целью сделать его непонятным для всех, кроме адресата.

Криптостойкостью — это характеристика шифра, определяющая его стойкость к дешифрованию без знания ключа (т.е. способность противостоять криптоанализу).

Криптоаналитик — учёный, создающий и применяющий методы криптоанализа. Криптография и криптоанализ составляют криптологию, как единую науку о создании и взломе шифров (такое деление привнесено с запада, до этого в СССР и России не применялось специального деления).

Криптографическая атака — попытка криптоаналитика вызвать отклонения в атакуемой защищённой системе обмена информацией. Успешную криптографическую атаку называют взлом или вскрытие.

Дешифрование (дешифровка) — процесс извлечения открытого текста без знания криптографического ключа на основе известного шифрованного. Термин дешифрование обычно применяют по отношению к процессу криптоанализа шифротекста (криптоанализ сам по себе, вообще говоря, может заключаться и в анализе криптосистемы, а не только зашифрованного ею открытого сообщения).

Криптографическая стойкость — способность криптографического алгоритма противостоять криптоанализу.

Имитозащита — защита от навязывания ложной информации. Другими словами, текст остаётся открытым, но появляется возможность проверить, что его не изменяли ни случайно, ни намеренно. Имитозащита достигается обычно за счет включения в пакет передаваемых данных имитовставки.

Имитовставка — блок информации, применяемый для имитозащиты, зависящий от ключа и данных.

Электронная цифровая подпись(электронная подпись) — асимметричная имитовставка (ключ защиты отличается от ключа проверки). Другими словами, такая имитовставка, которую проверяющий не может подделать.

Центр сертификации — сторона, чья честность неоспорима, а открытый ключ широко известен. Электронная подпись центра сертификации подтверждает подлинность открытого ключа.

Хеш-функция — функция, которая преобразует сообщение произвольной длины в число («свёртку») фиксированной длины. Для криптографической хеш- функции (в отличие от хеш-функции общего назначения) сложно вычислить обратную и даже найти два сообщения с общей хеш-функцией.

ВВЕДЕНИЕ

Цель практических работ состоит в изучении основных концепций информационной безопасности, понимание уровней информационной безопасности и целей. Определение угроз на аппаратном и сетевом уровнях.

Первая работа изучить основные понятия и уровни информационной безопасности, составляющие и виды информационных угроз. После чего подготовиться к тестированию по заданным аспектам. На занятии разобрать ситуацию по варианту или предложенную руководителем.

Вторая работа данная работа предполагает настройку устройства, предположительно компьютера. В практическом занятии студент должен провести настройку компьютера, целью является защита от самых распространённых ошибок допускаемыми системными администраторами небольших фирм. После чего протестировать и оформить результаты в отчёт.

Третья работа проведение семинара предполагает ознакомить студента с основными стандартами информационной безопасности. Изучить сервисы и механизмы защиты. Так же предполагает разбор нескольких ситуаций из примеров или предложенные студентами.

Четвёртая работа ознакомиться с одним из методов криптографического преобразования информации, а именно стеганографией. Рассмотреть примеры сокрытия данных в файле, использование шумов и стохастической модуляции. Реализовать преобразование одним из методов.

Пятая работа разделена на две части. Первая, предполагает изучение методов сжатия данных. Изучение алгоритма Хаффмана и Лемпеля-Зива, реализация сжатия больших текстов и оценка актуальности.

Шестая работа является базовой по настройке и работе в сетях, рассматривается вариант «белой», безопасной сети, её подключение и общая настройка прав. Работы выполняются в среде Cisco. Можно использовать другое ПО, если оно предоставляет требуемый функционал.

Седьмая работа – это продолжение шестой работы, где студент должен будет реализовать безопасное подключение всей сети к мировой сети Интернет.

Восьмая работа настроена на исследование анализа больших потоков данных, прослушивание сети. Ей цель применить знания анализа сетевых устройств, проанализировать трафик используя предоставленное ПО и продемонстрировать незащищённость протокола FTP.

1 ОСНОВНЫЕ ПОНЯТИЯ БЕЗОПАСНОСТИ

Цель: Изучить основные понятия и уровни информационной безопасности, составляющие и виды информационных угроз.

Теоретические ведомости

1 Основные понятия информационной безопасности

Обеспечение защиты информации волновало человечество всегда. В процессе эволюции цивилизации менялись виды информации, для её защиты применялись различные методы и средства.

Процесс развития средств и методов защиты информации можно разделить на три относительно самостоятельных периода:

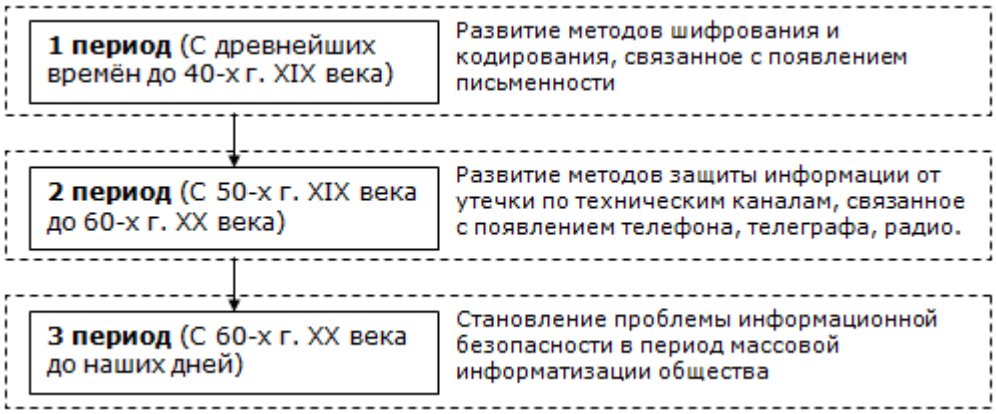

Рисунок 1.1 – Периоды развития

Наблюдаемые в последние годы тенденции в развитии информационных технологий могут уже в недалеком будущем привести к появлению качественно новых (информационных) форм борьбы, в том числе и на межгосударственном уровне, которые могут принимать форму информационной войны, а сама информационная война станет одним из основных инструментов внешней политики, включая защиту государственных интересов и реализацию любых форм агрессии. Это является одной из причин, почему полезно ознакомиться с основными принципами обеспечения ИБ в ведущих зарубежных странах.

Прежде чем говорить об обеспечении безопасности персональных данных, необходимо определить, что же такое информационная безопасность. Термин "информационная безопасность" может иметь различный смысл и трактовку в зависимости от контекста. В данном пособии под информационной безопасностью мы будем понимать защищённость информации и поддерживающей инфраструктуры от случайных или преднамеренных воздействий естественного или искусственного характера, которые могут нанести неприемлемый ущерб субъектам информационных отношений, в том числе владельцам и пользователям информации и поддерживающей инфраструктуры.[3]

Информационная безопасность – это защищённость информации и поддерживающей её инфраструктуры от случайных или преднамеренных воздействий естественного или искусственного характера, которые могут нанести ущерб владельцам или пользователям информации.

В ряде случаев понятие "информационная безопасность" подменяется термином "компьютерная безопасность". В этом случае информационная безопасность рассматривается очень узко, поскольку компьютеры только одна из составляющих информационных систем. Несмотря на это, в рамках изучаемого курса основное внимание будет уделяться изучению вопросов, связанных с обеспечением режима информационной безопасности применительно к вычислительным системам, в которых информация хранится, обрабатывается и передаётся с помощью компьютеров. Согласно определению, компьютерная безопасность зависит не только от компьютеров, но и от поддерживающей инфраструктуры, к которой можно отнести системы электроснабжения, жизнеобеспечения, вентиляции, средства коммуникаций, а также обслуживающий персонал.

2 Составляющие информационной безопасности

Обеспечение информационной безопасности в большинстве случаев связано с комплексным решением трёх задач:

Конфиденциальность – состояние информации, при котором доступ к ней осуществляют только субъекты, имеющие на него право.

Целостность – состояние информации, при котором отсутствует любое её изменение либо изменение осуществляется только преднамеренно субъектами, имеющими на него право;

Доступность – состояние информации, при котором субъекты, имеющие право доступа, могут реализовывать его беспрепятственно.

Угрозы информационной безопасности – совокупность условий и факторов, создающих потенциальную или реально существующую опасность нарушения безопасности информации.[1, 5]

Атака – это попытка реализации угрозы. Кто предпринимает такую попытку, называется *злоумышленником*. Потенциальные злоумышленники называются *источниками угрозы*.

Угроза является следствием наличия уязвимых мест или уязвимости в информационной системе. Уязвимости могут возникать по разным причинам, например, в результате непреднамеренных ошибок программистов при написании программ.

Угрозы можно классифицировать по нескольким критериям:

- по свойствам информации (доступность, целостность, конфиденциальность), против которых угрозы направлены в первую очередь;
- по компонентам информационных систем, на которые угрозы нацелены (данные, программы, аппаратура, поддерживающая инфраструктура);
- по способу осуществления (случайные/преднамеренные, действия природного/техногенного характера);
- по расположению источника угроз (внутри/вне рассматриваемой ИС).

Обеспечение информационной безопасности является сложной задачей, для решения которой требуется комплексный подход.

3 Уровни защиты информации

3.1 Законодательный уровень

Законодательный уровень является основой для построения системы защиты информации, так как даёт базовые понятия предметной области и определяет меру наказания для потенциальных злоумышленников. Этот уровень играет координирующую и направляющую роли и помогает поддерживать в обществе негативное (и карательное) отношение к людям, нарушающим информационную безопасность.

Законодательно-правовой уровень включает комплекс законодательных и иных правовых актов, устанавливающих правовой статус субъектов информационных отношений, субъектов и объектов защиты, методы, формы и способы защиты, их правовой статус. Кроме того, к этому уровню относятся стандарты и спецификации в области информационной безопасности. Система законодательных актов и разработанных на их базе нормативных и организационно-распорядительных документов должна обеспечивать организацию эффективного надзора за их исполнением со стороны правоохранительных органов и реализацию мер судебной защиты и ответственности субъектов информационных отношений. К этому уровню можно отнести и морально-этические нормы поведения, которые сложились традиционно или складываются по мере распространения вычислительных средств в обществе. Морально-этические нормы могут быть регламентированными в законодательном порядке, т. е. в виде свода правил и предписаний. Наиболее характерным примером таких норм является Кодекс профессионального поведения членов Ассоциации пользователей ЭВМ США. Тем не менее, эти нормы большей частью не являются обязательными, как законодательные меры.

3.2 Административный уровень

Это комплекс мер, предпринимаемых локально руководством организации. Включает комплекс взаимокоординируемых мероприятий и технических мер, реализующих практические механизмы защиты в процессе создания и эксплуатации систем защиты информации. Организационный уровень должен охватывать все структурные элементы систем обработки данных на всех этапах их жизненного цикла: строительство помещений, проектирование системы, монтаж и наладка оборудования, испытания и проверки, эксплуатация.

Разработка политики безопасности - дело тонкое, поскольку у каждой организации есть своя специфика. Здесь бессмысленно переносить практику режимных государственных организаций на коммерческие структуры, учебные заведения или персональные компьютерные системы. В этой области целесообразно предложить, во-первых, основные принципы разработки политики безопасности, а, во-вторых, - готовые шаблоны для наиболее важных разновидностей организаций.

3.3 Процедурный уровень

К процедурному уровню относятся меры безопасности, реализуемые людьми. В отечественных организациях накоплен богатый опыт составления и реализации процедурных (организационных) мер, однако проблема состоит в том, что они пришли из докомпьютерного прошлого, поэтому нуждаются в существенном пересмотре.

Можно выделить следующие группы процедурных мер:

- управление персоналом;
- физическая защита;
- поддержание работоспособности;
- реагирование на нарушения режима безопасности;
- планирование восстановительных работ.

Для каждой группы в каждой организации должен существовать набор регламентов, определяющих действия персонала. В свою очередь, исполнение этих регламентов следует отработать на практике.

3.4 Программно-технический уровень

Согласно современным воззрениям, включает три подуровня: физический, технический (аппаратный) и программный.

Физический подуровень решает задачи с ограничением физического доступа к информации и информационным системам, соответственно к нему относятся технические средства, реализуемые в виде автономных устройств и систем, не связанных с обработкой, хранением и передачей информации: система охранной сигнализации, система наблюдения, средства физического воспрепятствования доступу (замки, ограждения, решётки и т. д.).

Средства защиты аппаратного и программного подуровней непосредственно связаны с системой обработки информации. Эти средства либо встроены в аппаратные средства обработки, либо сопряжены с ними по стандартному интерфейсу.

К аппаратным средствам относятся схемы контроля информации по четности, схемы доступа по ключу и т. д.

К программным средствам защиты, образующим программный подуровень, относятся специальное программное обеспечение, используемое для защиты информации, например антивирусный пакет и т. д. Программы защиты могут быть как отдельные, так и встроенные. Подчеркнём, что формирование режима информационной безопасности является сложной системной задачей, решение которой в разных странах отличается по содержанию и зависит от таких факторов, как научный потенциал страны, степень внедрения средств информатизации в жизнь общества и экономику, развитие производственной базы, общей культуры общества и, наконец, традиций и норм поведения.

4 Виды информационных угроз

Информационные угрозы могут быть обусловлены:

– естественными факторами (пожар, наводнение, и др.);

– человеческими факторами.

Последние, в свою очередь, подразделяются на:

– Угрозы, носящие случайный, неумышленный характер. Это угрозы, связанные с ошибками процесса подготовки, обработки и передачи информации;

– Угрозы, обусловленные умышленными, преднамеренными действиями людей. Это угрозы, связанные с несанкционированным доступом к ресурсам АИС.

Умышленные угрозы преследуют цель нанесения ущерба пользователям АИС и, в свою очередь, подразделяются на активные и пассивные. Угрозы также подразделяются на внутренние, возникающие внутри управляемой организации, и внешние.

Под внутренними угрозами – понимаются угрозы безопасности информации инсайдером (исполнителем) которых является внутренний по отношению к ресурсам организации субъект (инсайдер).

Под внешними угрозами – понимаются угрозы безопасности информации инициатором (исполнителем) которых является внешний по отношению к ресурсам организации субъект (удаленный хакер, злоумышленник).

Задания

1 Тестирование

1) В чем заключается проблема информационной безопасности?
2) Дайте определение понятию «информационная безопасность».
3) Что понимается под «компьютерной безопасностью»?
4) Перечислите составляющие информационной безопасности.
5) Приведите определение доступности информации.
6) Приведите определение целостности информации.
7) Приведите определение конфиденциальности информации.
8) Каким образом взаимосвязаны между собой составляющие информационной безопасности? Приведите собственные примеры.
9) Перечислите задачи информационной безопасности общества.
10) Перечислите уровни формирования режима информационной безопасности.
11) Дайте краткую характеристику законодательно-правового уровня.
12) Какие подуровни включает программно-технический уровень?
13) Что включает административный уровень?
14) В чем особенность морально-этического подуровня?

2 Рассмотрение ситуации

Оценив ситуацию соответствующую варианту нужно:

1) Определить источник угрозы.
2) Пострадавшее лицо.
3) Классифицировать вид угрозы.
4) Определить угрозу доступности, целостности, конфиденциальности.
5) Организовать меры по защите.

Так же организовать меры по защите информации в данных обстоятельствах и дальнейшее упреждение данной модели.

Пример выполнения работы

Сотрудница отделения коммерческого банка разместила фото с id своей карты в социальной сети.

В данной ситуации мы можем явно видеть, что сотрудник допустил халатность. В результате чего безопасность компании ставиться под вопрос.

1) Источником угрозы является сотрудница, а так же любые лица пытающиеся проникнуть в административную часть здания с поддельным пропуском на её имя.

2) Пострадавшим лицом является учреждение, в частности отдел по безопасности данного объекта. При бездействии круг пострадавших лиц может сильно увеличиться.

3) Классификация угрозы:

– угроза обусловлена человеческим фактором;

– носящим случайный, неумышленный характер;

– угроза является внутренней.

4) Такие аспекты безопасности как доступность и целостность не нарушены. В данном контексте нарушена только конфиденциальность рабочих пропусков компании.

5) Меры по защите должны включать:

а) Немедленное блокирование пропуска сотрудницы, выдача нового.

б) Усиленная проверка входящих в здание по пропускам в течении недели.

в) Проверка персонала, находящегося в здании.

г) Добавление/удаление пропусков происходят в следящем режиме.

д) Сверка активности сотрудницы.

е) Провести инструктаж на тему "Политика безопасности в организации".

Варианты задания

1) В СМИ утекли результаты анализов одного из известных деятелей;

2) Ученик, взломав систему оценивания колледжа исправил себе бал по дисциплине;

3) Во время грозы были повреждены электролинии. В связи с этим более 200 клиентов охранной компании остались без наблюдения на 10 часов;

4) Используя брешь в интернет-сети страховой компании, хакер заменил данные нескольких клиентов;

5) Интернет-магазин использует небезопасный канал. Клиент, совершив покупку передал сумму третьему лицу;

6) Подкуплен сотрудник, после чего неизвестный проник в здание отделения полиции.

7) Сотрудник аудиторской компании использовал данные в своих целях;

8) После взлома сервера компании по информационной защите ключи доступа пользователей появились на «чёрном рынке»;

9) Ночью из офиса была украдена печать адвоката, объект находится под охраной;

10) Сотрудник компании по разработке ПО скрыто вставлял мониторинг в продукт;

11) Сбой в работе компании по обеспечению vps серверов;

12) Обнаружен задержка интернет канала биржи. Предположительно злоумышленники, подключившись к каналу получают данные первыми;

13) Сотрудник не соблюдал правила производства. В связи с чем завод потерял несколько партий продукта.

14) Зависание информационной системы на железной дороге привело к столкновению поездов.

15) Офис туристической компании был затоплен во время стихийного бедствия.

2 ПАКЕТЫ АНТИВИРУСНЫХ ПРОГРАММ

Цель: Ознакомление с основными функциями антивирусного ПО. Изучить современные средства защиты от вирусов.

Теоретические ведомости

Вредоносная программа – любое программное обеспечение, предназначенное для получения несанкционированного доступа к вычислительным ресурсам самой ЭВМ или к информации, хранимой на ЭВМ, с целью несанкционированного использования ресурсов ЭВМ или причинения вреда (нанесения ущерба) владельцу информации, и/или владельцу ЭВМ, и/или владельцу сети ЭВМ, путём копирования, искажения, удаления или подмены информации.

1 История

Сказать, где и когда появился первый вирус, невозможно, поскольку таких данных в природе не существует. Если на «компьютере» Чарльза Бэббиджа, «отца» первой вычислительной машины, вирусов ещё не было, к середине семидесятых годов прошлого века они стали весьма распространенным и неприятным для большинства явлением. Тем не менее, предпосылки к их созданию появились практически сразу же с созданием первых ЭВМ.

Еще в 1940 году математик Джон фон Нейман написал книгу[2], в которой были описаны самовоспроизводящиеся математические автоматы, то есть принципы, которые легли в основу всех вирусов. В 1959 году американский научный журнал «Scientific American» опубликовал статью Л. Пенроуза, рассказывавшую о самостоятельно распространяющихся биологических структурах. Автор рассмотрел способности подобных структур к мутациям, активации и размножению. Другой ученый, Ф. Шталь, полученные из этой статьи знания реализовал на практике. Работая

оператором в научно-исследовательской лаборатории, он имел доступ к мощнейшей для того времени ЭВМ – IBM 650. Эксперимент очень удивил Шталя, превзойдя все его ожидания. Получившийся в результате «мутации» математических алгоритмов электронный «зверек» удалил все следы своих «родителей», присутствовавших в системе, после чего самоуничтожился.

Естественно, все вышеперечисленные труды и опыты были направлены не для того, чтобы нынешние вирусописатели ежедневно выбрасывали в Интернет мегабайты новой «заразы». Изначально эти исследования, относившиеся к области создания искусственного интеллекта, представляли собой академический интерес. Однако любое открытие, сделанное в мирных целях, может быть без особых трудностей превращено в мощное оружие разрушения.

В 1961 году среди компьютерщиков была очень популярна игра «*Darwin*». Её сюжет и смысл были просты: игрок руководил «расой», которая должна была уничтожить своих конкурентов. Выигрывал тот, кто захватит всю отданную под игровой процесс оперативную память. Особых действий в игре не требовалось: необходимо было лишь размножить принадлежащих к своей расе на свободные ячейки ОЗУ или же захватить ячейки противника. Подобный алгоритм очень похож на логику работы деструктивных программ.

Широкое распространение компьютерных сетей стало катализатором появления на свет первых деструктивных программ – *компьютерных вирусов*.

1.1 Конструкторы вредоносных программ

В 1992 году хакер, известный под ником Dark Avenger, выпустил в свет утилиту MtE (Mutation Engine). С её помощью любой, даже самый примитивный вирус можно было сделать полиморфным. Этим же человеком был впервые создан вирус Peach, наделенный способностью обходить антивирусное ПО. Peach удалял базу изменений программы Central Point AntiVirus. Эта программа, не находив свою базу данных, считала, что

запущена впервые, и создавала её вновь. Таким образом, вирус обходил защиту и продолжал заражать систему.

Группа программистов, известная в сети, как Nowhere Man, выпустила конструктор вирусов VCL (Virus Creation Laboratory). Отныне любой школьник, даже не владеющий языками программирования, мог вооружиться конструктором и собрать вирус любого типа и разрушительной силы. С появлением VCL и так немалый «поток» новых компьютерных вредителей стал просто огромным.

2 Антивирусные склоки

К 1997 году операционная система *Linux*, ранее считавшаяся оплотом «чистоты и стабильности», больше не являлась платформой, свободной от вирусов. Linux.Bliss, распространявшийся посредством конференций UseNet, заражал исполняемые файлы этой ОС.

В этом же году было отмечено появление двух новых типов червей, распространявшихся через IRC и FTP. Особо большим их количеством мог «похвастаться» IRC, во многом из-за своей популярности, а также многочисленных «дыр» mIRC – основного клиента подобных сетей.

Под конец XX века в погоне за лидерством стали нередки скандалы среди производителей антивирусов. Так, представители компании McAfee объявили о том, что ее программисты обнаружили ошибку в антивирусе фирмы Dr.Solomon’s. Суть заявления сводилась к тому, что Dr.Solomon’s мог находить новые и технически совершенные вирусы только в специальном «усиленном» режиме, в который переключался лишь после нахождения обычных, примитивных червей. В результате антивирус показывал хорошие скоростные результаты при сканировании незараженных дисков, и отличные показатели обнаружения при работе с зараженными файлами. В ответ Dr.Solomon`s подала иск в суд на McAfee, причиной которого стала её «некорректно построенная рекламная компания». В итоге вся «заварушка» завершилась покупкой McAfee контрольного пакета акций Dr.Solomon`s.

Спустя некоторое время публичное заявление сделали тайваньские разработчики из фирмы Trend Micro, обвинившие McAfee и Symantec в якобы «нарушении их патента на сканирование данных». Миру были сразу представлены доказательства о «безгрешности» компаний, однако Trend Micro добилась своего, получив отменную бесплатную рекламу в средствах массовой информации.

3 Классификация

У компаний-разработчиков антивирусного программного обеспечения существуют собственные классификации и номенклатуры вредоносных программ. Приведённая в этой статье классификация основана на номенклатуре «Лаборатории Касперского».

3.1 По вредоносной нагрузке

1) Помехи в работе заражённого компьютера: начиная от открытия-закрытия поддона CD-ROM и заканчивая уничтожением данных и поломкой аппаратного обеспечения. Поломками известен, в частности, Win32.CIH.

2) Блокировка антивирусных сайтов, антивирусного ПО и административных функций ОС с целью усложнить лечение.

3) Саботирование промышленных процессов, управляемых компьютером

4) Инсталляция другого вредоносного ПО.

5) Загрузка из сети (downloader).

6) Распаковка другой вредоносной программы, уже содержащейся внутри.

7) Кража, мошенничество, вымогательство и шпионаж за пользователем. Для кражи может применяться сканирование жёсткого диска, регистрация нажатий клавиш (Keylogger) и перенаправление пользователя на поддельные сайты, в точности повторяющие исходные ресурсы.

8) Похищение данных, представляющих ценность или тайну.

9) Кража аккаунтов различных служб (электронной почты, мессенджеров, игровых серверов…). Аккаунты применяются для рассылки спама. Также

через электронную почту зачастую можно заполучить пароли от других аккаунтов, а виртуальное имущество в MMOG — продать.

10) Кража аккаунтов платёжных систем.

11) Блокировка компьютера, шифрование файлов пользователя с целью шантажа и вымогательства денежных средств (см. Ransomware). В большинстве случаев после оплаты компьютер или не разблокируется, или вскоре блокируется второй раз.

12) Использование телефонного модема для совершения дорогостоящих звонков, что влечёт за собой значительные суммы в телефонных счетах.

13) Платное ПО, имитирующее, например, антивирус, но ничего полезного не делающее (fraudware или scareware).

14) Прочая незаконная деятельность:

15) Получение несанкционированного (и/или дарового) доступа к ресурсам самого компьютера или третьим ресурсам, доступным через него, в том числе прямое управление компьютером (так называемый backdoor).

16) Организация на компьютере открытых релеев и общедоступных прокси-серверов.

17) Заражённый компьютер (в составе ботнета) может быть использован для проведения DDoS-атак.

18) Сбор адресов электронной почты и распространение спама, в том числе в составе ботнета.

19) Накрутка электронных голосований, щелчков по рекламным баннерам.

20) Генерация монет платёжной системы Bitcoin.

21) Шуточное ПО, делающее какие-либо беспокоящие пользователя вещи.

22) Adware — программное обеспечение, показывающее рекламу.

23) Spyware — программное обеспечение, занимающееся массовым сбором малоценной информации — например, конфигурации компьютера, каталогов диска, активности пользователя.

24) «Отравленные» документы, дестабилизирующие ПО, открывающее их (например, архив размером меньше мегабайта может содержать гигабайты данных и надолго «завесить» архиватор).

25) Программы удалённого администрирования могут применяться как для того, чтобы дистанционно решать проблемы с компьютером, так и для неблаговидных целей.

26) Руткит нужен, чтобы скрывать другое вредоносное ПО от посторонних глаз. Это возможно благодаря тесной интеграции руткита с операционной системой.

27) Иногда вредоносное ПО для собственного «жизнеобеспечения» устанавливает дополнительные утилиты: IRC-клиенты, программные маршрутизаторы, открытые библиотеки перехвата клавиатуры… Такое ПО вредоносным не является, но из-за того, что за ним часто стоит истинно вредоносная программа, детектируется антивирусами. Бывает даже, что вредоносным является только скрипт из одной строчки, а остальные программы вполне легитимны.

28) Файлы, не являющиеся истинно вредоносными, но в большинстве случаев нежелательные

3.2 По методу размножения

Эксплойт – теоретически безобидный набор данных (например, графический файл или сетевой пакет), некорректно воспринимаемый программой, работающей с такими данными. Здесь вред наносит не сам файл, а неадекватное поведение ПО с ошибкой. Также эксплойтом называют программу для генерации подобных «отравленных» данных.

Логическая бомба в программе срабатывает при определённом условии, и неотделима от полезной программы-носителя.

Троянская программа. По своему действию является противоположностью вирусам и червям. Его предлагают загрузить под видом законного приложения, однако вместо заявленной функциональности он

делает то, что нужно злоумышленникам. Трояны не самовоспроизводятся и не распространяются сами по себе. Нынешние трояны эволюционировали до таких сложных форм, как, например, backdoor (троян, пытающийся взять на себя администрирование компьютера) и троян-загрузчик (устанавливает на компьютер жертвы вредоносный код).

Компьютерный вирус размножается в пределах компьютера и через сменные диски. Размножение через сеть возможно, если пользователь сам выложит заражённый файл в сеть. Вирусы, в свою очередь, делятся по типу заражаемых файлов (файловые, загрузочные, макро-, автозапускающиеся); по способу прикрепления к файлам (паразитирующие, «спутники» и перезаписывающие) и т. д.

Сетевой червь способен самостоятельно размножаться по сети. Делятся на IRC-, почтовые, размножающиеся с помощью эксплойтов и т. д.

Загрузчик – является небольшой частью кода, используемой для дальнейшей загрузки и установки полной версии. После того как загрузчик попадает в систему путем сохранения вложения электронного письма или, например, при просмотре зараженной картинки, он соединяется с удаленным сервером и загружает весь вирус.

4 Виды антивирусной защиты

Современные антивирусы – это комплексные программные пакеты, как правило, содержащие несколько взаимосвязанных и взаимодополняющих модулей, нацеленные на борьбу со всем спектром компьютерных угроз.

В современных антивирусах могут задействоваться следующие виды антивирусной защиты:

Сравнение с вирусным образцом – вирусной сигнатурой кода, шаблоном поведения вредоносной программы или цифровым отпечатком в «черном» списке известных угроз. Эта разновидность антивирусной защиты заключается в исследовании подозрительной программы на наличие

признаков, характерных для вредоносного ПО. Например, реализуя данный вид защиты, антивирус ищет *сигнатуры* -- последовательности кода, уникальные для определённого вируса.

Поведенческий мониторинг – разновидность антивирусной защиты, основанная на проверке объектов во время осуществления чтения, записи и других операций. Для проведения мониторинга антивирусная программа располагается в оперативной памяти и действует как обработчик системных событий. При старте какой-либо операции, которая может привести к заражению, антивирусный монитор запускает проверку обрабатываемого объекта (документа, программы и т.д.).

Обнаружение изменений – вид антивирусной защиты, базирующийся на контроле целостности программных компонентов компьютера. При заражении вирусы модифицируют файлы, системный реестр или загрузочные сектора диска. Антивирусная программа определяет, был ли изменен объект с помощью подсчета кодов циклического контроля (CRC-сумм) и других методов.

Эвристический анализ. Данный вид антивирусной защиты основан на том, что выполняемые вирусами действия и их последовательность отличаются от поведения большинства программ. Поэтому анализ последовательностей команд и системных вызовов подозрительного программного обеспечения помогает выносить правильное решение о его вредоносности.

Лечение – разновидность антивирусной защиты, состоящая в удалении вредоносных объектов и восстановлении нормальных параметров компьютерной системы.

Репутационный сервис – новейший вид антивирусной защиты, получивший распространение в последние годы и базирующийся на проверке репутации программ, веб-ресурсов и почтовых систем. Такая проверка проводится с использованием «облачных» серверов репутации, поддерживаемых ведущими разработчиками антивирусного ПО, и основана на

постоянно обновляемых списках «легитимных», вредоносных и подозрительных ресурсов. Преимуществом репутационных сервисов является очень высокая скорость реакции на появление новых угроз.

Существуют и устаревшие, теперь уже редко используемые виды антивирусной защиты, например, иммунизация, которая заключается в том, что в памяти компьютера размещается программа, сообщающая вирусам, избегающим повторного заражения, о том, что система уже инфицирована.

Реализуют антивирусную защиту следующие модули:

- Антивирусный сканер
- Антивирусный монитор, использующий многочисленные технологии защиты
- Поведенческий блокиратор
- Антивирусный ревизор или система контроля CRC
- Антивирусный фаг или доктор.

Задания

1) Подготовить краткий доклад по выбору антивирусного ПО для полноценной защиты. Разрешается использовать любые доступные источники информации.

Рекомендация: Собранный материал будет наиболее актуальным, если включить в него данные, полученные практическим путем. Для этого при возможности, установите демонстрационную версию заданного пакета ПО и протестируйте ее в течении нескольких дней.

2) Создайте таблицу «Оценка антивирусного ПО» где, запишите достоинства и недостатки данного пакета. Дайте оценку каждому из параметров на основе подготовленного материала.

3) Провести анализ собранной информации и сделать выводы.

3 МИРОВЫЕ СТАНДАРТЫ БЕЗОПАСНОСТИ

Тема: Стандарты информационной безопасности распределённых систем.

Цель: Изучить сервисы и механизмы защиты распределённых систем. Разбор планирования систем.

Требования к семинару

1 Студент должен знать

1) Основное содержание стандартов по информационной безопасности распределённых систем;
6) Основные сервисы безопасности в вычислительных сетях;
7) Наиболее эффективные механизмы безопасности;
8) Задачи администрирования средств безопасности.

2 Студен должен уметь

9) Выбирать механизмы безопасности для защиты распределенных систем.

Термины для подготовки

Распределённая информационная система – совокупность аппаратных и программных средств, используемых для накопления, хранения, обработки, передачи информации между территориально удалёнными пользователями.

Сервис (*Сервисная деятельность*) – это вид деятельности, направленный на удовлетворение потребностей социальных субъектов посредством оказания услуг.

Сервис безопасностис – это деятельность государственных и частных организаций, а также отдельных специалистов, направленная на удовлетворение потребностей социальных субъектов в безопасности.

Цель сервиса безопасности – удовлетворение потребностей в безопасности индивидуальных и групповых социальных субъектов. *Сущность сервиса* безопасности состоит в оказании услуг, направленных на обеспечение безопасности.

Услуга безопасности – это деятельность субъекта безопасности, направленная на удовлетворение потребности заказчика в безопасности, а также результат взаимодействия исполнителя и заказчика услуги безопасности, выраженный в виде полезного эффекта.

Темы для обсуждения

1) Механизмы безопасности.
2) Сервисы безопасности в вычислительных сетях.
3) Функций и механизмов безопасности.
4) Администрирование средств безопасности.
5) Международные стандарты.
6) Стандарты ГОСТ и ДСТУ.

Литература для ознакомления

1. Щербаков А. Ю. Введение в теорию и практику компьютерной безопасности. – М.: Издательство Молгачева С. В., 2001.
2. Теория и практика обеспечения информационной безопасности / Под ред. П. Д. Зегжды. – М: Яхтсмен, 1996.
3. Галатенко В. А. Основы информационной безопасности. – М: Интернет-Университет Информационных Технологий – ИНТУИТ. РУ, 2003.
4. Галатенко В. А. Стандарты информационной безопасности. – М: Интернет-Университет Информационных Технологий – ИНТУИТ. РУ, 2004.
5. www.iso.ch – Web-сервер Международной организации по стандартизации.

4 МЕТОДЫ СОКРЫТИЯ ИНФОРМАЦИИ

Цель: Ознакомиться с методом стеганографии. Рассмотреть примеры сокрытия информации в файлах, реализовать один из методов.**Теоретические ведомости**

Стеганография – набор средств и методов сокрытия факта передачи сообщения.

Слово ***стеганография*** в переводе с греческого буквально означает ***тайнопись*** (steganos - тайна, секрет; graphy - запись).

Стеганография представляет собой совокупность методов, основывающихся на различных принципах, которые обеспечивают сокрытие самого факта существования секретной информации в той или иной среде, а также средств реализации этих методов. К ней можно отнести огромное множество секретных средств связи, таких как невидимые чернила, микрофотоснимки, условное расположение знаков, тайные (скрытые) каналы, средства связи с плавающими частотами, голография и т.д.

Основным определяющим моментом в стеганографии является стеганографическое преобразование. Стеганографические технологии активно используются для решения следующих основных задач:

- защиты информации с ограниченным доступом от несанкционированного доступа;
- защиты авторских прав на некоторые виды интеллектуальной собственности;
- преодоления систем мониторинга и управления сетевыми ресурсами;
- камуфляжа программного обеспечения;
- создания скрытых каналов утечки чувствительной информации от законного пользователя.

1 Классификация стеганографических методов

В современной стеганографии, в целом, можно выделить в направления:

Рисунок 4.1– Классификация методов стеганографической защиты

По аналогии с криптографическими системами, в стеганографии различают системы с *секретным ключом* и системы с *открытым ключом*. Учитывая все многообразие стеганографических систем, сведем их к следующим типам: *безключевым* стегосистемам, системам с *секретным ключом*, системам с *открытым ключом* и *смешанным стегосистемам*.

Классификация методов сокрытия информации. Большинство методов компьютерной стеганографии базируется на двух принципах.

Первый – файлы, которые не требуют абсолютной точности, могут быть до определенной степени видоизменены без потери функциональности.

Второй принцип основан на отсутствии специального инструментария или неспособности органов чувств человека надежно различать незначительные изменения в таких исходных файлах.

Для методов компьютерной стеганографии можно ввести определенную классификацию(рис. 4.2).

Рисунок 4.2– Классификация методов сокрытия информации

В основном, для таких методов характерны низкая степень скрытности, низкая пропускная способность и слабая производительность.

Методы лингвистической стеганографии – сокрытия секретных сообщений в тексте - известны еще со средневековья. Можно выделить следующие методы, которые встречаются в современных лингвистических стеганографах:

– методы искажения формата текстового документа;
– синтаксические методы;
– семантические методы;
– методы генерации стеганограмм с помощью скрываемого сообщения.

2 Методы искажения формата текстового документа

Сокрытие данных путем *изменения формата текстовых* файлов обычно проводится так, чтобы стандартные текстовые редакторы не смогли выявить признаков присутствия дополнительной информации.

Методы работают успешно до тех пор, пока тексты представлены в коде ASCII.

Синтаксические методы. К этим методам лингвистической стеганографии относятся методы изменения пунктуации и методы изменения

стиля и структуры текста. К синтаксическим методам относятся методы изменения стиля или структуры текста без существенного изменения его значения или тона.

Семантические методы. Эти методы стеганографии аналогичны синтаксическим методам.

Таблица 4.1 – Фрагмент таблицы синонимов

1	0
след	отпечаток
дыра	отверстие
оборона	Защита
овации	аплодисменты

На рисунке 4.2 приведен пример подхода к сокрытию данных, в котором секретное сообщение управляет перефразированием текста контейнера. В результате получается стеганограмма, которая имеет тот же самый смысл, что и текст контейнера.

Рисунок 4.3– Пример работы семантической стегосистемы SubiText

Методы генерации стеганограмм. Методы, которые полностью порождают стеганограмму на основе защищаемых данных. В таких методах секретная информация не внедряется в текст, а представляется полностью всей стеганограммой.

Сокрытие данных в изображении и видео. Визуальная среда (цифровые изображения и видео) обладают большой избыточностью различной природы:

– кодовой избыточностью;

- меж пиксельной избыточностью;
- психовизуальной зависимостью.

Сокрытие информации в звуковой среде. Особое развитие нашли методы цифровой стеганографии в аудио среде. С их помощью обеспечивается пересылка больших объемов скрытых данных в звуковых сообщениях, которые транслируются по телевизионной, радио или телефонной сети.

3 Стеганографические методы защиты данных в звуковой среде

Метод наименьших значащих битов применяется при цифровом представлении аудио сигнала и пригоден для использования при любых скоростях связи.

Методы широкополосного кодирования используют те же принципы, что методы сокрытия данных в изображениях.

Метод сокрытия в эхо-сигнале. Скрывать данные можно также путем внедрения эха в звуковой сигнал.

Музыкальные стегосистемы. Музыкальная форма звуковой среды занимает большую часть информационного пространства Internet.

Основное отличие музыкальной стеганографии от импровизации состоит в том, что целью является не расширение образов базового музыкального произведения, а внесение изменений, которые сохраняют мелодию основного произведения, соответствуют всем правилам построения данного произведения и при этом кодируют скрываемое сообщение, не искажая главной темы произведения.

Задания

С помощью изученного материала сокрыть один текстовый файл внутри контейнера. В зависимости от варианта из табл. 4.2. Варианты выбирать по остатку от деления на количество вариантов (пример: 15%8=7).

Размер файлов подбирать самостоятельно, нежелательно использовать маленькие файлы.

Рассмотреть альтернативные методы, сравнить с предоставленным методом, записать преимущества и недостатки.

В отчёт приложить данные из *контейнера*, *ключа* и *стеганограммы*.

Таблиця 4.2 – Таблица вариантов

№	Описание
1	Методом LSB скрыть текстовую информацию в изображении
2	Методом КДБ внедрить звуковой файл в изображение
3	Скрыть изображение в контейнере используя алгоритм Blowfish
4	Используя особенности формата RAR добавить текст(>1Мб)
5	Внедрить текстовый файл в записанный шум, используя формат wav
6	Скрыть зашифрованный текстовый файл в PDF документе используя сигнатуры
7	Используя метод фазового кодирования скрыть информацию в аудиофайле
8	Создать архив со встроенным изображением (Rarjpeg)

Вопросы для самоконтроля

1) Дайте определение понятию стеганография.
2) Какие задачи решаются при помощи стеганографических технологий.
3) Классификация стеганографических методов.
4) Классификация методов сокрытия информации.
5) Типы стеганографических систем.
6) Методы лингвистической стеганографии.
7) Синтаксические и семантические методы.
8) Сокрытие данных в изображении и видео.

5 АРХИВАЦИЯ ДАННЫХ

Цель: Изучение методов сжатия данных, алгоритма Хаффмана и Лемпеля-Зива..

Архивация данных

Архивация (сжатие данных) – есть процесс представления информации в ином виде (перекодирования) с потенциальным уменьшением объёма, требуемого для её хранения. Существует множество классов различных алгоритмов сжатия данных, каждый из которых ориентирован на свою область применения[4].

Основоположником науки о сжатии информации принято считать *Клода Шеннона*. Его теорема об оптимальном кодировании показывает, к чему нужно стремиться при кодировании информации и насколько та или иная информация при этом сожмется. Кроме того, им были проведены опыты по эмпирической оценке, избыточности английского текста. Шеннон предлагал людям угадывать следующую букву и оценивал вероятность правильного угадывания. На основе ряда опытов он пришел к выводу, что количество информации в английском тексте колеблется в пределах $0,6 - 1,3$ бита на символ.

Сжатие данных – это процесс, обеспечивающий уменьшение объёма данных путём сокращения их избыточности. Сжатие данных связано с компактным расположением порций данных стандартного размера. Сжатие данных можно разделить на два основных типа:

Сжатие без потерь (полностью обратимое) – это метод сжатия данных, при котором ранее закодированная порция данных восстанавливается после их распаковки полностью без внесения изменений. Для каждого типа данных, как правило, существуют свои оптимальные алгоритмы сжатия без потерь.

Сжатие с потерями – это метод сжатия данных, при котором для обеспечения максимальной степени сжатия исходного массива данных часть содержащихся в нём данных отбрасывается. Для текстовых, числовых и табличных данных использование программ, реализующих подобные методы сжатия, является неприемлемыми. В основном такие алгоритмы применяются для сжатия аудио и видеоданных, статических изображений.

1 Алгоритмы архивации данных

Алгоритм сжатия данных – это алгоритм, который устраняет избыточность записи данных.

Отношение сжатия – одна из наиболее часто используемых величин для обозначения эффективности метода сжатия.

$$Отношение\ сжатия = \frac{размер\ \ выходного\ \ потока}{размер\ \ входного\ \ потока} \qquad (5.1)$$

Значение 0,6 означает, что данные занимают 60% от первоначального объема. Значения больше 1 означают, что выходной поток больше входного.

Коэффициент сжатия – величина, обратная отношению сжатия.

$$Коэффициент\ \ сжатия = \frac{размер\ входного\ потока}{размер\ выходного\ потока} \qquad (5.2)$$

Значения больше 1 обозначают сжатие, а значения меньше 1 расширение.

Средняя длина кодового слова – это величина, которая вычисляется как взвешенная вероятностями сумма длин всех кодовых слов.

$$L_{cp} = p_1 \cdot L_1 + p_2 \cdot L_2 + \ldots + p_n \cdot L_n, \qquad (5.3)$$

где p_n – вероятности кодовых слов, L_1, L_2, L_3 – длины кодовых слов.

Статистические методы – методы сжатия, присваивающие коды переменной длины символам входного потока, причем более короткие коды

присваиваются символам или группам символам, имеющим большую вероятность появления во входном потоке. Лучшие статистические методы применяют кодирование Хаффмана.

Словарное сжатие – это методы сжатия, хранящие фрагменты данных в "словаре" (некоторая структура данных). Если строка новых данных, поступающих на вход, идентична какому-либо фрагменту, уже находящемуся в словаре, в выходной поток помещается указатель на этот фрагмент. Лучшие словарные методы применяют метод Зива-Лемпела.

Рассмотрим несколько известных алгоритмов сжатия данных более подробно.

1.1 Алгоритм Хаффмана

В основе алгоритма Хаффмана лежит идея кодирования битовыми группами. Сначала проводится частотный анализ входной последовательности данных, то есть устанавливается частота вхождения каждого символа, встречащегося в ней. После этого, символы сортируются по уменьшению частоты вхождения.

Основная идея состоит в следующем: чем чаще встречается символ, тем меньшим количеством бит он кодируется. Результат кодирования заносится в словарь, необходимый для декодирования. Рассмотрим простой пример, иллюстрирующий работу алгоритма Хаффмана.

Пусть задан текст «beep boop beer!», рассмотрим таблицу с частотами всех символов:

Символ	‘b’	‘e’	‘p’	‘ ’	‘o’	‘r’	‘!’
Частота	3	4	2	2	2	1	1

По частоте использования

Символ	‘r’	‘!’	‘p’	‘o’	‘ ’	‘b’	‘e’

После этого создадим элементы бинарного дерева для каждого символа и представим их как очередь с приоритетом, в качестве которого будем использовать частоту.

Возьмём первые два элемента из очереди и создадим третий(рис. 5.1), который будет их родителем. Этот новый элемент поместим в очередь с приоритетом, равным сумме приоритетов двух его потомков. Иначе говоря, равным сумме их частот.

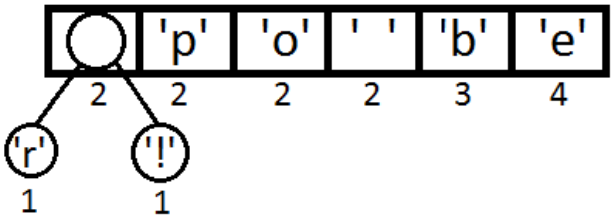

Рисунок 5.1– Пример объединения элементов

Далее будем повторять шаги, аналогичные предыдущему:

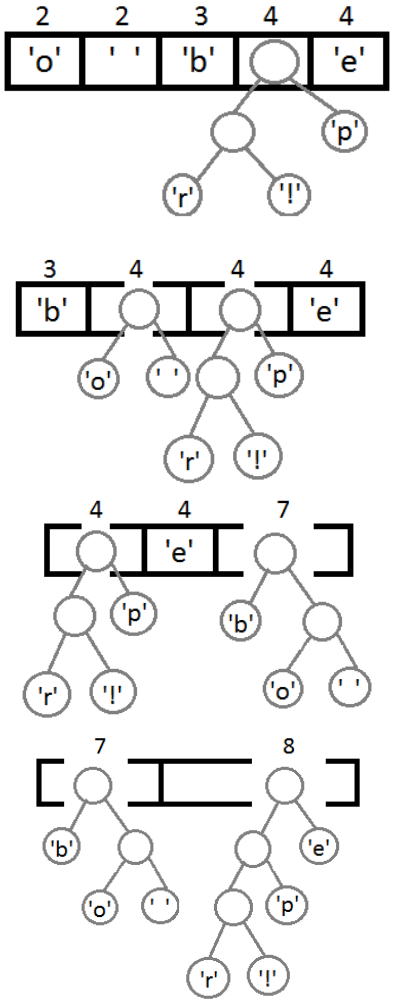

Рисунок 5.2 – Построение дерева

Теперь, после объединения последних двух элементов с помощью их нового родителя, мы получим итоговое бинарное дерево:

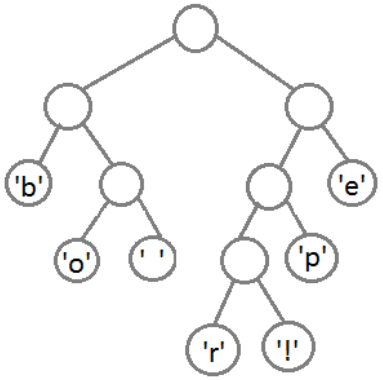

Рисунок 5.3 – Вид бинарного дерева

Осталось присвоить каждому символу его код(рис. 1.1). Для этого запустим обход в глубину и каждый раз, рассматривая правое поддерево, будем записывать в код 1, а рассматривая левое поддерево – 0.

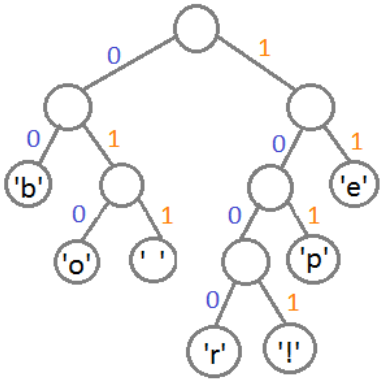

Рисунок 5.4 – Построение кода

В результате соответствие символов кодовым значениям получится следующим:

Таблица 5.1 – Кодовые значения символов

Символ	‘b’	‘e’	‘p’	‘ ’	‘o’	‘r’	‘!’
Кодовое значение	00	11	101	011	010	1000	1001

Декодирование битов происходит следующим образом: нужно обходить дерево, отбрасывая левое поддерево, если встретилась единица и правое, если

встретился 0. Продолжать обход нужно до тех пор, пока не встретим лист, т.е. искомое значение закодированного символа.

Например, закодированной строке «101 11 101 11» и нашему дереву декодирования соответствует строка «pepe».

Входная строка:

```
beep boop beer!
```

Входная строка в двоичном виде:

```
0110 0010 0110 0101 0110 0101 0111 0000 0010 0000 0110 0010 0110
1111 0110 1111 0111 0000 0010 0000 0110 0010 0110 0101 0110 0101
0111 0010 0010 0001
```

Закодированная строка:

```
0011 1110 1011 0001 0010 1010 1100 1111 1000 1001
```

Разница между ASCII-кодировкой строки и её же видом в коде Хаффмана очевидна.

Алгоритм Хаффмана универсальный, его можно применять для сжатия данных любых типов, но он малоэффективен для файлов маленьких размеров (за счет необходимости сохранения словаря). В настоящее время данный метод практически не применяется в чистом виде, обычно используется как один из этапов сжатия в более сложных схемах. Это единственный алгоритм, который не увеличивает размер исходных данных в худшем случае (если не считать необходимости хранить таблицу перекодировки вместе с файлом).

1.2 Алгоритм Лемпеля-Зива

Процесс сжатия выглядит следующим образом. Последовательно считываются символы входного потока и происходит проверка, существует ли в созданной таблице строк такая строка. Если такая строка существует, считывается следующий символ, а если строка не существует, в поток

заносится код для предыдущей найденной строки, строка заносится в таблицу, а поиск начинается снова. Например, если сжимают байтовые данные (текст), то строк в таблице окажется 256 (от «0» до «255»). Если используется 10-битный код, то под коды для строк остаются значения в диапазоне от 256 до 1023. Новые строки формируют таблицу последовательно, т. е. можно считать индекс строки ее кодом. Алгоритму декодирования на входе требуется только закодированный текст, поскольку он может воссоздать соответствующую таблицу преобразования непосредственно по закодированному тексту. Алгоритм генерирует однозначно декодируемый код за счет того, что каждый раз, когда генерируется новый код, новая строка добавляется в таблицу строк. LZW постоянно проверяет, является ли строка уже известной, и, если так, выводит существующий код без генерации нового. Таким образом, каждая строка будет храниться в единственном экземпляре и иметь свой уникальный номер. Следовательно, при дешифровании при получении нового кода генерируется новая строка, а при получении уже известного, строка извлекается из словаря.[6]

Кодирование

Пусть мы сжимаем последовательность(табл. 5.2).

abacabadabacabae

1) Изначально инициализируем таблицу, добавив в неё все строки из одного символа.

2) Согласно алгоритму, мы добавим к изначально пустой строке *«a»* и проверим, есть ли строка *«a»* в таблице. Поскольку мы при инициализации занесли в таблицу все строки из одного символа, то строка *«a»* есть в таблице.

3) Далее мы читаем следующий символ *«b»* из входного потока и проверяем, есть ли строка *«ab»* в таблице. Такой строки в таблице пока нет.

4) Добавляем в таблицу <5> *«ab»*. В поток: <0>;

5) *«ba»* — нет. В таблицу: <6> *«ba»*. В поток: <1>;

6) *«ac»* — нет. В таблицу: <7> *«ac»*. В поток: <0>;

7) *«ca»* — нет. В таблицу: <8> *«ca»*. В поток: <2>;

8) *«ab»* — есть в таблице; *«aba»* — нет.

В таблицу: <9> *«aba»*. В поток: <5>;

9) *«ad»* — нет. В таблицу: <10> *«ad»*. В поток: <0>;

10) *«da»* — нет. В таблицу: <11> *«da»*. В поток: <3>;

11) *«aba»*— есть в таблице; *«abac»* — нет.

В таблицу: <12> *«abac»*. В поток:<9>;

12) *«ca»* — есть в таблице; *«cab»* — нет.

В таблицу: <13> *«cab»*. В поток:<8>;

13) *«ba»* — есть в таблице; *«bae»* — нет.

В таблицу: <14> *«bae»*. В поток:<6>;

Последняя строка *«e»,* за ней идет конец сообщения. Выводим в поток <4>.

Таблиця 5.2 – Словарь кодирования LZW

Текущая строка	Текущий символ	Следующий символ	Вывод		Словарь	
			Код	Биты		
ab	a	b	0	000	5:	ab
ba	b	a	1	001	6:	ba
ac	a	c	0	000	7:	ac
ca	c	a	2	010	8:	ca
ab	a	b	-	-	-	-
aba	b	a	5	101	9:	aba
ad	a	d	0	000	10:	ad
da	d	a	3	011	11:	da
ab	a	b	-	-	-	-
aba	b	a	-	-	-	-
abac	a	c	9	1001	12:	abac
ca	c	a	-	-	-	-
cab	a	b	8	1000	13:	cab
ba	b	a	-	-	-	-
bae	a	e	6	0110	14:	bae
e	e	-	4	0100	-	-

Декодирование

Особенность LZW заключается в том, что для декомпрессии нам не надо сохранять таблицу строк в файл для распаковки. Алгоритм построен таким образом, что мы в состоянии восстановить таблицу строк, пользуясь только потоком кодов. Теперь представим, что мы получили закодированное сообщение, приведённое выше, и нам нужно его декодировать. Прежде всего, нам нужно знать начальный словарь, а последующие записи словаря мы можем реконструировать уже на ходу, поскольку они являются просто конкатенацией предыдущих записей.

Таблиця 5.3 – Декодировка LZW

Данные		На выходе	Новая запись	
Биты	Код		Полная	Частичная
000	0	a	- -	5: a?
001	1	b	5: ab	6: b?
000	0	a	6: ba	7: a?
010	2	c	7: ac	8: c?
101	5	ab	8: ca	9: ab?
000	0	a	9: aba	10: a?
011	3	d	10: ad	11: d?
1001	9	aba	11: da	12: aba?
1000	8	ca	12: abac	13: ca?
0110	6	ba	13: cab	14: ba?
0100	4	e	14: bae	- -

Достоинства и недостатки

+ Не требует вычисления вероятностей встречаемости символов/кодов.
+ Данный тип компрессии не вносит искажений в исходный графический файл, и подходит для сжатия растровых данных любого типа.
+ Для декомпрессии не надо сохранять таблицу строк в файл для распаковки. Алгоритм построен таким образом, что мы в состоянии восстановить таблицу строк, пользуясь только потоком кодов.
– Алгоритм не проводит анализ входных данных поэтому не оптимален.

Задания

1) Взять данные соответственно варианту из файла *«Prac5_Var.txt»*

2) Удалить лишнюю информацию методом *Хаффмана*.

3) Провести операцию методом *Лемпеля-Зива*.

4) Сравнить результаты проведённых операций.

5) Используя архиваторы сжать файл (> 4 Гбайт) повторив шаги 2-4.

6) Описать актуальность архивации для различных объёмов данных.

7) Сделать выводы по применению методов сжатия в различных криптосистемах.

Вопросы для самоконтроля

1) Что такое архивация данных?

2) Цель архивации?

3) Какие Вы знаете методы архивации?

4) Опишите принцип дерева Хаффмана.

5) Опишите алгоритм LZ77 или его аналог.

6) Сферы применения заданных алгоритмов.

7) Как выбрать алгоритм, если данные заранее известны?

6 ПОСТРОЕНИЕ БЕЗОПАСНОЙ СЕТИ

Цель: вивчити можливості і правила конфігурації міжмережевих екранів для забезпечення безпеки комп'ютерних мереж.

Теоретические ведомости

Міжмережевий екран (firewall, брендмауер) - система міжмережевого захисту, що дозволяє розділити загальну мережу на дві або більш частин і реалізувати набір правил, визначальних умови проходження пакетів з даними через кордон з однієї частини загальної мережі в іншу. Як правило, ця межа проводиться між корпоративною (локальної) мережею і глобальною мережею Internet, хоча її можна організувати і усередині корпоративної мережі підприємства. Міжмережевий екран пропускає через себе весь трафік, ухвалюючи для кожного пакету рішення - пропускати його або відкинути - на основі визначених для нього набору правил фільтрації

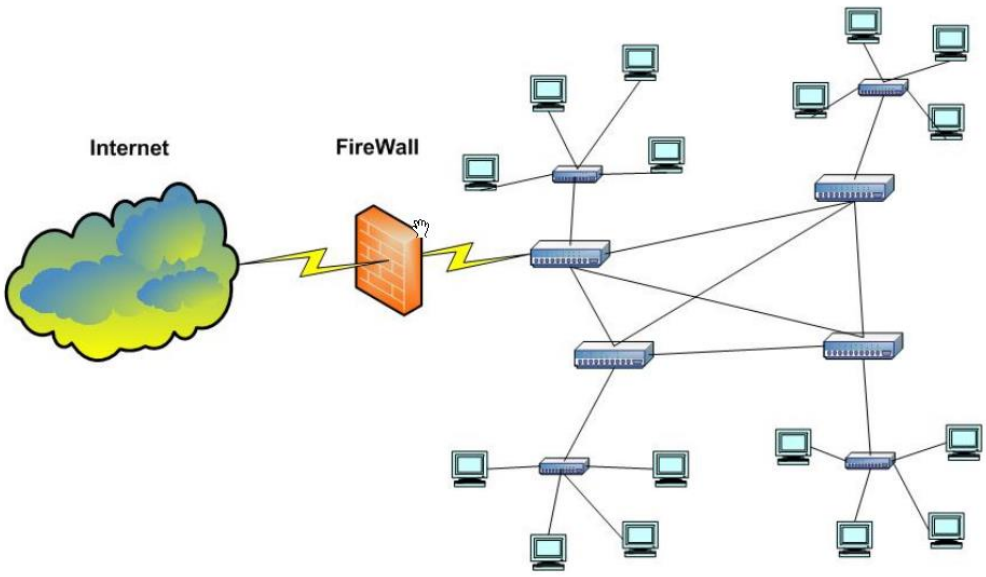

Рисунок 6.1– Схема встановлення міжмережевого екрану

Існують чотири основні засоби, за допомогою яких міжмережеві екрани здійснюють контроль доступу і забезпечують реалізацію політик захисту:

Управління сервісами – визначення типів служб Internet, до яких можна дістати доступ з внутрішньої мережі назовні і із зовнішнього оточення усередину; міжмережевий екран може фільтрувати потік даних на основі IP- адрес і номерів портів TCP;

Управління напрямом руху – визначення напряму, в якому можуть ініціюватися і проходити через міжмережевий екран запити до тих або інших служб;

Управління користувачами – надання доступу до служб залежно від прав доступу користувачів, що звертаються до цих служб; ця функція звичайно застосовується до локальних користувачів; однак вона може застосовуватися і до потоку даних, що поступає від зовнішніх користувачів, що вимагає реалізації в якійсь формі технології аутентифікації, наприклад, забезпечуваної протоколом IPSec;

Управління поведінкою – контроль за використанням окремих служб; так, міжмережевий екран може фільтрувати електронну пошту, відсіваючи спам чи ж вирішувати доступ ззовні певної частини інформації, що знаходиться на локальному Web-сервері.

Можна виділити наступні типи міжмережевих екранів:

– фільтруючі маршрутизатори (packet-filtering router);

– шлюзи мережевого рівня;

– шлюзи прикладного рівня.

Звичайно міжмережеві екрани включають всі або більшість з цих компонент.

Фільтруючий маршрутизатор є маршрутизатором або працюючою на сервері програмою, що фільтрує пакети, які входять в корпоративну мережу і виходять з неї. Фільтрація пакетів звичайно здійснюється на основі інформації, що міститься в IP і TCP заголовках пакетів, найчастіше для цього використовуються:

– IP адреса відправника пакету;

– IP адреса одержувача пакету;

– порт системи відправника;

– порт системи одержувача.

Фільтр пакетів звичайно представляється у вигляді списку правил, що використовують значення полів заголовків IP і TCP. Якщо виявляється відповідність одному з правил, то на підставі цього правила ухвалюється рішення про можливість передачі пакету далі. При невідповідності всім правилам виконується операція, передбачена для використання за умовчанням. Для неї існують наступні варіанти локальних політик захисту:

Default = discard. Все, що не дозволено, заборонено.

Default = forward. Все, що незаборонено, дозволено.

Як приклад роботи фільтруючого маршрутизатора розглянемо реалізацію політики безпеки, що допускає певні з'єднання з локальною корпоративною мережею 123.4/16 (рис. 6.2). При цьому з'єднання по протоколу telnet дозволені тільки з хостом 123.4.5.6, що виконує роль прикладного telnet-шлюзу, а з'єднання по протоколу SMTP - тільки з двома хостами 123.4.5.7 і 123.4.5.8, що виконують роль серверів електронної пошти. Обмін по протоколу NNTP дозволений тільки від зовнішнього сервера новин з адресою 129.6.48.254 і лише з NNTP сервером локальної мережі 123.4.5.9. Використання мережевої служби синхронізації часу NTP дозволене для всіх комп'ютерів локальної мережі.

Позитивні якості фільтруючих маршрутизаторів:

– порівняно невисока вартість;

– гнучкість у визначенні правил фільтрації;

– невелика затримка при проходженні пакетів.

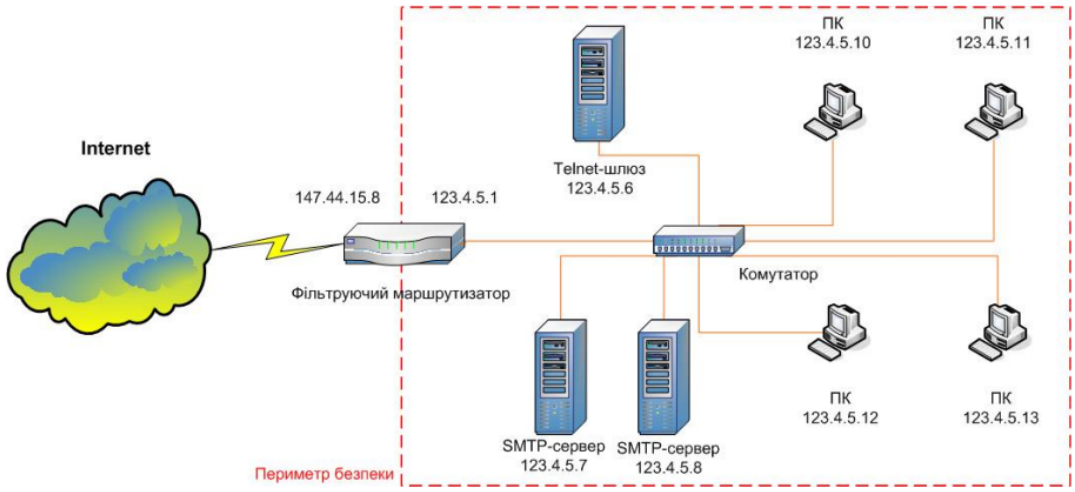

Рисунок 6.2– Приклад набору правил фільтрації

Таблиця 6.1 – Правила фільтрації

Протокол	Адреса від-правника	Адреса одер-жувача	Порт відправ-ника	Порт одер-жувача	Дія
TCP		123.4.5.6	>1023	23	allow
TCP		123.4.5.7	>1023	25	allow
TCP		123.4.5.8	>1023	25	allow
TCP	1239.6.48.254	123.4.5.9	>1023	119	allow
UDP		123.4/16	>1023	123	allow
*	*	*	*		deny

Недоліки фільтруючих маршрутизаторів:

– внутрішня мережа маршрутизується з Internet;

– правила фільтрації важкі в описі і відсутні засоби їх тестування;

– відсутня аутентифікація на призначеному для користувача рівні.

Шлюзи мережевого рівня інколи називають системами трансляції мережевих адрес (Network Address Translation - NAT), оскільки вони виключають пряму взаємодію між авторизованим клієнтом і зовнішнім хост - комп'ютером. Шлюз мережевого рівня приймає запит довіреного клієнта на конкретні послуги і після перевірки чи задовольняє клієнт базовим критеріям фільтрації (наприклад, чи може DNS- сервер визначити IP-адреса клієнта і асоційоване з ним ім'я) встановлює з'єднання із зовнішнім хостом. Шлюз мережевого рівня виконує процедуру трансляції адрес, при якій відбувається перетворення внутрішніх IP адрес локальної мережі на одну "надійну" IP

адресу інтерфейсу міжмережевого екрану, через який передаються всі витікаючі з локальної мережі пакети.

Кожен раз, коли витікаючий пакет передається через NAT – маршрутизатор локальна IP адреса відправника замінюється на "чесну" IP адресу (рис. 6.3). Окрім цього, в полі порту відправника заголовка TCP заноситься 16-бітовий індекс таблиці трансляції адрес (з максимум 65536 рядками). Кожен рядок таблиці містить цей індекс, первинну IP адресу відправника і первинний порт відправника. Після описаної заміни прораховуються контрольні суми заголовків IP і TCP і заносяться у відповідні поля пакету. Коли NAT- маршрутизатор отримує у відповідь пакети для даного процесу, значення порту одержувача в заголовку TCP використовується для пошуку індексу в таблиці трансляції адрес. З визначеного по індексу рядка витягується локальна IP адреса відправника і первинний TCP порт відправника, які заносяться у відповідні поля пакету. Після цього знову перераховуються контрольні суми в полях IP і TCP заголовків, заносяться у відповідні поля пакету, після чого пакет прямує на робочу станцію локальної мережі.

Шлюз мережевого рівня відстежує процедуру квитування зв'язку по протоколу TCP через порядкові номери сегментів даних. Коли сеанс завершується, шлюз видаляє відповідний елемент з таблиці і розриває з'єднання. Недоліком шлюзів мережевого рівня є те, що після встановлення зв'язку вони не можуть перевіряти вміст пакетів на прикладному рівні. Також для них відсутня призначена для користувача аутентифікація.

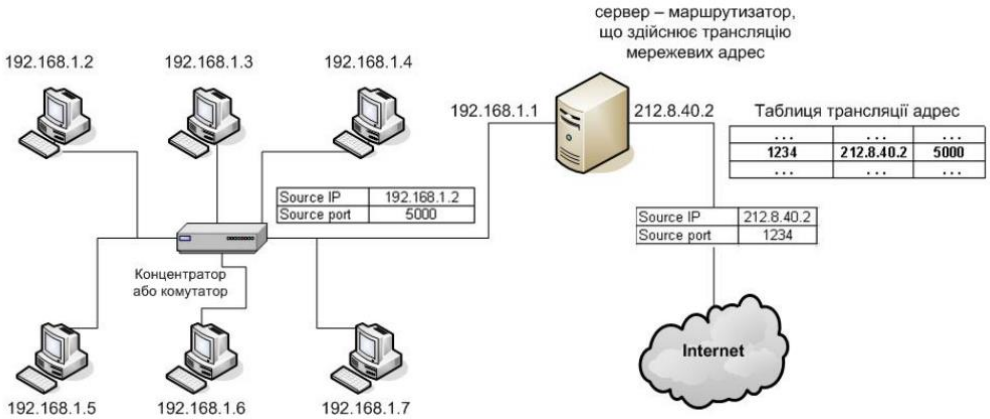

Рисунок 6.3– Трансляція мережевих адрес шлюзом мережевого рівня

Шлюзи прикладного рівня також виключають пряму взаємодію між

авторизованим клієнтом і зовнішнім хостом, проте при цьому вони фільтрують всі вхідні і витікаючі пакети на прикладному рівні. Це досягається установкою і настройкою програмного забезпечення повноважних серверів – проксі - серверів (proxy servers), що перенаправляють через шлюз інформацію прикладного рівня, що генерується хостами (рис. 6.4). Користувач зв'язується з цим шлюзом за допомогою такого побудованого на основі TCP/IP застосування, як HTTP-браузер, Telnet-термінал або FTP- клієнт. Ці застосування передають шлюзу ім'я віддаленого вузла, до якого необхідно дістати доступ. Шлюз зв'язується з відповідним застосуванням віддаленого вузла і ретранслює сегменти TCP, що містять дані застосування локального користувача. Шлюз можна настроїти так, щоб він підтримував тільки певні можливості застосування, які адміністратор мережі вважає допустимими з погляду безпеки.

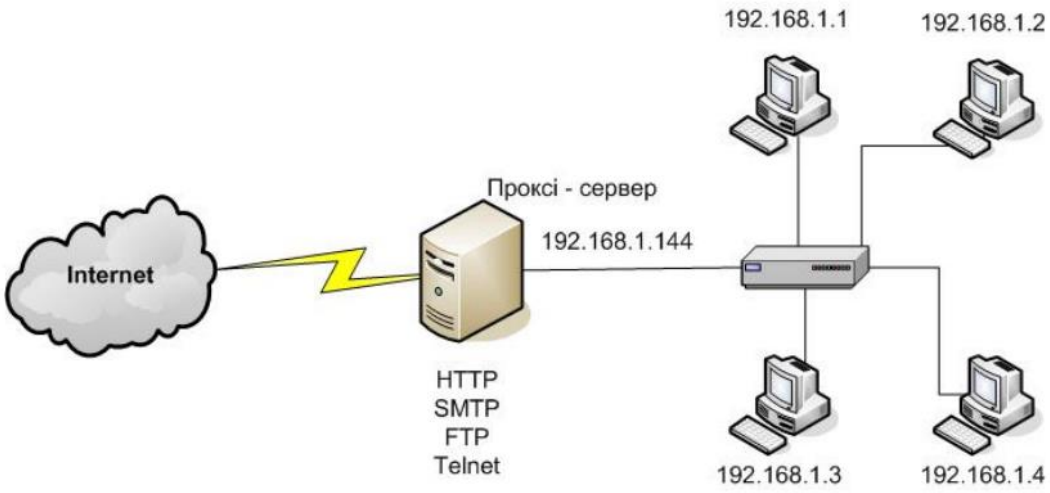

Рисунок 6.4– Шлюз прикладного рівня–проксі–сервер

У загальному випадку шлюзи прикладного рівня забезпечують надійніший захист, ніж фільтри пакетів. Замість того щоб перевіряти численні можливі комбінації дозволів і заборон на рівні TCP і IP, шлюзу прикладного рівня доводиться розглядати лише обмежене число застосувань. Крім того, при цьому легко протоколювати і контролювати весь вхідний потік даних прикладного рівня. Проте, для досягнення вищого рівня безпеки і гнучкості

шлюзи прикладного рівня і фільтруючі маршрутизатори можуть об'єднуватися в одному міжмережевому екрані. Переваги використання проксі-серверів:

– пропускають пакети лише тих служб, які їм доручено обслуговувати (частіше за все HTTP, HTTPS, FTP, Gopher);

– приховують структуру мережі, що захищається;

– дозволяють виконувати кешування запрошуваної Інтернет інформації і будувати ієрархію кешів;

– дозволяють виконувати аутентифікацію користувачів;

– дозволяють вести статистику відвідувань за витікаючими IP адресами.

Основним недоліком шлюзів даного типа є додаткове навантаження на процесор, що створюється кожним з'єднанням. Насправді між двома кінцевими користувачами встановлюється два зв'язані з'єднання, загальною точкою яких є шлюз, і шлюзу доводиться перевіряти весь потік даних в обох напрямах, щоб переправити його далі.

Міжмережеві екрани можуть бути інтегровані в маршрутизатори або виконані як окремі пристрої.

Основні схеми захисту на базі міжмережевих екранів:

– міжмережевий екран - фільтруючий маршрутизатор;

– міжмережевий екран на основі двопортового шлюзу;

– міжмережевий екран на основі екранованого шлюзу;

– міжмережевий екран - екранована підмережа.

Міжмережевий екран - фільтруючий маршрутизатор - найбільш простий в реалізації - виконує фільтрацію вхідних і витікаючих пакетів на основі аналізу їх адрес і портів (рис.6.5).

Міжмережевий екран на основі двопортового шлюзу прикладного рівня часто є комп'ютером з двома мережевими інтерфейсами, при передачі даних між якими здійснюється фільтрація (рис. 6.6).

Для забезпечення додаткового захисту між шлюзом прикладного рівня і Internet звичайно розміщують фільтруючий маршрутизатор. Між прикладним шлюзом і фільтруючим маршрутизатором утворюється внутрішня екранована

підмережа - демілітаризована зона, в якій можна розташовувати доступні ззовні інформаційні сервери.

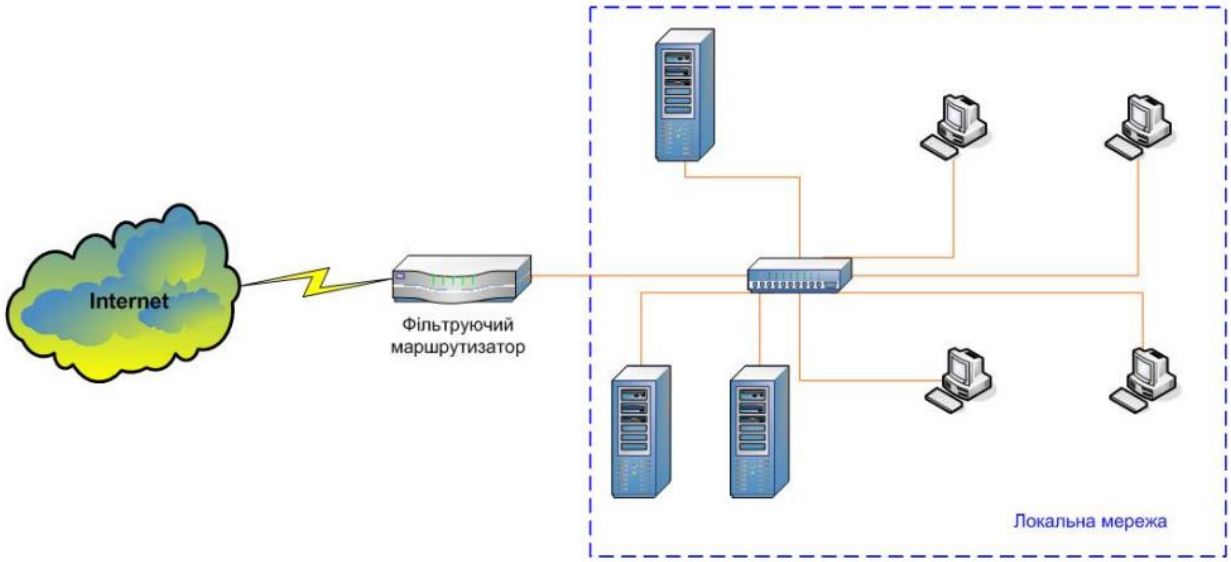

Рисунок 6.5– Міжмережевий екран на основі фільтруючого маршрутизатора

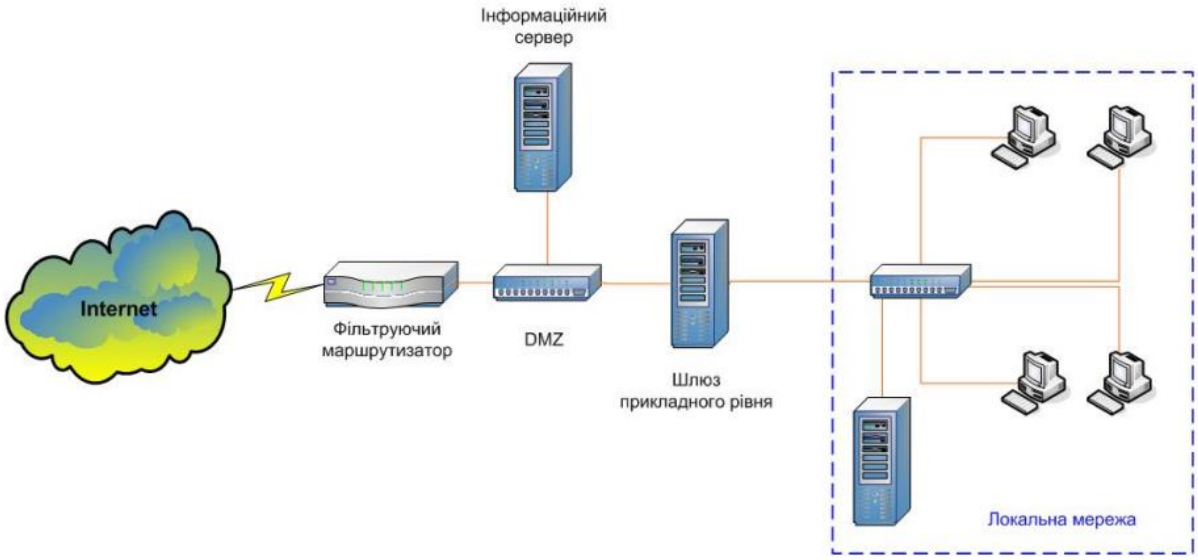

Рисунок 6.6– Міжмережевий екран з шлюзом і фільтруючим маршрутизатором

Міжмережевий екран на основі екранованого шлюзу прикладного рівня, реалізованого на комп'ютері лише з одним мережевим інтерфейсом, є гнучкіший варіант реалізації міжмережевого екрану, оскільки є можливість реалізувати з'єднання з Internet для деяких служб через проксі - сервер, а для деяких через фільтруючий маршрутизатор (рис.6.7).

Міжмережевий екран, що складається з екранованою зовнішнім і внутрішнім фільтруючими маршрутизаторами підмережею показаний на рис. 6.8.

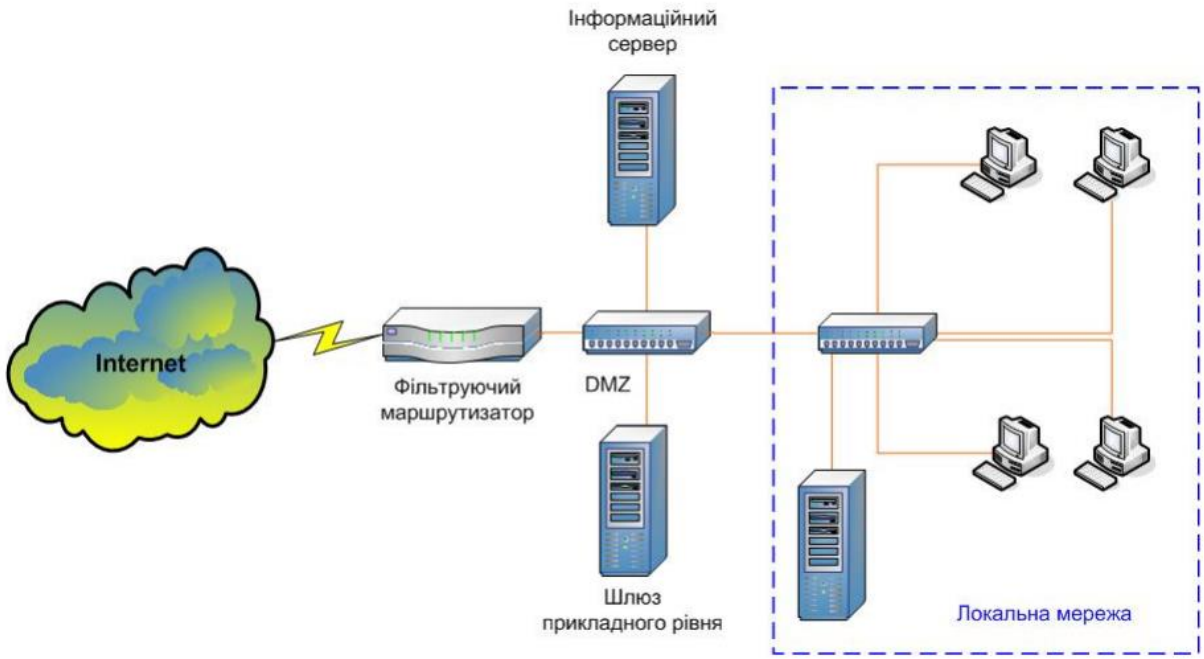

Рисунок 6.7– Міжмережевий екран з екранованим шлюзом і маршрутизатором

У цій підмережі розташовується шлюз прикладного рівня і сервери. Зовнішній маршрутизатор дозволяє трафік зовнішніх користувачів лише до проксі - серверу і серверам в демілітаризованій зоні. Внутрішній маршрутизатор захищає внутрішню мережу як від Internet, так і від екранованої підмережі (у разі її компрометації).

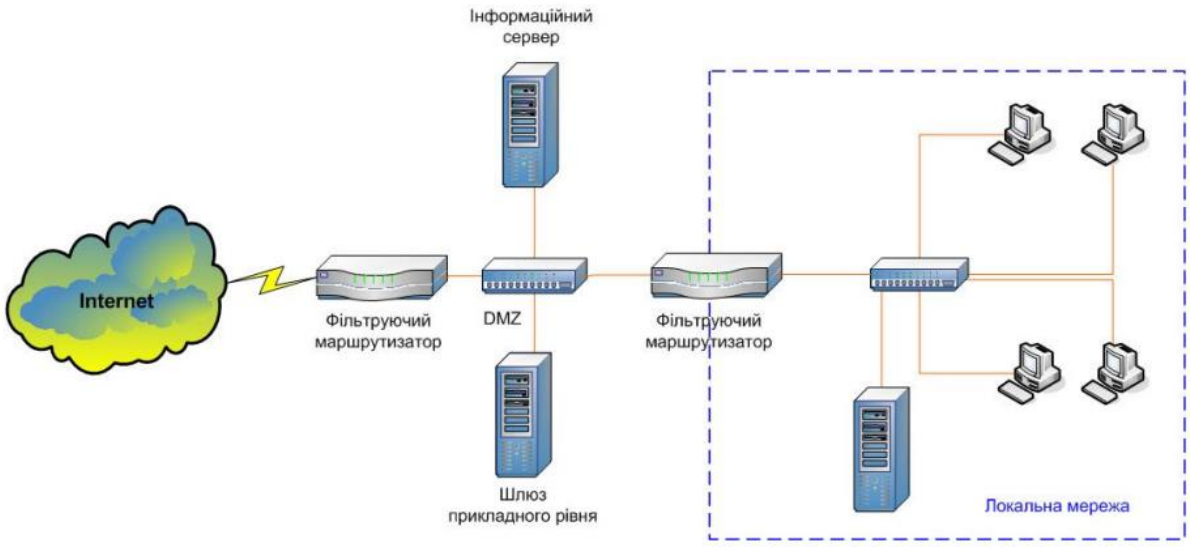

Рисунок 6.8– Міжмережевий екран з екранованою маршрутизторами підмережею

Ход работы

Для изучения материала по безопасным сетям нужно понимать модель OSI, стек проколов TCP/IP, а также ознакомиться с технологией VLAN. Для выполнения работы используется «Cisco packet tracer», возможно использование альтернативного имеющего весь необходимый функционал.

1 Постановка задачи

Существует некая фирма «СтройМех», занимающаяся продажей различного оборудования для производства. На данном этапе руководство решило серьёзно подойти к продвижению продукции посредством интернет-продаж. Итак, у нас существует производственно - технический и финансовый отделы и сеть для клиентов в одном офисе.

Требуется:

1) Создать структуру сети, подключить пользователей, выдать ip;
2) Трафик каждого отдела должен быть изолирован;
3) Настроить доступ к Web-серверу;
4) Клиенты должны иметь возможность подключения к сети и сайту.

2 Планирование узлов сети

Для начала нам нужно определиться со структурой сети. При проектировании сетей желательно придерживаться «иерархической модели»[9]. Согласно этой модели, сеть разбивается на три логических уровня:

Core layer (ядро сети) – высокопроизводительные устройства, главное назначение — быстрый транспорт.

Distribution layer (уровень распространения) – обеспечивает применение политик безопасности, QoS, агрегацию и маршрутизацию в VLAN, определяет широковещательные домены.

Access-layer (уровень доступа) – как правило, L2 свичи, назначение: подключение конечных устройств, маркирование трафика для QoS, защита от

колец в сети (STP) и широковещательных штормов, обеспечение питания для PoE устройств.

2.1 Иерархическая модель сети

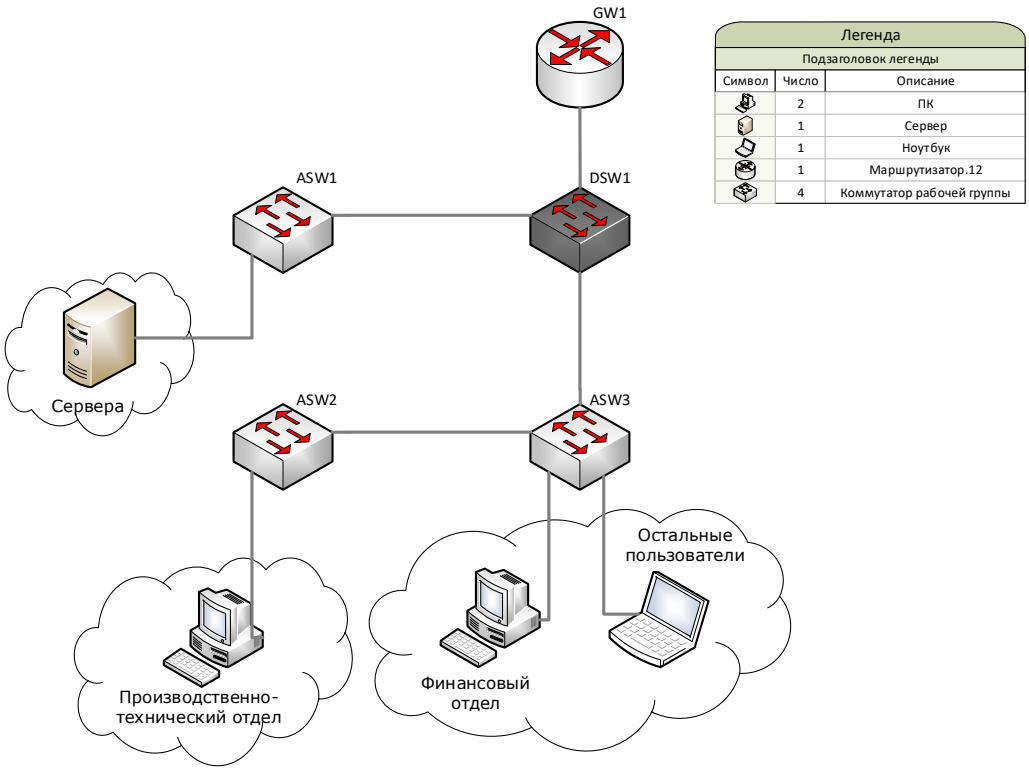

Рисунок 6.9– Модель сети

Реализовав структуру сети, мы распределили устройства соответственно их ролям и порядковому номеру:

gw – *Gateway*, шлюз сети;

dsw – *Distribution switch*, коммутатор распространения;

asw – *Access switch*, коммутатор доступа.

2.2 Подготовка списков VLAN

Для простоты демонстрации каждая группа будет выделена с отдельный VLAN, а также добавлен специальный VLAN для управления устройствами.

Номера VLAN никак не привязаны к реальным адресам и являются условными идентификаторами. Список VLAN представлен ниже:

Таблиця 6.2 – Список VLAN

№	Название	Примечание
1	default	Не используется
2	control	Для управления устройствами
3	servers	Для серверных машин
101	techDep	Производственно-технический отдел
102	finDep	Финансовый отдел
103	-	Зарезервирован
104	other	Другие пользователи, гости

После определения VLAN сетей можно определить IP-план. Учитывая, что сеть является небольшой имеет смысл использовать локальный диапазон адресов. Локальная сеть 192.168.0.0 /16 используется в качестве нашего диапазона адресов. Маска «/16» является обозначением «255.255.0.0». В частности, наложив маску на наш адрес мы получим адрес шлюза сети. Обычно это первый адрес.

2.3 Распределение адресов в подсетях

Для каждой подсети будет использоваться свой диапазон IP с маской «/24» (/24=255.255.255.0). На таблице изображён план распределения IP-адресов в сети, а также принадлежность к определённым VLAN.

2.4 Подключение оборудования по портам

При моделировании сети будем отталкиваться от того, что имеем: маршрутизатор cisco2811, коммутаторы cisco2960 и 2950. Все эти устройства полностью подходят для решения поставленных задач.

Таблиця 6.3 – IP-план сети

IP-адрес		**Примечание**	**VLAN**
Начальный	Конечный		
192.168.0.0 /16		Общая сеть	
192.168.0.0 /24		Серверные машины	3
192.168.0.1		Шлюз	
192.168.0.2		Веб-сервер	
192.168.0.3	192.168.0.254	Зарезервировано	
192.168.1.0 /24		control	2
192.168.1.1		Шлюз, адрес маршрутизатора	
192.168.1.2		Коммутатор распространения (dsw1)	
192.168.1.3		Коммутатор доступа (asw1)	
192.168.1.4		Коммутатор доступа (asw2)	
192.168.1.5		Коммутатор доступа (asw3)	
192.168.1.6	192.168.1.254	Зарезервировано	
192.168.101.0 /24		Произв.-техн. отдел	101
192.168.101.1		Шлюз	
192.168.101.2	192.168.101.254	Пул для пользователей	
192.168.102.0 /24		Финансовый отдел	102
192.168.102.1		Шлюз	
192.168.102.2	192.168.102.254	Пул для пользователей	
192.168.104.0 /24		Другие пользователи	104
192.168.104.1		Шлюз	
192.168.104.2	192.168.104.254	Пул для пользователей	

Таблиця 6.4 – Статус портов

Имя устройства	Порт	Примечание	Access	Trunk
Маршрутизатор (gw1)	FE 0/0	UpLink		
	FE 0/1	zoneA-dsw1		2,3,101,102,104
Коммутатор распространения (dsw1)	FE 0/24	zoneA-gw1		2,3,101,102,104
	GE 0/1	zoneA-asw3		2,101,102,104
	GE 0/2	zoneA-asw1		2,3
Коммутатор доступа (asw1)	GE 0/1	zoneA-dsw1		2,3
	FE 0/1-5	servers	3	
Коммутатор доступа (asw3)	GE 0/1	zoneA-dsw1		2,101,102,104
	GE 0/2	zoneA-asw2		2,101
	FE 0/1-10	finDep	102	
	FE 0/11-24	other	104	
Коммутатор доступа (asw2)	GE 0/1	zoneA-asw3		2,101
	FE 0/1-10	techDep	101	

2.5 Схемы сети

На основании этих данных можно составить все три схемы сети на этом этапе. Для этого можно воспользоваться «Microsoft Visio» или альтернативным редактором для построения схем сетей.

Вся сеть должна быть строго документирована: от принципиальной схемы, до имени интерфейса. Схемы сети L1, L2 в соответствии с уровнями модели OSI (Физический, канальный).

На схеме L1(рис. 6.10), мы отражаем физические устройства сети с номерами подключённых портов.

На схеме L2(рис. 6.11), мы указываем виртуальные сети, VLAN.

В больших сетях требуется указывать схему уровня L3[11]. Но учитывая, что наша сеть имеет всего один маршрутизатор использовать данную схему нет нужды.

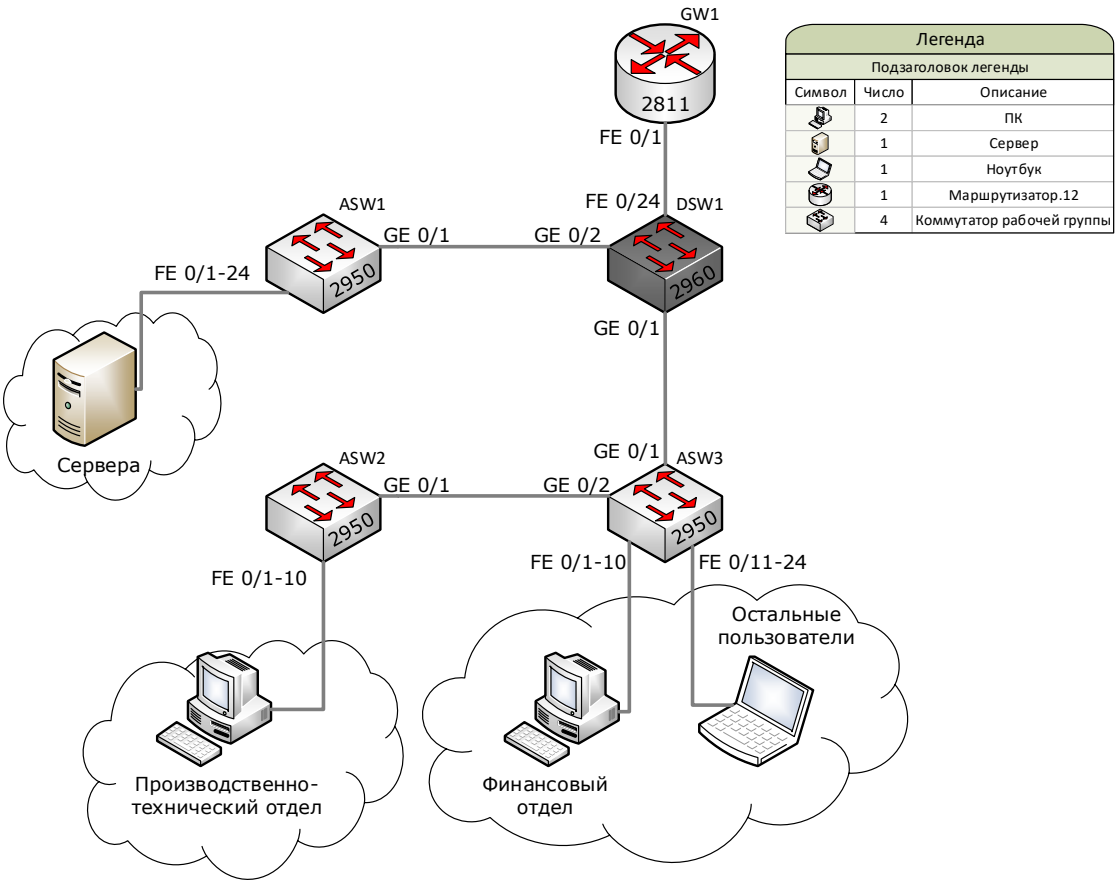

Рисунок 6.10– Схема физического уровня сети

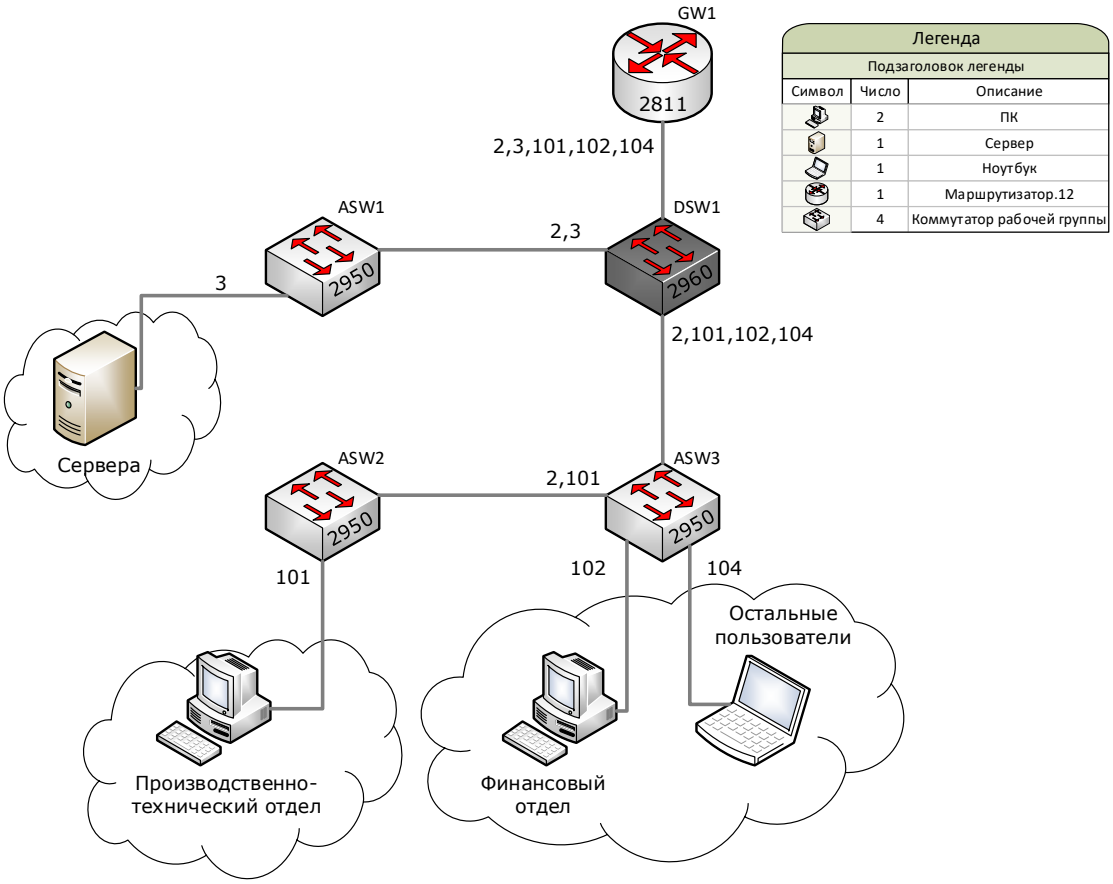

Рисунок 6.11– Схема канального уровня сети

3 Реализация структуры сети, подключение устройств

На данном этапе нам нужно создать и подключить все устройства, а также распределить ip-адреса по конечным клиентам.

Для начала создадим сеть из наших коммутаторов и маршрутизатора, соединив их портами из табл. 6.4,результат видим на рисунке 6.12.

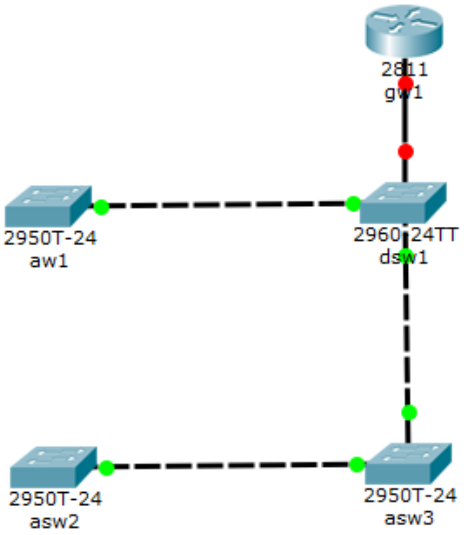

Рисунок 6.12– Создание структуры сети, шаг первый

После чего добавим в каждую подсеть несколько клиентов, таким образом, чтобы протестировать связь каждого узла. В результате создание сети закончено, следующим шагом нам нужно настроить конечные устройства. Отталкиваясь от созданной ранее таблицы и схемы уровня L1 зададим IP адрес каждой машины и шлюз.

Изменить IP-адрес можно разными способами. Один из них – это нажав на иконку компьютера. После чего откроется дополнительное окно, где нужно выбрать «Desktop», а потом «IP Configuration». В открывшемся окне запишите «IP Address» и «GateWay» в соответствии с табл. 6.3, как показано на рисунке 6.13.

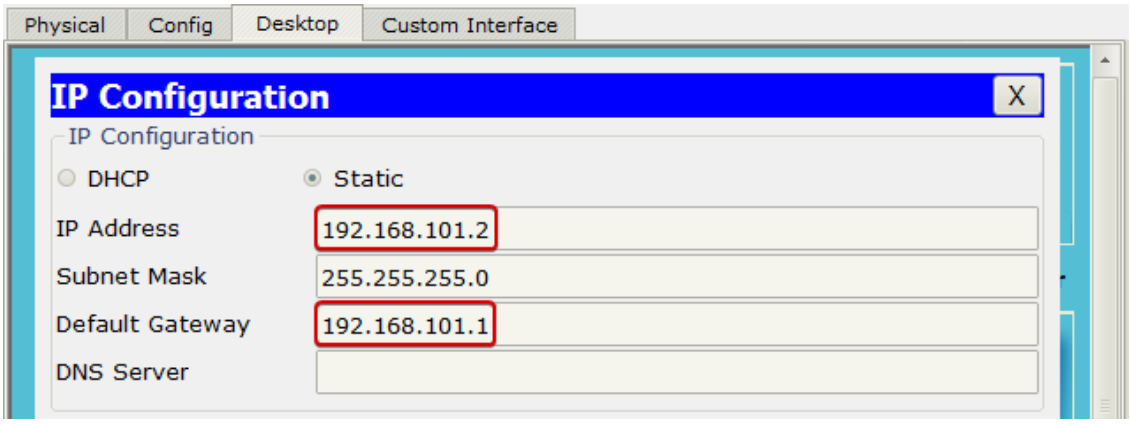

Рисунок 6.13

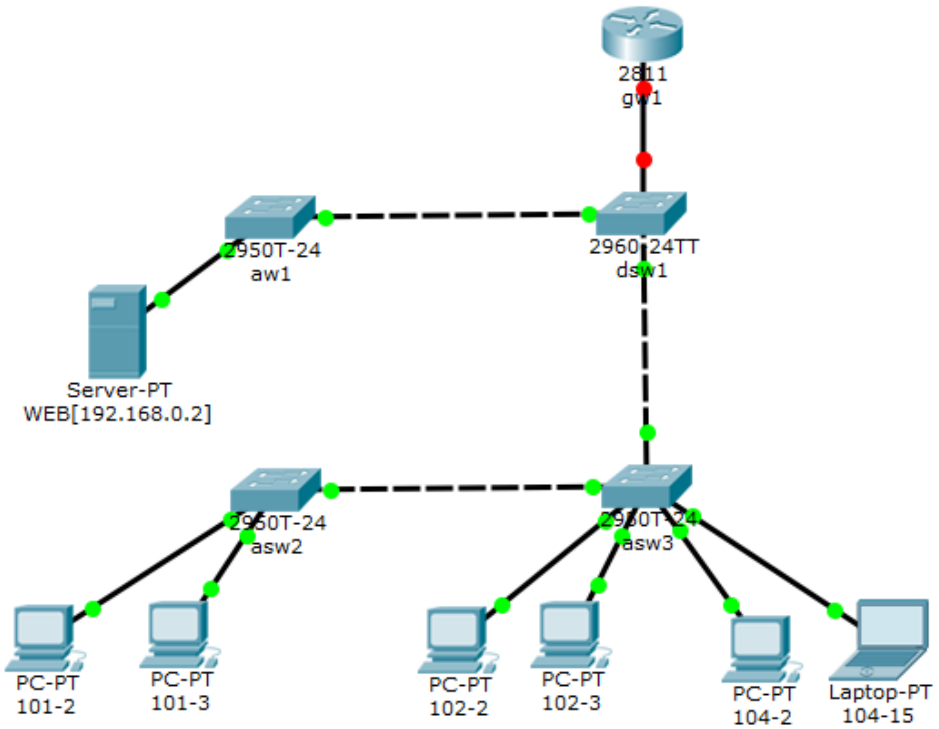

Рисунок 6.14– Схема сети

Для проверки работоспособности сети может оправить команду «ping» на одном из компьютеров, команда отобразит следующий результат:

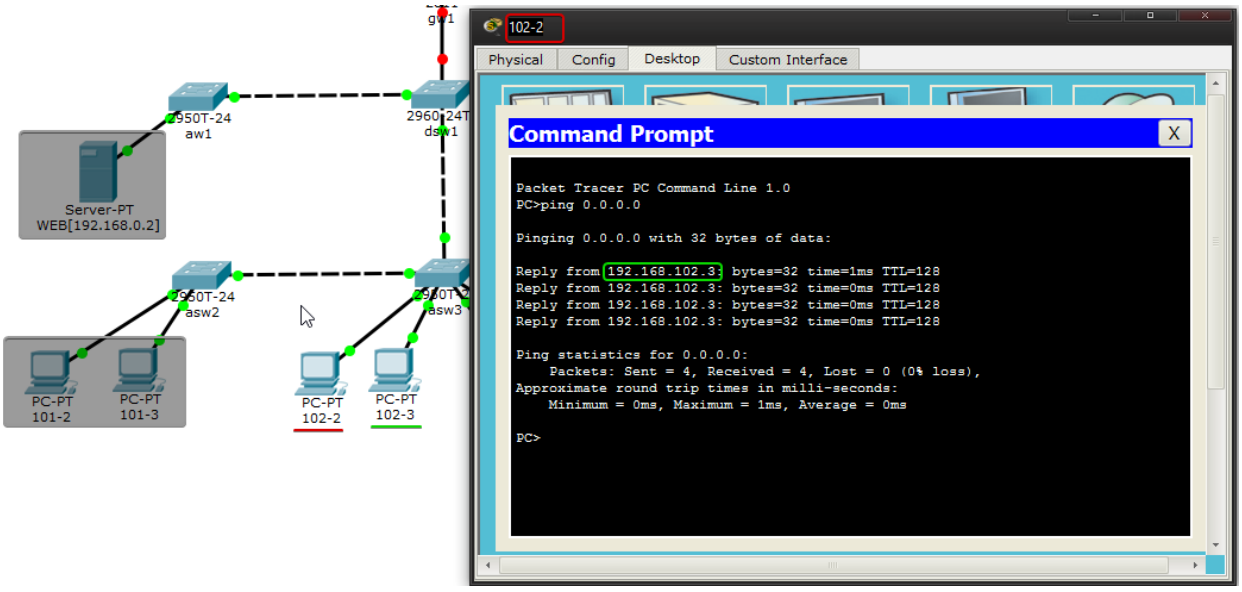

Рисунок 6.15

Можно заметить, что на запрос отреагировало только устройство находящееся в одной подсети, а именно 192.168.102.0 /24. Все остальные устройства, выделенные серым цветом, не ответили на запрос.

4 Конфигурация узлов сети

Первым делом нужно понимать принцип работы с коммутаторами в реальных системах. Коммутатор не имеет интерфейса, который предоставляет нам Cisco. Связь с коммутатором происходит посредством консольного управления. Обычно они выделены синим цветом, как показано на рис. 6.16.

Рисунок 6.16– Синим цветом показаны разъёмы под управляющий кабель

В Cisco мы можем эмулировать данное действие подключив консольный кабель к портам «Console» и «RS 232», как показано на рисунке 6.17. После

чего открыв компьютер и выбрав «Terminal» на рабочем столе. Подтвердив настройки мы увидим окно подключения к устройству.

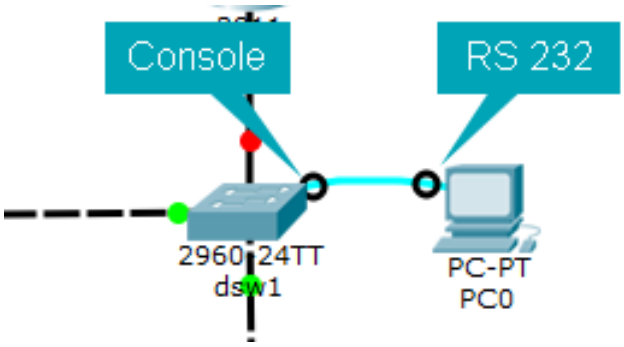

Рисунок 6.17– Консольное подключение

Для работы нужно зайти в привилегированный режим написав команду «enable» или «en», после чего перейти в режим настройки конфигураций «conf t» Множество команд возможно дописывать, используя клавишу TAB. Так же если неизвестна команда можно ввести символ «?» и нам покажут возможные команды на данном этапе[8].

Для быстрой работы в эмуляторе будет нелогичным использование консольного кабеля, намного проще использовать командный интерфейс(CLI) внутри самого устройства.

Так же требуется записать настройки в стартовую конфигурацию устройства. Для этого требуется ввести команду «write memory» / «wr m» в привилегированном режиме. В ином случае при перезагрузке устройства будет загружена прошлая конфигурация.

4.1 Добавление требуемых VLAN во все сегменты сети

Для разделения сети на сегменты нам нужно настроить базу VLAN на каждом из устройств. На схеме L2 отображены все используемые VLAN идентификаторы. Устройство должно содержать в своей базе список VLAN с которыми сталкивается по одному из каналов тогда.

Для примера коммутатор «asw3» подключён к таким VLAN каналам: 2, 101, 102 и 104. О канале 3 ему не известно. Чтобы активировать VLAN №2 нам требуется выбрать его и ввести новое имя:

```
Switch>
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 2
Switch(config-vlan)#name control
Switch(config-vlan)#exit
```

Таким же образом настраиваем все коммутаторы нашей сети.

Для настройки маршрутизатора используется тот же подход, но немного другие команды:

```
Router#vlan database
Router(vlan)#vlan 2 name control
VLAN 2 added:
    Name: control
Router(vlan)#vlan 3 name servers
VLAN 3 added:
    Name: servers
Router(vlan)#exit
APPLY completed.
Exiting....
```

4.2 Настройка коммутаторов

Для обеспечения работы всех сегментов сети требуется настроить работу VLAN на каждом узле сети.

Первым делом требуется предотвратить несанкционированное подключение через незадействованные порты. В данной части мы переведём все порта коммутаторов в выключенный режим. Отличие коммутатора и маршрутизатора в том, что у коммутатора все порты по умолчанию в состоянии прослушивания, а порты маршрутизатора выключены.

```
Switch>en
Switch#conf t
Switch(config)#int range gigabitEthernet 0/1-2
Switch(config-if-range)#shutdown
Switch(config-if-range)#exit
Switch(config)# int range fa 0/1-24
Switch(config-if-range)#sh
Switch(config-if-range)#ex
```

Команда «**interface** *номер*» – вход в режим конфигурирования интерфейса. Параметр «**range**» – означает, что следующие интерфейсы являются диапазоном. После выполнения заданных команд для всех устройств наша сеть будет иметь вид:

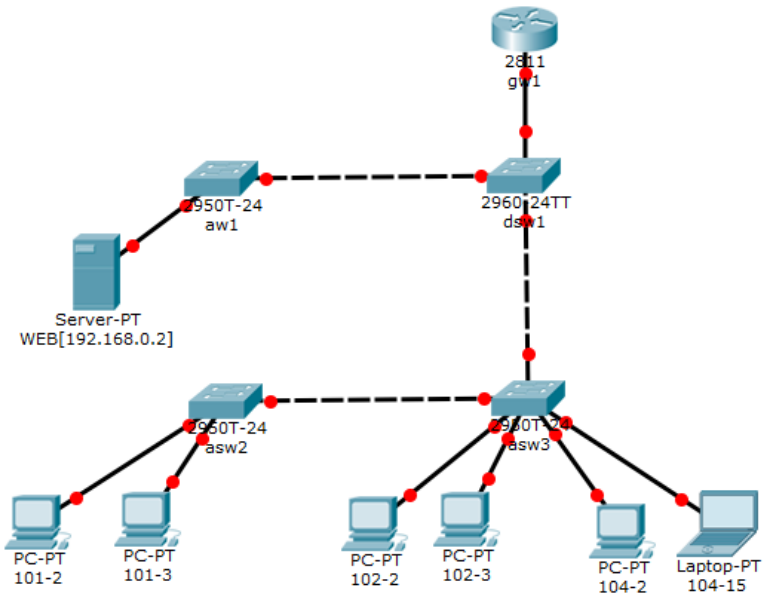

Рисунок 6.18– Сеть после отключения узлов

Вторая часть включает в себя настройку портов распространения (Trunk). В ней мы должны распределить доступ VLAN сегментов для передачи узел-шлюз.

Для примера указана настройка коммутатора «ASW3». В соответствии с таблицей 6.4 требуется указать порты «GE 0/1-2» как «Trunk», а «FE 0/1-24» как «Access».

```
Switch>en
Switch#conf t
Switch(config)#int gigabitEthernet 0/1
```

```
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 2,101-102,104
Switch(config-if)#no shutdown
Switch(config-if)#ex

Switch(config)#int gigabitEthernet 0/2
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 2,101
Switch(config-if)#no shutdown
Switch(config-if)#ex
Switch(config)#int range fa 0/1-10
Switch(config-if)# switchport mode access
Switch(config-if)#switchport access vlan 102
Switch(config-if)#no sh
Switch(config-if)#ex

Switch(config)#int range fa 0/11-24
Switch(config-if)# switchport mode access
Switch(config-if)#switchport access vlan 104
Switch(config-if)#no sh
Switch(config-if)#ex
```

Так же нам может понадобиться удалённый доступ к коммутатору по сети, а значит он должен иметь ip адрес. Давайте добавим адрес коммутаторов в соответствии с ip-планом сети (табл. 6.3).

```
Switch(config)#int vlan 2
Switch(config-if)#ip address 192.168.1.5 255.255.255.0
Switch(config-if)#ex
```

После чего мы должны сохранить изменения на коммутаторе:

```
Switch(config)#do write memor
```

На примере настройки коммутатора «ASW3» настроить коммутаторы «ASW1», «ASW2» и «DSW1». Чтобы проверить текущую настройку коммутатора нужно ввести команду «*show vlan brief*» в привилегированном режиме.

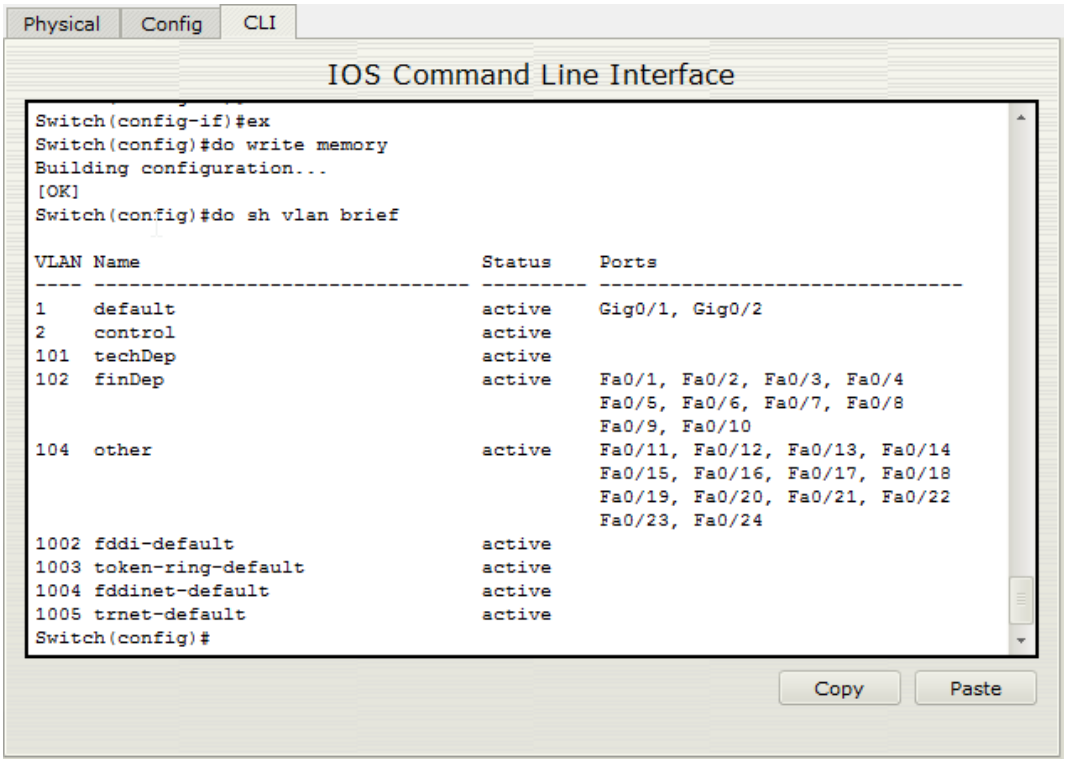

Рисунок 6.19– Состояние VLAN на коммутаторе asw3 после настройки

4.3 Конфигурация маршрутизатора

Последним шагом является настройка маршрутизатора, который в нашей сети выполняет функцию шлюза. Настройка маршрутизатора отличается от коммутаторов. Для начала нам не нужно выключать все порты, они выключены по умолчанию. Так же нам нужно настроить шлюз для каждого из сегментов сети VLAN.

Первым шагом мы включим порт подключённый к коммутатору «dsw1»:

```
Router(config)#interface fa 0/1
Router(config-if)#no shutdown
Router(config-if)#exit
```

Следующим шагом нужно используя тегирование кадра по стандарту «802.1q» прикрепить наши VLAN сегменты. Для этого нам нужно настроить виртуальные интерфейсы:

```
Router(config)#interface FastEthernet0/1.2
Router(config-subif)#description control
Router(config-subif)#encapsulation dot1Q 2
Router(config-subif)#ip address 192.168.1.1 255.255.255.0
Router(config-subif)#ex
```

5 Проверка работоспособности

Чтобы проверь работоспособность сети можно открыть «Web Browser» в компьютере и ввести адрес веб сервера, как изображено на рис. 6.20

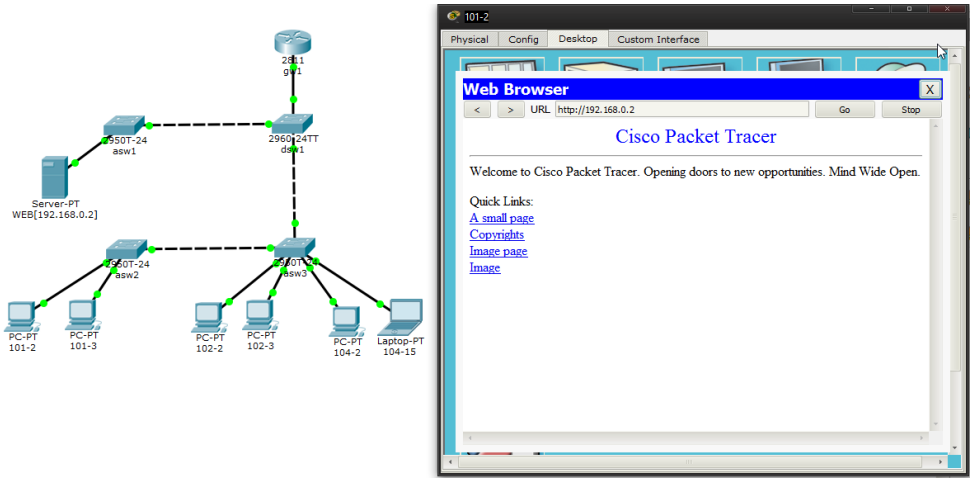

Рисунок 6.20– Отображение сайта на сервере WEB

Больше подробностей можно получить введя команду «show run», которая отобразит общую информацию о всей настройке (рис. 6.21).

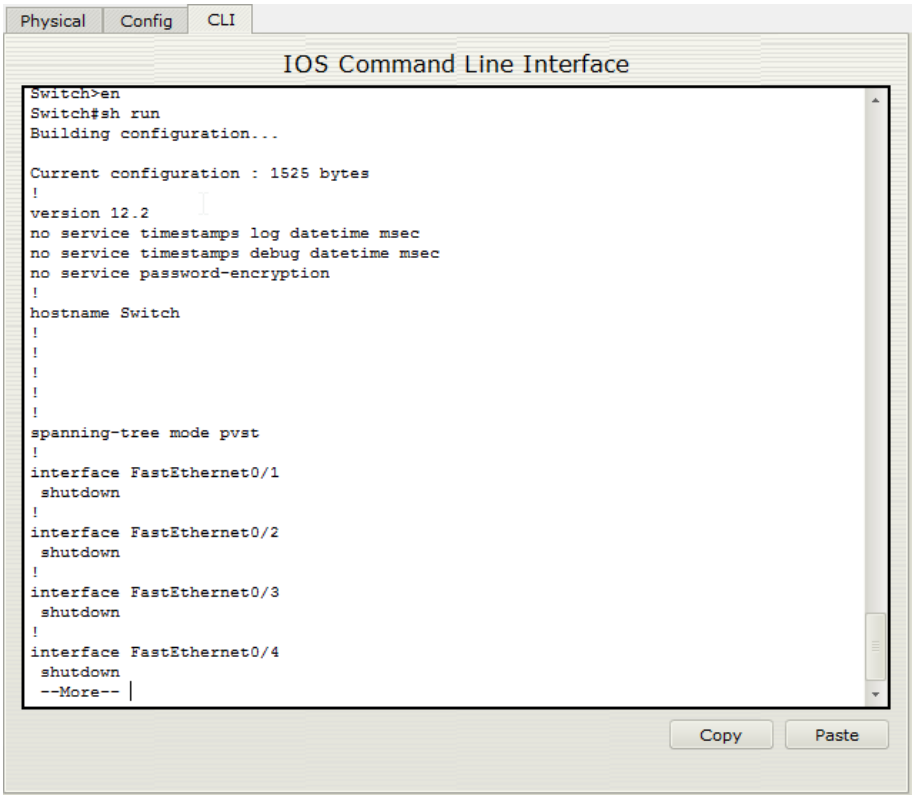

Рисунок 6.21– Результат выполнения «show run» на dsw1

Вопросы для самоконтроля

1) Що називають міжмережевими екранами? Які типи міжмережевих екранів Ви знаєте?

2) Опишіть принцип роботи фільтруючих маршрутизаторів, їх переваги і недоліки.

3) Опишіть принцип роботи систем трансляції адрес, їх переваги і недоліки.

4) Опишіть принцип роботи проксі - серверів, їх переваги і недоліки.

5) Опишіть основні схеми захисту мереж за допомогою міжмережевих екранів.

7 ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ СЕТЕВЫХ УСТРОЙСТВ

Цель: На практике освоить способы защиты маршрутизатора, подключение по SSH. Также использовать ACL списки, для коммуникации внутри сети..

Теоретические ведомости

1 Списки доступу

Списки доступу(ACL) – є послідовністю команд, що дозволяють або забороняють передачу пакетів через маршрутизатор. Команди стандартних IP списків доступу ухвалюють рішення про дозвіл або заборону пакетів винятково на підставі значення адреси відправника пакету. Списки доступу можуть використовуватися для управління розповсюдженням і прийомом пакетів з маршрутною інформацією, формування трафіку, визначення трафіку, який дозволить забезпечити необхідний захист. Причини застосування політики захисту на базі списків доступу можуть бути різні, наприклад, - необхідність запобігання зовнішнім атакам на комп'ютери локальної мережі, необхідність ізоляції трафіку між підрозділами компанії, необхідність розподілу навантаження на мережу і ін. При використанні списків доступу для організації міжмережевого екрану маршрутизатори зможуть обмежувати або запобігати доступу до комп'ютерів внутрішньої мережі із зовнішньої мережі, наприклад, Internet. Звичайно списку доступу такого типа розміщуються в маршрутизаторі, об'єднуючому такі мережі. При використанні списків доступу для організації ізоляції трафіку між підрозділами, список доступу звичайно розміщується на маршрутизаторі усередині локальної мережі.

1.1 Типы ACL

Стандартные (Standard): могут проверять только адреса источников.
Расширенные (Extended): могут проверять адреса источников, а также адреса получателей, в случае IP ещё тип протокола и TCP/UDP порты.

Обозначаются списки доступа либо номерами, либо символьными именами. ACL[10] также используются для разных сетевых протоколов. Мы в свою очередь будем работать с IP. Обозначаются они следующим образом, нумерованные списки доступа:

Стандартные: *от 1 до 99.*

Расширенные: *от 100 до 199.*

Список доступу звичайно містить достатньо багато рядків вказаного вище формату з метою регулювання трафіку, витікаючого з багатьох адрес. Кожен рядок списку доступу повинен містити однаковий ідентифікатор списку доступу. Списки доступу обробляються зверху "вниз", що означає, що спочатку інформація з пакету оцінюється першим рядком списку, потім другий і т.д. Маршрутизатор припиняє роботу із списком доступу після першої відповідності інформації з пакету параметрам команди, тому найзагальніші інструкції повинні бути розміщені на початку списку.

2 Структура ACL

2.1 Стандартный список доступа

---

access-list <номер списка от 1 до 99> {**permit** | **deny** | **remark**} {**address** | **any** | **host**} [source-wildcard] **[log]**

где: `permit`: разрешить;

`deny`: запретить;

`remark`: комментарий о списке доступа;

`address`: запрещаем или разрешаем сеть;

`any`: разрешаем или запрещаем всё;

`host`: разрешаем или запрещаем хосту;

`source-wildcard`: WildCard маска сети;

`log`: включаем логгирование пакеты проходящие через запись ACL.

2.2 Расширенный список доступа

```
access-list <номер списка 100 - 199> {permit | deny | remark} protocol source [source-wildcard] [operator operand] [port <порт или название протокола> [established]
```

где: `protocol source`: протокол будем разрешать/закрывать (TCP, IP и т.д)

`operator`:

`A.B.C.D` – адрес получателя;

`any` – любой конечный хост;

`eq` – только пакеты на этом порте;

`gt` – только пакеты с большим номером порта;

`host` – единственный конечный хост;

`lt` – только пакеты с более низким номером порта;

`neq` – только пакеты не на данном номере порта;

`range` – диапазон портов.

`port`: номер порта (TCP или UDP), можно указать имя;

`established`: разрешаем прохождение TCP-сегментов, которые являются частью уже созданной TCP-сессии.

Існує одна особливість використання списків доступу. Після створення списку доступу, маршрутизатор Cisco відхилятиме будь-які пакети з адресами відправника, не вказаними ні в одній команді списку.

Списки доступу створюються в глобальному режимі конфігурації маршрутизатора. Всі стандартні списки доступу IP повинні бути пронумеровані в діапазоні 1-99, ми використовуватимемо #1. Припустимо, що необхідно дозволити трафік від адреси 192.168.101.13 і заборонити решту трафіку. Процедура конфігурації включає наступні команди:

```
1 Router>
2 Router>en
3 Router#conf t
4 Router(config)# access-list 1 permit 192.168.101.13
```

Оскільки маска адреси відправника не вказана, маршрутизатор використовує значення за умовчанням «0.0.0.0» (що означає точну відповідність адресі). Також маршрутизатор автоматично відхиляє всі пакети з адресами відправника іншими, чим «192.168.101.13».

3 Подключение к интерфейсу

Перш, ніж список доступу розпочне працювати, його необхідно прив'язати до певного інтерфейсу. Команда, що виконує цю функцію, має вигляд:

---

ip access-group <номер списка или имя ACL> {**in** | **out**}

где: `in` – входящее направление;

`out` – исходящее направление.

Списки доступу можуть бути прив'язані до інтерфейсу маршрутизатора, як для витікаючих, так і для вхідних пакетів. У разі прив'язки для вхідних пакетів, маршрутизатор аналізує адресу відправника кожного вхідного в інтерфейс пакету на відповідність командам списку. Якщо команда явно або неявно "дозволяє" пакет, він маршрутизується і передається у напрямку до мережі одержувача, якщо ж команди списку доступу явно або неявно "забороняють" пакет, пакет відкидається. У разі прив'язки для витікаючих пакетів, маршрутизатор аналізує адресу відправника кожного витікаючого з інтерфейсу пакету на відповідність командам списку і виконує аналогічні дії

Для прив'язки створеного списку доступу до інтерфейсу маршрутизатора FE0/0 для вхідних пакетів необхідно виконати наступні команди:

```
1 Router(config)#access-list 1 permit 192.168.101.13
2 Router(config)#int fa 0/0
3 Router(config-if)#ip access-group 1 in
```

Створимо список доступу #2, з наступними критеріями: вирішуються всі пакети з мережі 192.168.3.0 255.255.255.128, але забороняються всі пакети

з мережі 192.168.3.128 255.255.255.128, також забороняються всі пакети з мережі 192.168.104.0, окрім пакетів комп'ютера 192.168.104.2, решта трафіку дозволена. Процедура конфігурації включає наступні команди:

```
1 Router(config) #access-list 2 deny 192.168.3.128 0.0.0.127
2 Router(config) #access-list 2 permit 192.168.104.5
3 Router(config) #access-list 2 deny 192.168.104.0 0.0.0.255
4 Router (config) #access-list 2 permit any
```

4 Виды ACL

4.1 Динамический (Dynamic ACL)

Позволяют сделать следующее, например у вас есть маршрутизатор, который подключен к какому-то серверу и нам нужно закрыть доступ к нему из внешнего мира, но в тоже время есть несколько человек, которые могут подключаться к серверу.

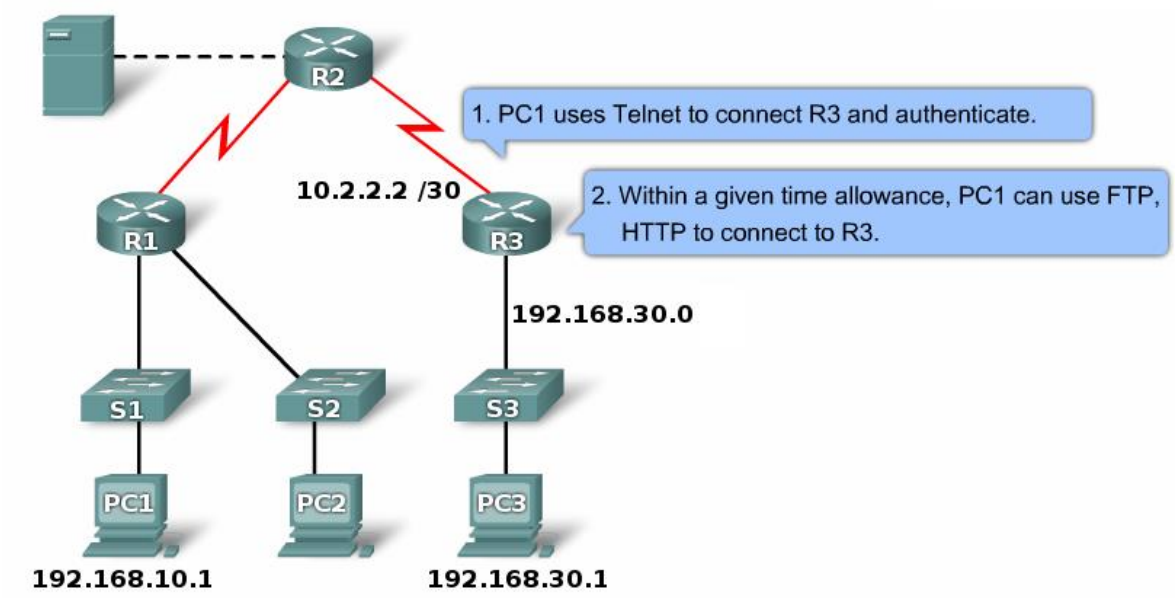

Рисунок 7.1– Схема динамического доступа

Мы настраиваем динамический список доступа, прикрепляем его на входящем направлении, а дальше людям, которым нужно подключиться, подключаться через Telnet к данному устройству, в результате динамический ACL открывает проход к серверу, и уже человек может зайти скажем через HTTP попасть на сервер. По умолчанию через 10 минут этот проход

закрывается и пользователь вынужден ещё раз выполнить Telnet чтобы подключиться к устройству.

Создаем пользователей для подключения через Telnet.

```
1 R3(config)#username Student password 0 cisco
```

Разрешаем подключаться к серверу по Telnet всем узлам.

```
2 R3(config)#access-list 101 permit tcp any host 10.2.2.2 eq telnet
3 R3(config)#access-list 101 dynamic testlist timeout 15 permit ip
  192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255
```

Закрепляем 101 ACL за интерфейсом в входящем направлении.

```
4 R3(config)#interface serial 0/0/1
5 R3(config-if)#ip access-group 101 in
```

Как только пользователь аутентифицируеться, сеть 192.168.30.0 будет доступна, через 5 минут бездействия сеанс закроется.

```
6 R3(config)#line vty 0 4
7 R3(config-line)#login local
8 R3(config-line)#autocommand access-enable host timeout 5
```

4.2 Рефлексивный (Reflexive ACL)

Здесь ситуация немножко отличается, когда узел в локальной сети отправляет TCP запрос в Интернет, у нас должен быть открытый проход, чтобы пришел TCP ответ для установки соединения. Если прохода не будет – мы не сможем установить соединение, и вот этим проходом могут воспользоваться злоумышленники, например проникнуть в сеть. Рефлексивные ACL работают таким образом, блокируется полностью доступ (deny any) но формируется ещё один специальный ACL, который может читать параметры пользовательских сессий, которые сгенерированны из локальной сети и для них открывать проход в deny any, в результате получается что из Интернета не смогут установить соединение. А на сессии сгенерированны из локальной сети будут приходить ответы.

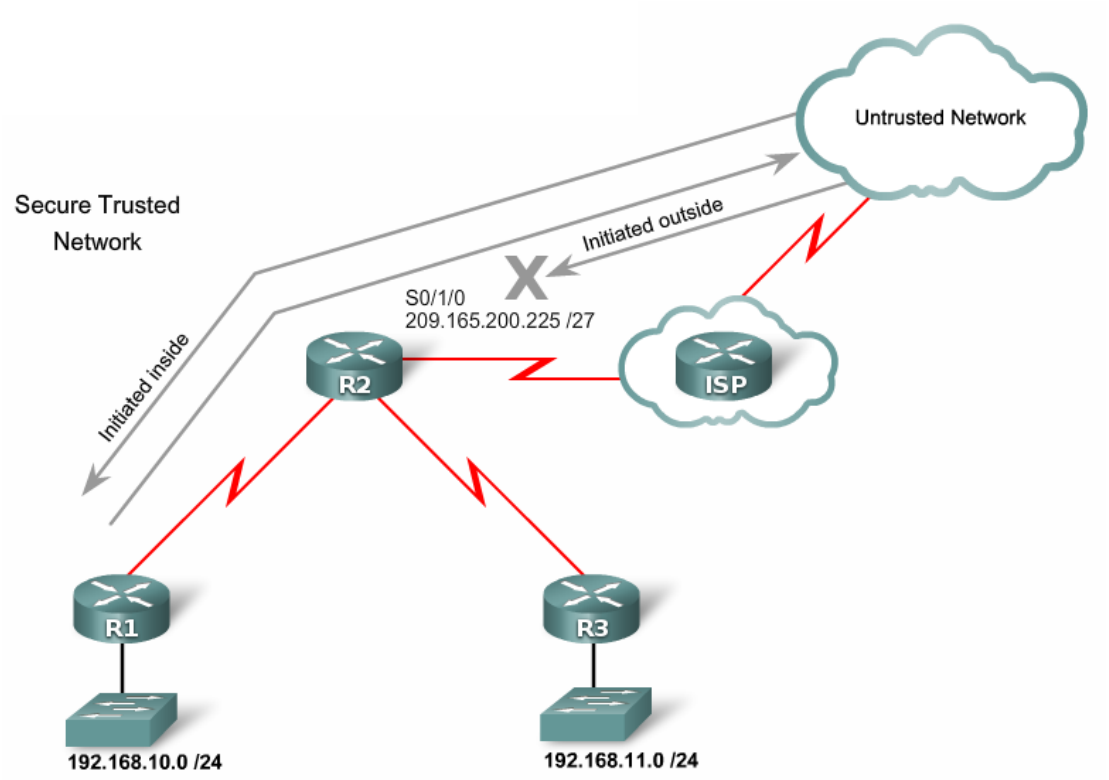

Рисунок 7.2– Схема рефлексивного доступа

Заставляем маршрутизатор отслеживать трафик, который инициировался изнутри.

```
1 R2(config)#ip access-list extended OUTBOUNDFILTERS
2 R2(config-ext-nacl)#permit tcp 192.168.0.0 0.0.255.255 any reflect
  TCPTRAFFIC
3 R2(config-ext-nacl)#permit icmp 192.168.0.0 0.0.255.255 any reflect
  ICMPTRAFFIC
```

Создаем входящую политику, которая требует, чтобы маршрутизатор проверял входящий трафик, чтобы видеть инициировался ли изнутри и связываем TCPTRAFFIC к INBOUNDFILTERS.

```
4 R2(config)#ip access-list extended INBOUNDFILTERS
5 R2(config-ext-nacl)#evaluate TCPTRAFFIC
6 R2(config-ext-nacl)#evaluate ICMPTRAFFIC
```

Применяем входящий и исходящий ACL на интерфейс.

```
7 R2(config)#interface serial 0/1/0
8 R2(config-if)#ip access-group INBOUNDFILTERS in
9 R2(config-if)#ip access-group OUTBOUNDFILTERS out
```

4.3 Ограничение по времени (Time-based ACL)

Обычный ACL, но с ограничением по времени, вы можете ввести специальное расписание, которое активирует ту или иную запись списка доступа. И сделать такой фокус, например пишем список доступа, в котором запрещаем HTTP-доступ в течении рабочего дня и вешаем его на интерфейс маршрутизатора, то есть, сотрудники предприятия пришли на работу, им закрывается HTTP-доступ, рабочий день закончился, HTTP-доступ открывается, пожалуйста, если хотите – сидите в Интернете.

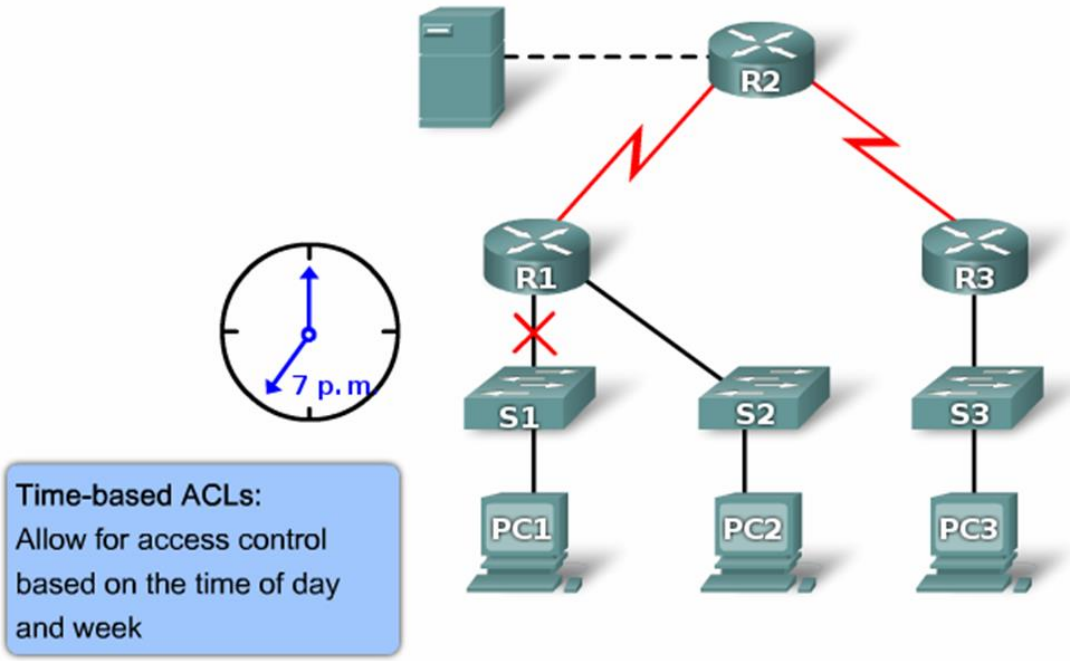

Рисунок 7.3– Схема доступа по времени

Создаем список времени, в котором добавляем дни недели и время.

```
1 R1(config)#time-range EVERYOTHERDAY
2 R1(config-time-range)#periodic Monday Wednesday Friday 8:00 to
  17:00
```

Применяем time-range к ACL.

```
3 R1(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255 any eq
  telnet time-range EVERYOTHERDAY
```

Закрепляем ACL за интерфейсом

```
4 R1(config)#interface s0/0/0
5 R1(config-if)#ip access-group 101 out
```

5 Отображение конфигураций

Информацию о списках доступа можно получить используя команду:

---

show access-lists [ACL номер | имя]

Где вызов без параметров отображает *все списки доступа*, а указав в дополнительный параметр номер или имя списка получим информацию исключительно данного списка.

Для примера отобразим текущие списки на маршрутизаторе:

```
Router#show access-lists
Extended IP access list TEST
Standard IP access list 1
     10 permit host 1.1.1.1
Standard IP access list 2
     10 deny 192.168.3.128 0.0.0.127
     20 permit host 192.168.104.5
     30 deny 192.168.104.0 0.0.0.255
Router#
```

Задания

Используя сеть, созданную на прошлом практическом занятии, настроим основные меры безопасности и заблокируем часть трафика используя ACL[7, 12].

1 Настройка мер безопасности

1) Выберите более надёжные пароли для коммутатора:

   a) Правила хорошего тона рекомендуют первым делом изменить имя маршрутизатора.

```
Router#hostname R1
```

   b) Чтобы соблюсти рекомендации, измените зашифрованный пароль привилегированного режима.

```
R1(config)#enable secret PASSWORD123
```

c) Укажите, что пароль должен включать не менее десяти символов.

```
3 R1(config)# security passwords min-length 10
```

2) Активируйте подключения SSH:

a) В качестве имени домена укажите любой сайт site.com.

```
4 R1(config)#ip domain-name site.com
```

b) Создайте в базе данных локальных пользователей запись, которая будет использоваться при подключении к маршрутизатору через SSH. Пароль должен соответствовать стандартам надёжных паролей, а пользователь — иметь права доступа уровня администратора.

```
5 R1(config)#username admin privilege 15 secret ADMIN!@123
```

c) Настройте транспортный ввод для vty-линий таким образом, чтобы они могли принимать подключения SSH, но не разрешайте подключения Telnet.

```
6 R1(config)#line vty 0 4
7 R1(config-line)#transport input ssh
```

d) Для проверки подлинности в vty-линиях должна использоваться база данных локальных пользователей.

```
8 R1(config-line)#login local
9 R1(config-line)#exit
```

e) Создайте ключ шифрования RSA с длиной 2048 бит.

```
10 R1(config)#crypto key generate rsa
   The name for the keys will be: R1.site.com
   Choose the size of the key modulus in the range of 360 to 2048 for
     your General Purpose Keys. Choosing a key modulus greater than
     512 may take a few minutes.

11 How many bits in the modulus [512]: 2048
   % Generating 2048 bit RSA keys, keys will be non-exportable...[OK]

12 R1(config)#
```

3) Обеспечьте защиту консоли и vty-линий:

a) Завершение сеанса подключения, при пяти минут неактивности.

```
13 R1(config)# line console 0
14 R1(config-line)# exec-timeout 5 0
15 R1(config-line)# line vty 0 4
16 R1(config-line)# exec-timeout 5 0
17 R1(config-line)# exit
18 R1(config)#
```

b) Обеспечение блокировка от метода перебора паролей.

```
19 R1(config)# login block-for 30 attempts 2 within 120
```

4) Сохранение конфигурации маршрутизатора

```
20 R1(config)# do write memory
```

5) Проверка доступа с помощью протокола SSH к маршрутизатору «GW1». Для этого зайдите на один из компьютеров, откройте терминал и введите:

```
1 ssh -l admin 192.168.1.1
```

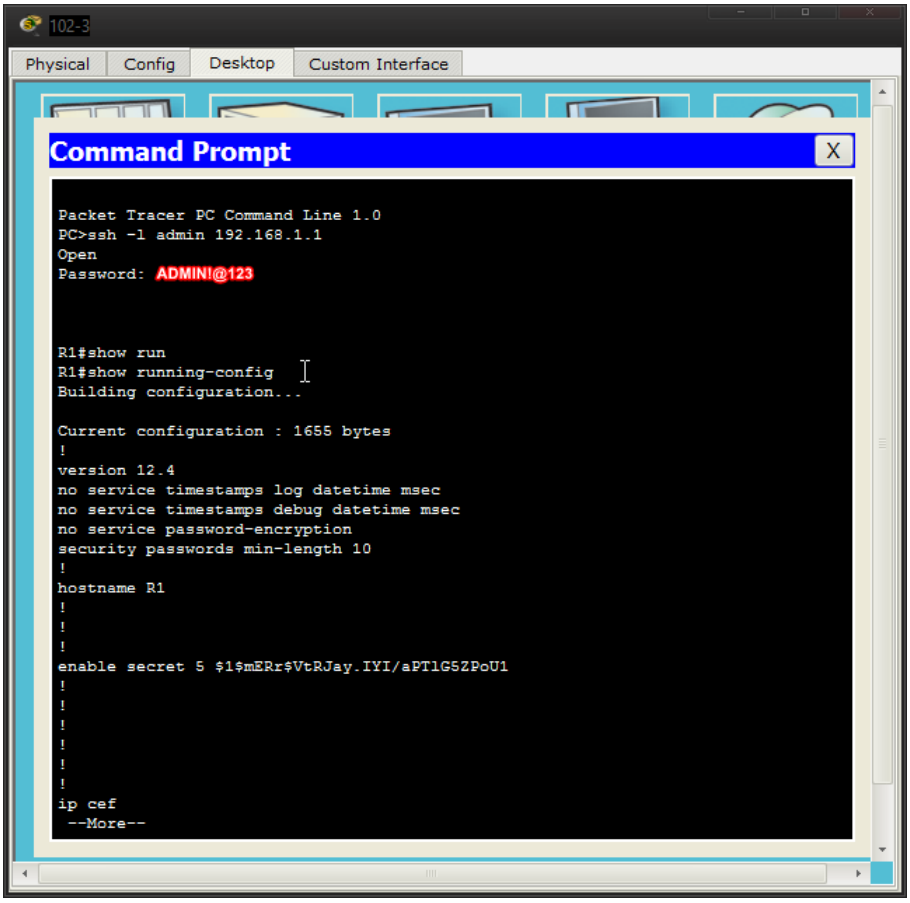

Рисунок 7.4– Результат удалённого выполнения команды «show run»

2 Конфигурация списков доступа

После настройки безопасной требуется разделить доступ пользователей к различным участкам сети:

1) Найдите свой вариант в таблице 7.1.

2) Реализуйте доступ к VLAN подсетям указанными видами ACL.

3) Ограничьте работу(доступ) к сайту в неположенной время, в соответствии с вариантом.

В отчёте нужно продемонстрировать передачу или блокировку пакетов по сети. Используя команду «show», для отображения информации различного характера добавить в отчёт листинги команд: «show access-lists», сокращённый вариант «show running-config», «show vlan».

Вопросы для самоконтроля

1) Що є списки доступу, використовувані маршрутизаторами Cisco? Для чого вони використовуються?

2) Приведіть формат стандартного IP списку доступу і опишіть використання його параметрів.

3) Назвіть особливості використання маски підмережі відправника для вказівки діапазонів адрес комп'ютерів в командах списків доступу.

4) Опишіть алгоритм перевірки пакетів командами списків доступу.

5) Назвіть послідовність і команди конфігурації списків доступу маршрутизаторів Cisco.

Таблиця 7.1 – Варианты задания

№	VLAN сегмент сети			Время работы сайта	
	techDep [101]	finDep [102]	other [104]	от	до
1	Time-based	Dynamic		8	15
2	Dynamic		Time-based	14	21
3	Reflexive	Dynamic		7	13
4		Dynamic	Time-based	19	22
5	Dynamic		Reflexive	16	21
6	Dynamic	Reflexive		0	12
7	Reflexive	Time-based		5	16
8	Time-based		Reflexive	16	24
9	Dynamic		Reflexive	15	19
10		Dynamic	Reflexive	4	15
11	Time-based		Dynamic	8	19
12		Dynamic	Time-based	8	11
13		Dynamic	Reflexive	3	9
14	Time-based	Dynamic		14	19
15	Dynamic	Reflexive		20	22
16	Time-based	Reflexive		16	24
17		Time-based	Dynamic	4	14
18	Dynamic	Time-based		12	24
19		Reflexive	Dynamic	1	3
20		Time-based	Reflexive	9	13
21		Time-based	Dynamic	1	19
22	Reflexive		Time-based	3	20
23	Reflexive	Dynamic		12	14
24		Reflexive	Time-based	2	5
25		Reflexive	Dynamic	17	19
26	Time-based		Dynamic	19	22
27	Reflexive		Dynamic	19	21
28	Dynamic	Time-based		7	22
29	Dynamic		Time-based	18	19
30	Reflexive		Dynamic	3	12

8 АНАЛИЗ ДАННЫХ С ПОМОЩЬЮ WIRESHARK

Цель: На практике применить знания анализа сетевых устройств. Проанализировать трафик используя предоставленное ПО. Продемонстрировать незащищённость протокола FTP.

Теоретические ведомости

1 Анализ трафика

Sniffer (от англ. to sniff – нюхать) – это сетевой анализатор трафика, программа или программно-аппаратное устройство, предназначенное для перехвата и последующего анализа, либо только анализа сетевого трафика, предназначенного для других узлов.

Перехват трафика может осуществляться:

1) обычным «прослушиванием» сетевого интерфейса (метод эффективен при использовании в сегменте концентраторов (хабов) вместо коммутаторов (свичей), в противном случае метод малоэффективен, поскольку на сниффер попадают лишь отдельные фреймы);

2) подключением сниффера в разрыв канала;

3) ответвлением (программным или аппаратным) трафика и направлением его копии на сниффер;

4) через анализ побочных электромагнитных излучений и восстановление таким образом прослушиваемого трафика;

5) через атаку на канальном (2-й) или сетевом (3-й) уровне, приводящую к перенаправлению трафика жертвы или всего трафика сегмента на сниффер с последующим возвращением трафика в надлежащий адрес.

В начале 1990-х широко применялся хакерами для захвата пользовательских логинов и паролей. Широкое распространение хабов позволяло захватывать трафик без больших усилий в больших сегментах сети.

Снифферы применяются как в благих, так и в деструктивных целях. Анализ прошедшего через сниффер трафика, позволяет:

- Отслеживать сетевую активность приложений.
- Отлаживать протоколы сетевых приложений.
- Локализовать неисправность или ошибку конфигурации.
- Обнаружить паразитный, вирусный и закольцованный трафик, наличие которого увеличивает нагрузку сетевого оборудования и каналов связи.
- Выявить в сети вредоносное и несанкционированное ПО, например, сетевые сканеры, флудеры, троянские программы, клиенты пиринговых сетей и другие.
- Перехватить любой незашифрованный (а порой и зашифрованный) пользовательский трафик с целью узнавания паролей и другой информации.

1.1 Базовый принцип работы снифферов

Давайте рассмотрим с вами рис. 8.1. На нем изображена схематично структура сетевой подсистемы ОС. Вся базовая инфраструктура реализована в виде драйверов и работает в режиме ядра. Пользовательские процессы и реализации прикладных протоколов, в частности интерфейс сниффера работают в пользовательском режиме. На рисунке отображены 2 пользовательских процесса («сетевой процесс 1» и «сетевой процесс 2»).

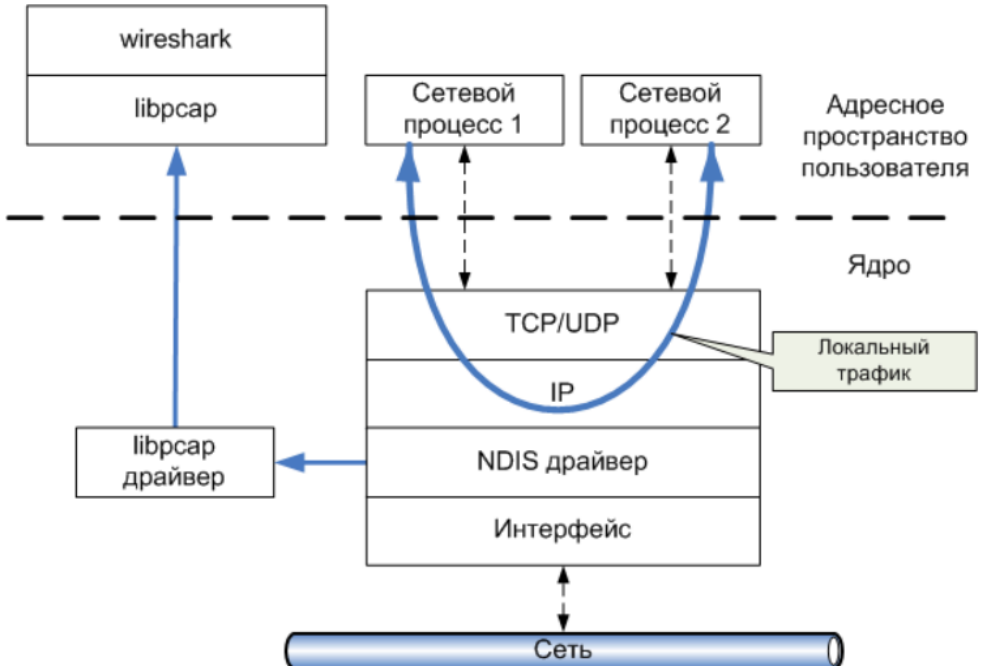

Рисунок 8.1– Принцип «захвата» сниффером сетевого трафика

Сниффер использует библиотеку в режиме «захвата» пакетов, т.е. может получать копию всех данных проходящих через драйвер сетевого интерфейса. Сами данные не изменяются.

Основной нюанс использования сниффера заключается в том, что он не позволяет производить анализ локального трафика, т.к. он не проходит через драйвер сетевого устройства (см. рис. 8.1.). Т.е., если вы захотите проанализировать сниффером трафик между 2-ми сетевыми процессами на локальной машине (например, ftp-сервер и ftp-клиент), то у вас ждет разочарование. Однако, например при использовании виртуальных машин, сниффер будет работать без проблем, т.к. виртуальные машины эмулируют реальную среду и сетевые адаптеры, поэтому трафик идет через драйвера как и в нормальной ситуации при взаимодействии с другими физическими сетевыми машинами.

Также к недостаткам большинства снифферов стоит отнести и тот факт, что, позволяя анализировать трафик, проходящий через сетевой интерфейс, они не могут указать, какое именно приложение генерирует или получает его. Это объясняется те, что информация об этом хранится на сетевом (например, IP) уровне сетевого стека, а большинство снифферов использует собственную реализацию стека протоколов (например, библиотеку WinPcap), которая (как уже было показано) работает непосредственно с драйверами устройств.

Также, снифферы вносят дополнительную нагрузку на процессор, т.к. могут обрабатывать достаточно объемный сетевой трафик, в особенности для высокоскоростных соединений (Fast Ethernet, Gigabit Ethernet и др.).

2 Программа Wireshark

Данный сниффер позволяет в режиме реального времени захватывать пакеты из сети, и анализировать их структуру. Также можно анализировать структуру пакетов из файла, содержащего трафик.

Первым делом нам требуется выбрать сетевой интерфейс для захвата трафика из сети. Правый график показывает активность интерфейсов. В данном случае весь трафик идёт через «NAT», как изображено на рис. 8.2.

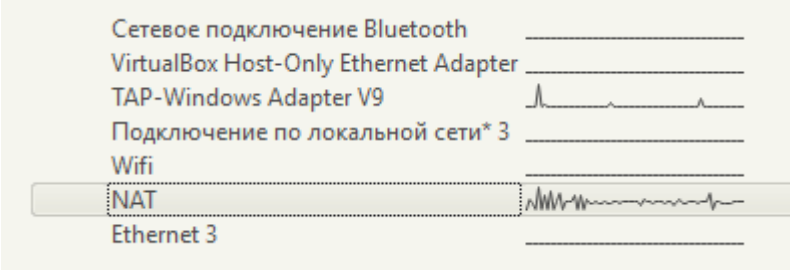

Рисунок 8.2– Список доступных интерфейсов

После выбора определённого интерфейса откроется основное окно программы, изображено на рис. 8.3. В стандартном режиме окно сниффера делится на 3 фрейма (панели): список захваченных пакетов, «анализатор» протоколов и исходные данные пакетов. Размер каждого фрейма можно менять по своему усмотрению.

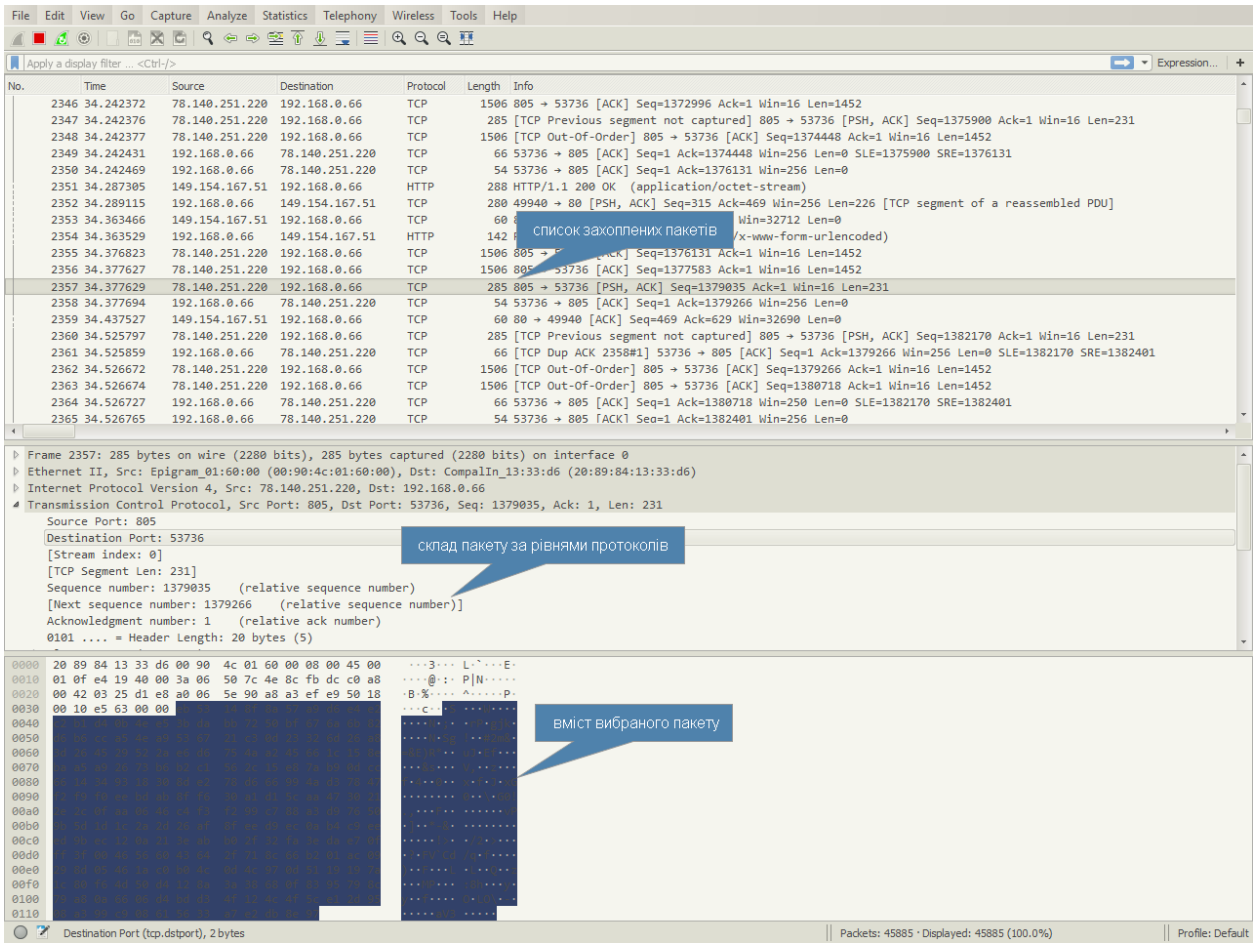

Рисунок 8.3– Главное окно программы

Рассмотрим эти панели подробнее:

Верхняя панель содержит список пакетов, захваченных из сети. Список можно отсортировать по любому полю – для этого нажать на заголовок соответствующего поля. Каждая строка содержит следующие поля (по умолчанию):

– порядковый номер пакета (No.);

– время поступления пакета (Time);

– источник пакета (Source);

– пункт назначения (Destination);

– протокол (Protocol);

– информационное поле (Info).

Список отображаемых полей настраивается в Edit/Perferencis/Columns. Для того, чтобы изменения возымели эффект необходимо перезапустить программу, предварительно нажав кнопку Save. При нажатии правой кнопки мыши на том или ином пакете, появится контекстное меню. Нажатием на среднюю кнопку мыши можно помечать группу интересующих нас пакетов.

Средняя панель содержит т.н. «дерево протоколов» для выбранного в верхнем окне пакета. В этой панели в иерархическом виде для выбранного в верхнем окне захваченного пакета отображается вложенность протоколов в соответствии с моделью взаимодействия открытых систем OSI. По нажатию на правую кнопку мыши вызывается контекстное меню. При «раскрытии» каждого из протокола нажатием на значек «+» слева, выводятся поля данных соответствующих протоколов.

Нижняя панель содержит шестнадцатеричное представление выбранного пакета. При выборе того или иного поля в средней панели автоматически будет подсвечиваться соответствующий участок 16-ого представления.

Ход работы

3 «Захват» пакетов с фильтрами

После выбора интерфейса или нажатия на кнопку «Start» начнётся перехват пакетов. Если сетевая активность высокая, то можно сразу увидеть массу непонятных входящих или исходящих пакетов. Они нас пока мало волнуют, сейчас мы займемся изучением всем известной утилиты *ping*.

3.1 Утилита ping

Нажмем *Win+R* и введем в строке выполнить *cmd*. Откроется консоль, введем там команду *ping <IP адрес>*, как показано на рисунке 8.4. IP адрес следует писать, исходя из конфигурации конкретной сети.

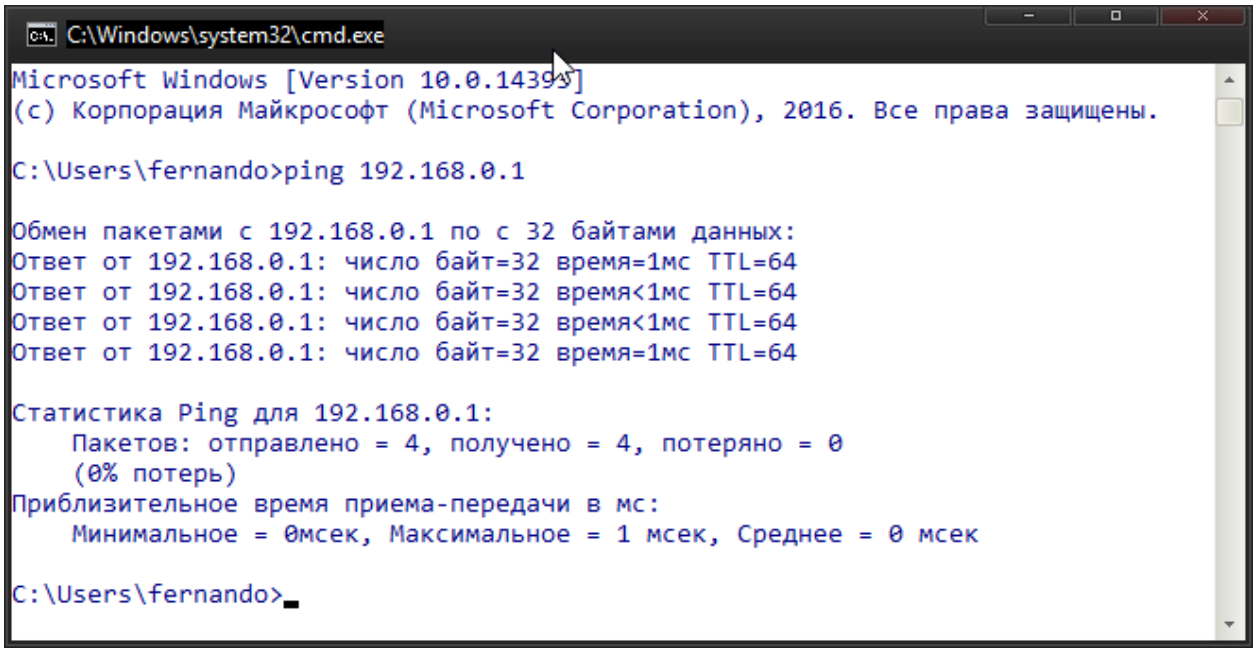

Рисунок 8.4– Выполнение команды ping

После выполнения команды наш трафик будет находиться в Wireshark. Теперь нам требуется отфильтровать данный из общего потока чтобы найти нужные нам пакеты.

Сделать это можно с помощью выражений на панели рис. 8.5. Когда условие введено верно строка подсвечена зелёным цветом, при ошибке красным. Так же с правой стороны имеется кнопка «Expression…», с помощью которой вы можете настроить более точные фильтры используя параметры.

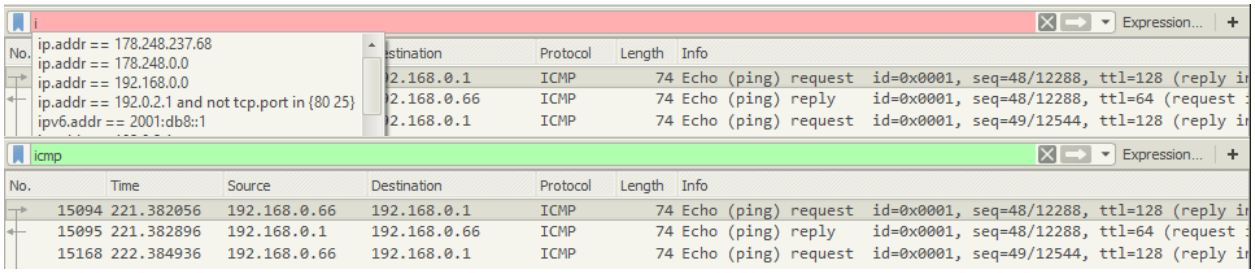

Рисунок 8.5– Строка фильтров

Проанализировав трафик после применения фильтра ICMP можно определить различную информацию о пакете, адресатах и передаваемых данных.

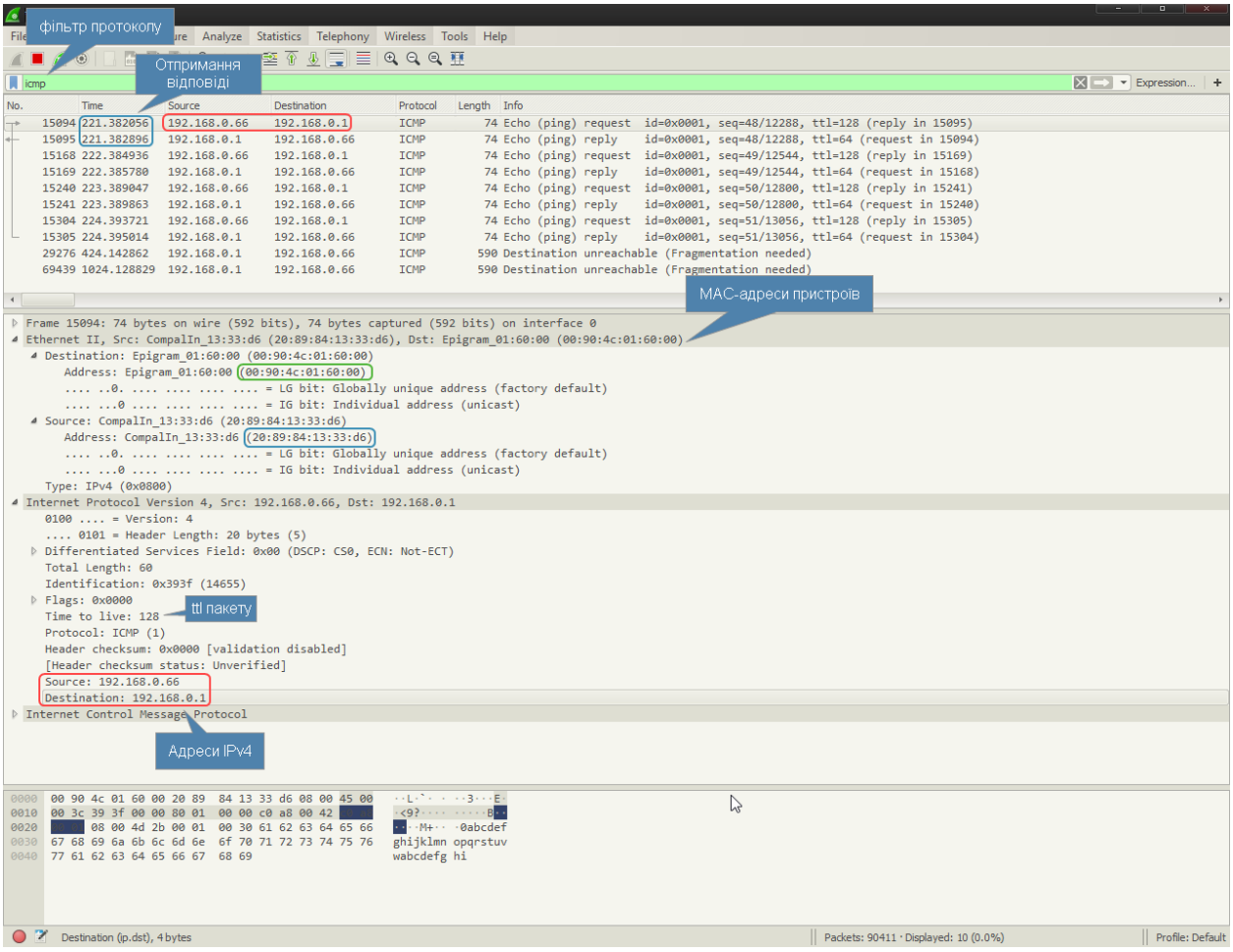

Рисунок 8.6– Пакеты запроса ping к маршрутизатору

4 Проанализировать работу по FTP соединению

В этом пункте рассматривается перехват файла, передающегося по протоколу FTP без шифрования, и убедимся в опасности передачи незащищённого трафика по сети.

Предположим, некий *пользователь,* решил зайти в хранилище и скачать документ. Хранилище использует протокол FTP для передачи данных своим клиентам.

Наиболее часто анализ пакетов происходит уже после непосредственного получения в полном объёме. Но в практической части мы будем параллельно рассматривать действия сервера и клиента.

1) Первым делом нужно запустить сканирование пакетов в Wireshark и настроить фильтр по протоколу «ftp». Будем использовать несколько фильтров, так как FTP использует дополнительный протокол «ftp-data» для передачи данных.

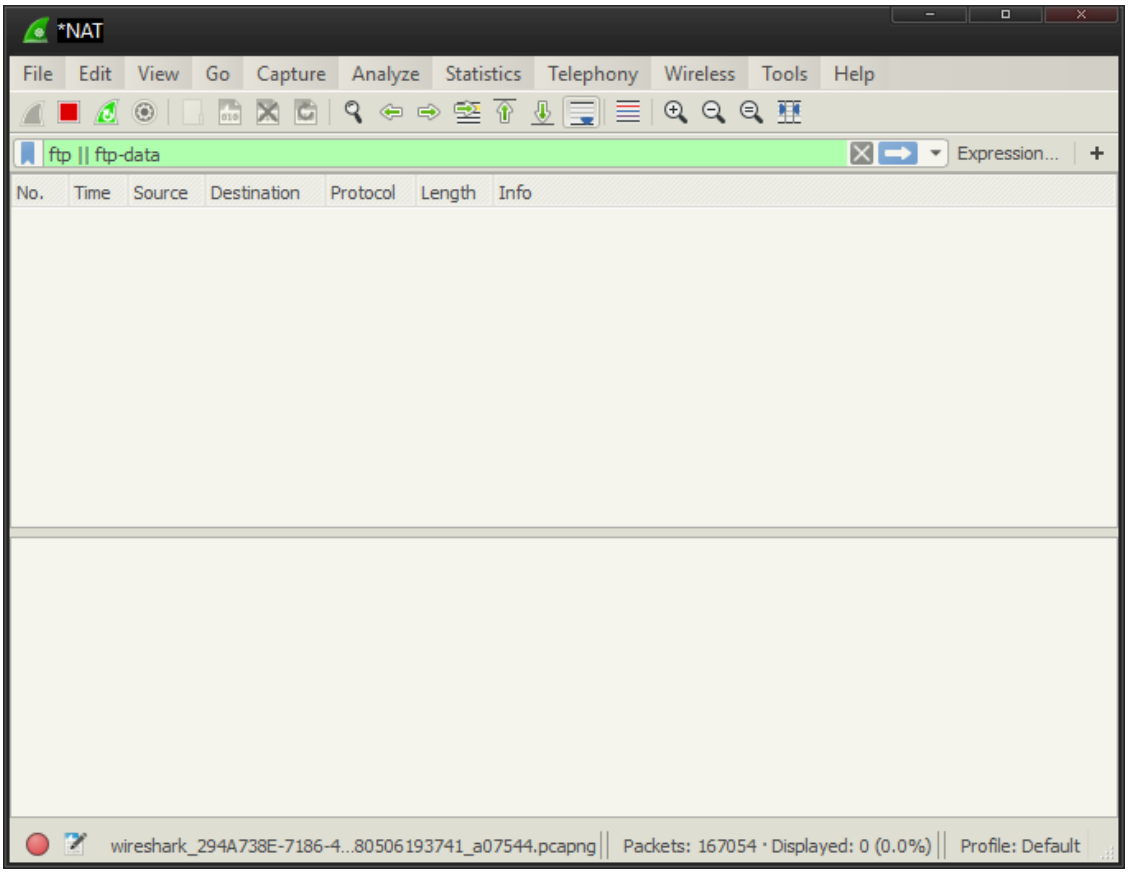

Рисунок 8.7– Установка фильтра для ftp канала

4.1 Авторизация пользователя

Запустить браузер и ввести в строку адрес ftp-сервера, на примере это *«ftp://192.168.0.1»*. Использовать браузер является небезопасным методом, но множество пользователей именно так подключаются. Так что на примере мы будем использовать браузер, для простоты и наглядности.

Перед отображением формы наш сниффер показал следующие строки:

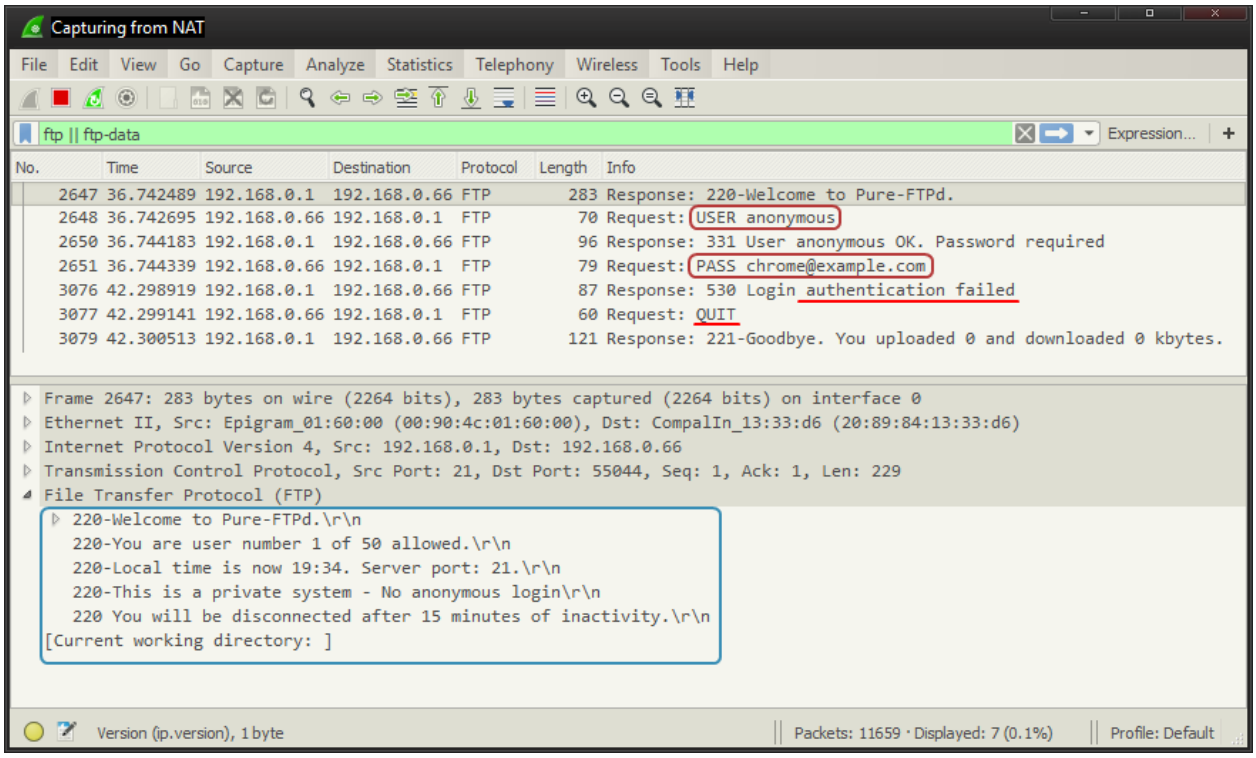

Рисунок 8.8– Анонимная авторизация

Данная передача является стандартом для FTP-протокола. Первым делом сервер высылает приветствие, данные в синей рамке. Клиент отвечает на него послав «**USER** anonymous», желая авторизоваться как гость. Сервер отвечает, что логин принят и ожидает пароля. В качестве пароля клиент высылает почтовый адрес «**PASS** chrome@example.com».

На данном сервере анонимная авторизация отключена. Сервер отправляет 530 ошибку. Клиент отправляет команду «**QUIT**». Сервер подтверждает закрытие канала связи.

Данные пакеты были отправлены ещё до отображения формы клиенту. Браузер всегда старается создать соединение без использования пароля.

После чего на экране клиента появилось сообщение, где ему предложили ввести свой логин и пароль:

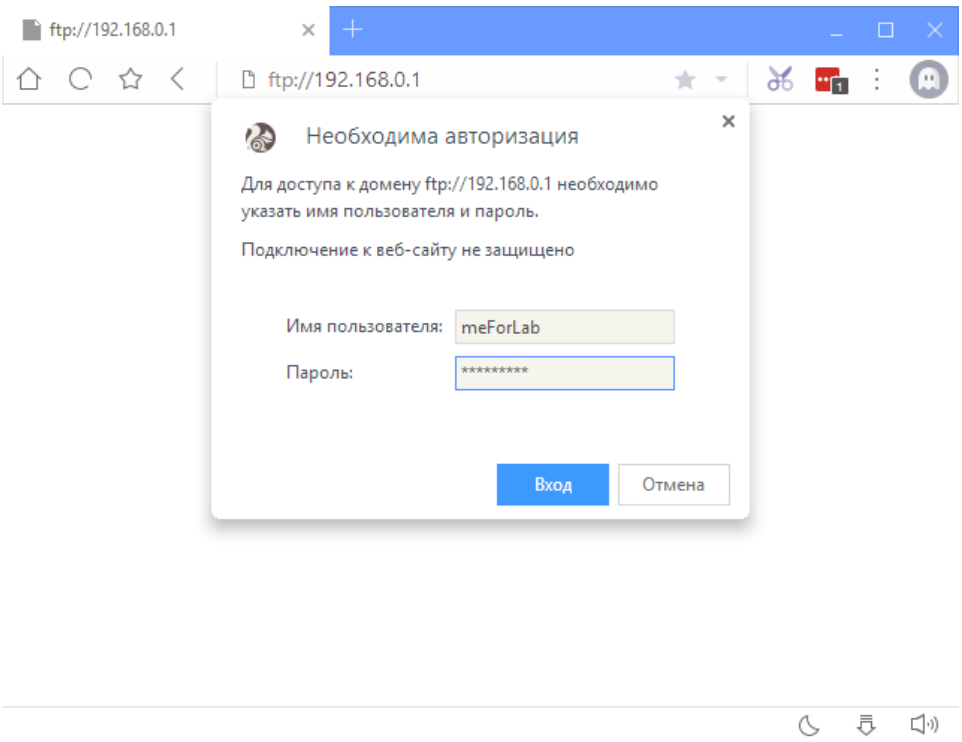

Рисунок 8.9– Авторизация пользователя в браузере

Клиент, заполнив строки и нажав «Вход» отправил данные серверу. Сниффер перехватил пакеты. На данном рисунке 8.10 мы видим данные авторизации: логин и пароль пользователя(1). После чего сервер указывает текущую директорию «/» и клиент-сервер настраивают кодировку и правила отображения (2).

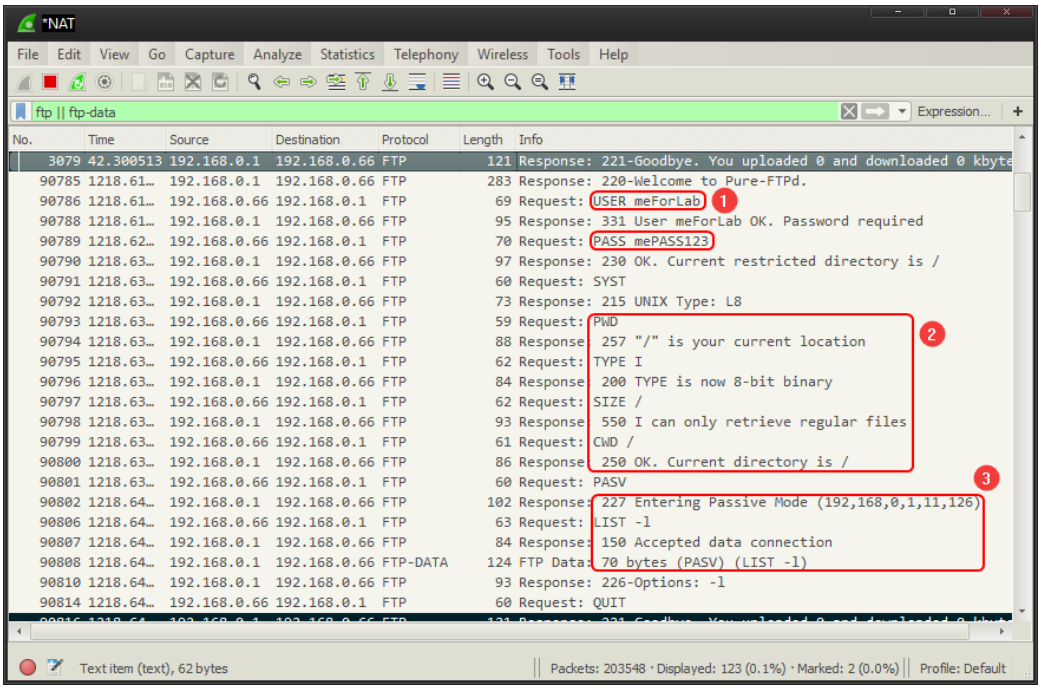

Рисунок 8.10– Авторизация пользователя meForLab

После чего клиент отправляет «PASV», для перехода в пассивный режим. Сервер отвечает «227 Entering Passive Mode (192,168,0,1,11,126)».

Данное сообщение означает, что клиенту нужно обратиться по адресу «192.168.0.1:2942». Порт 2942 выбран из расчёта по формуле:

$$(x \cdot 256) + y = (11 \cdot 256) + 126 = 2942 \qquad (8.1)$$

Клиент подключается, используя протокол «ftp-data» к выбранному адресу и считывает данные. В нашем случае он загрузил данные о каталоге:

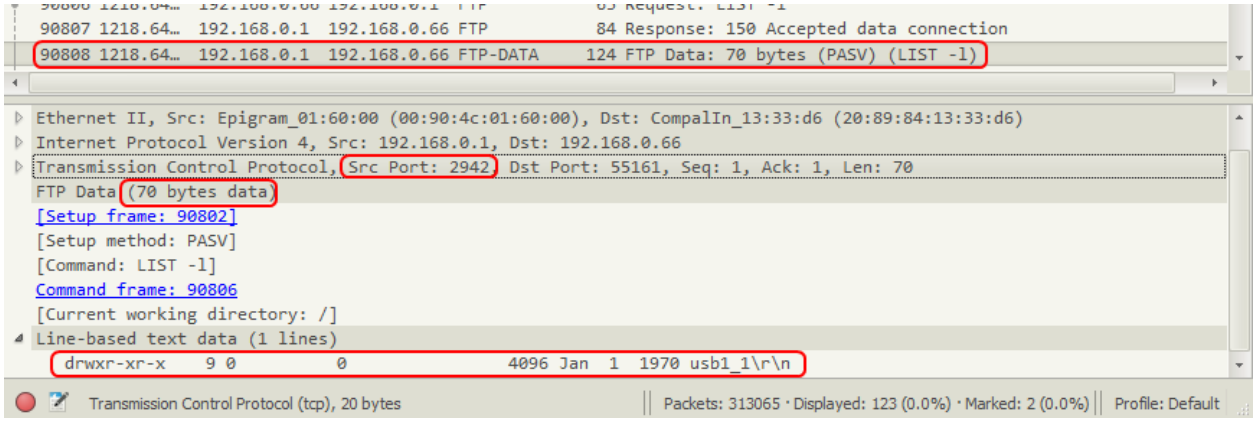

Рисунок 8.11

После ввода данных клиент переходит к каталогу и открывает файл *expl.txt*. Содержимое файла изображено на рис. 8.12.

```
////////////////////////////////////////////////
/-----------------!FOR_LAB!--------------------/
1.Text for example.
2.Пример содержание текста на русском языке.
3.Приклад тексту використовуючі українську мову.
/-----------------!FOR_LAB!--------------------/
////////////////////////////////////////////////
```

Рисунок 8.12– Содержимое файла

Данные полученные из потока сниффером изображены на рис. 8.13. Используя стандарты сниффер определяет содержимое файла по 8-bit. Русские и украинские символы занимают по 16-bit. Так их прочесть не выйдет, так что нужно сохранить данные и открыть их в редакторе.

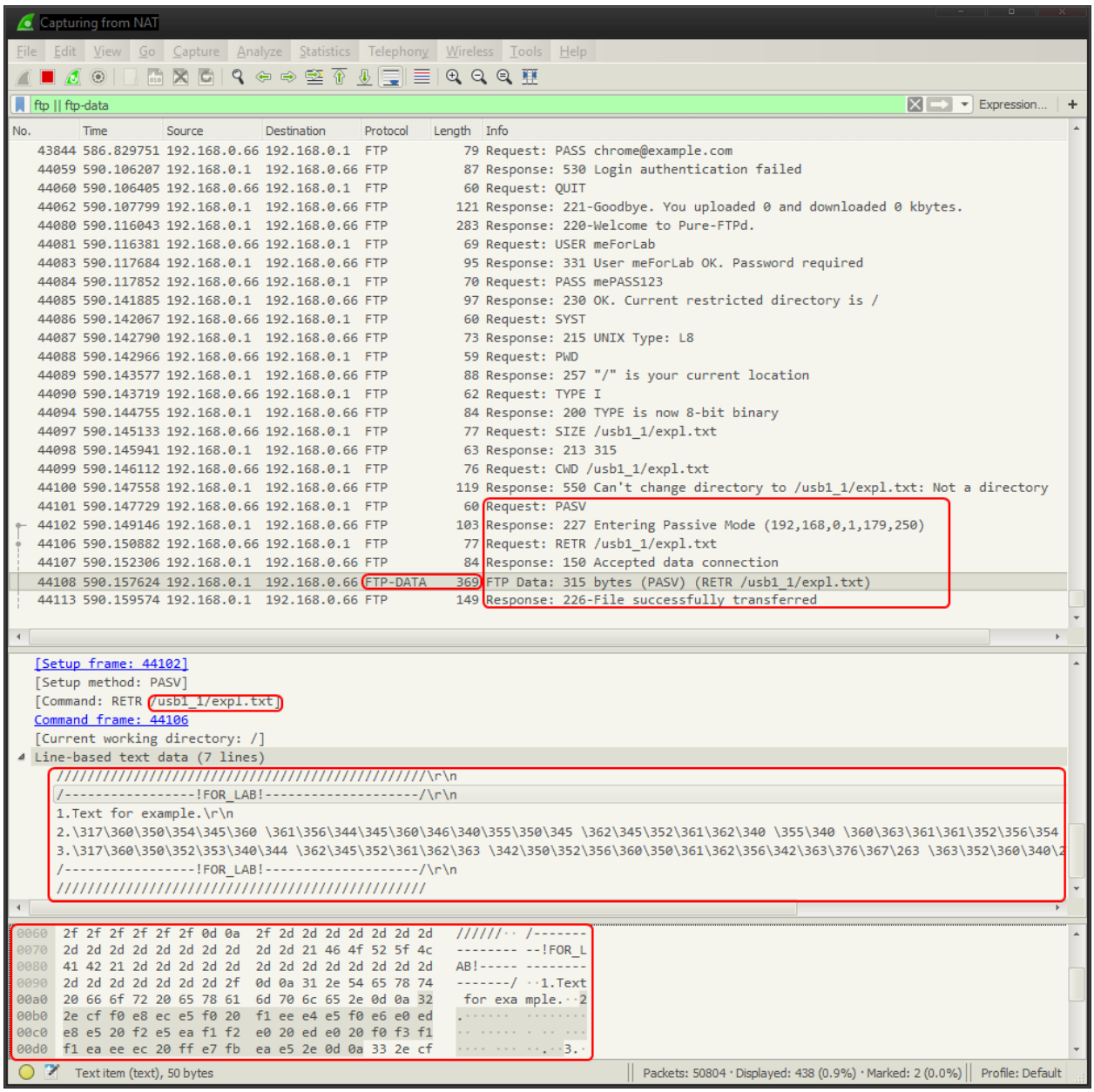

Рисунок 8.13– Данные о файле в сниффере

Передача файлов не всегда происходит через один пакет. В таком случае нам потребуется собрать пакеты в один файл. Для этого нужно выбрать один из пакетов, участвующих в передаче файла. После чего нажать правой клавишей, выбрать пункт «Follow» и «TCP Stream…»(рис. 8.14).

После анализа всего трафика перед нами откроется окно, где будет собраны все данные из различных пакетов одной передачи. В данном окне мы можем выбрать файл(Entire conversation), кодировку(Show and save data as), размер потока(Stream).

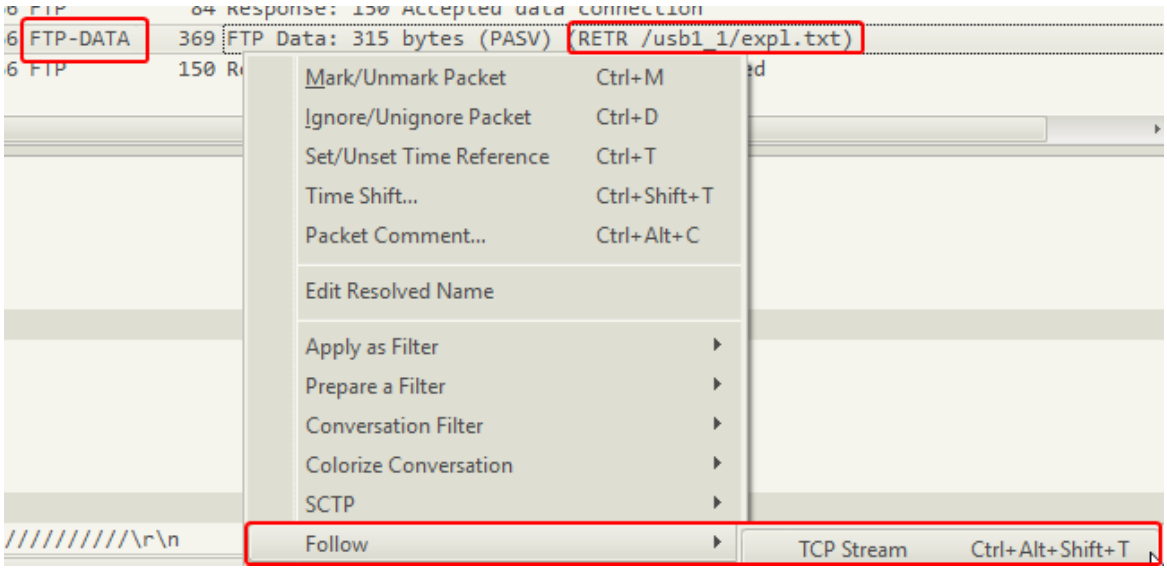

Рисунок 8.14– Выбор пакета для объединения

Для сохранения файла: выберите кодировку «Raw», нажмите «Save as…» и введите имя файла (включая тип txt).

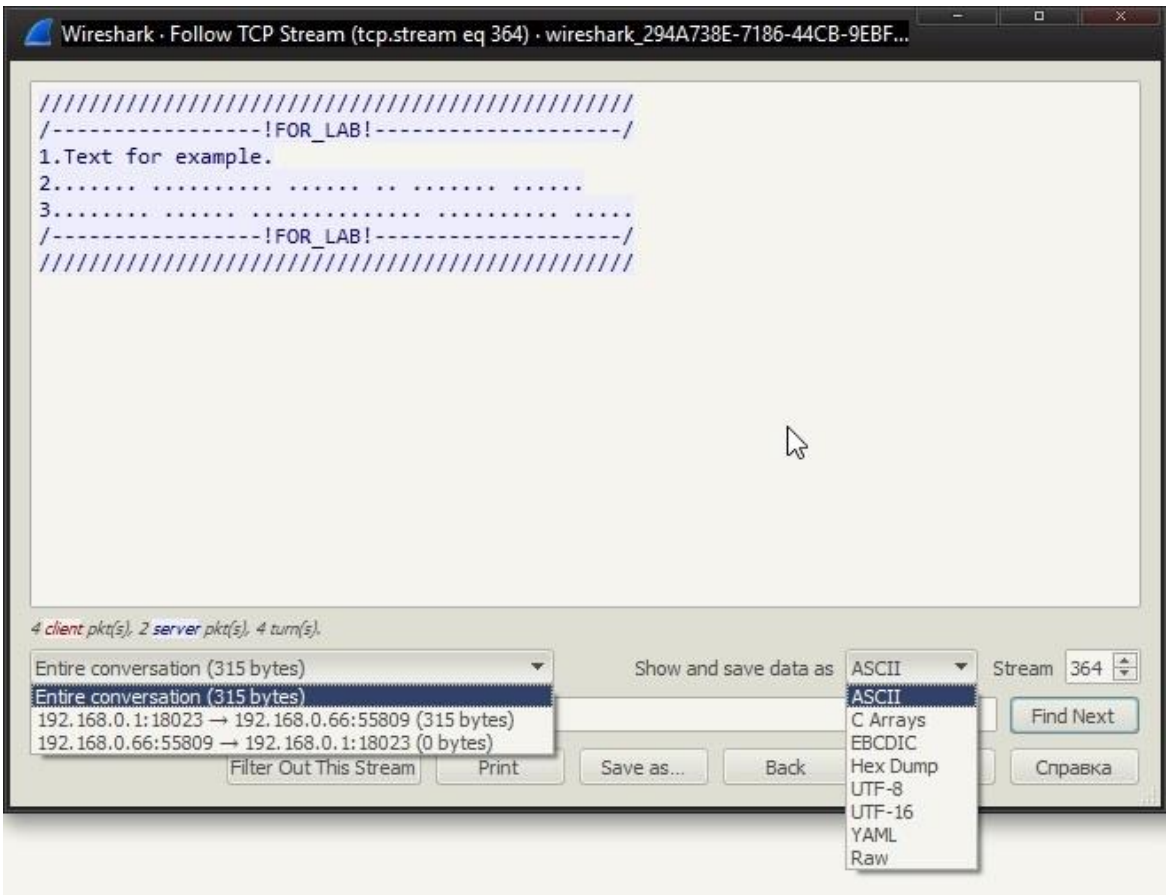

Рисунок 8.15– Окно работы со стримами

5 Сбор статистики

Общая статистика – количество полученных/переданных пакетов, средняя скорость передачи и т.д. доступны через пункт «Statistics», выбрав «Summary». Получить информацию по статистике обработанных протоколов в полученных пакетах можно через пункт «Statistics», «Protocol Hierarchy». Статистику по типу ip-пакетов, их размеру и порту назначения можно получить выбрав подпункты меню «IP-address», «Packet length» и «Port type» соответственно. Одной из наиболее интересных возможностей является генерация диаграммы взаимодействия между узлами, которая доступна пунктом меню «Flow Graph…»(рис. 8.16). В результате можно наблюдать в достаточно наглядной форме процесс взаимодействия на уровне протоколов.

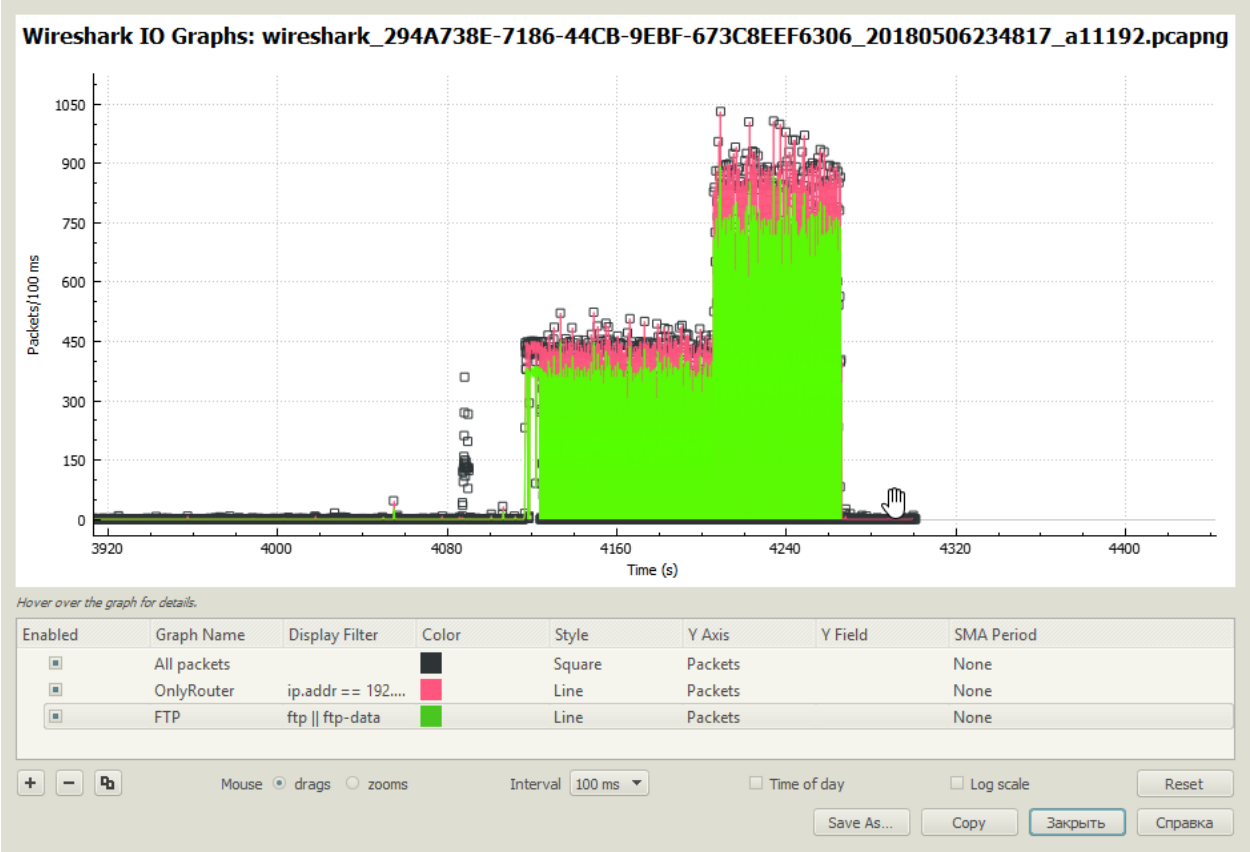

Рисунок 8.16– График трафика сети при загрузке файла большого размера

Задания

1) Изучить теоретический материал и программу Wireshark.

2) Используя инструменты анализатора начать прослушивать трафик в сети.

3) Установить фильтр по протоколу icmp и выполнив команду ping произвести анализ отправленных и полученных пакетов.

4) Проанализировать работу по FTP соединению. Обязательные действия:

– Получить запись авторизации пользователя;

– Список переходов по каталогам;

– Найти и заполнить таблицу свойств передаваемого файла;

– Сохранить копию файла используя Wireshark;

– Сравнить хеш-сумму оригинала и загруженного файла.

5) Научиться производить сбор статистики и структурировать данные перехваченных пакетах на примере данных из http трафика.

В отчёте должны быть представлены заполненные таблицы и описание выполнения вместе с скриншоты.

Вопросы для самоконтроля

1) Каковы основные цели мониторинга сетевого трафика.

2) Чем отличается мониторинг трафика от фильтрации.

3) Что такое сниффер и его функционал.

4) Как происходит захват трафика.

5) Работа сниферов с локальными запросами.

6) Какие безопасные протоколы передачи вы можете предложить.

7) Как защитить сеть от прослушивания.

ДОДАТОК А

ДОДАТОК Б