СОДЕРЖАНИЕ

Введение

В постановлении Президента Республики Узбекистан «О мерах по дальнейшему внедрению и развитию современных информационно-коммуникационных технологий» от 21 марта 2012 год, № ПП-1730 [1] особое внимание уделено вопросам «Совершенствования системы регулирования в сфере информационно-коммуникационных технологий с учетом состояния развития информационных ресурсов технологий и систем...».

Атака на компьютерную систему - это действие, предпринимаемое злоумышленником, которое заключается в поиске и использовании той или иной уязвимости. Таким образом, атака - это реализация угрозы. Заметим, что такое толкование атаки (с участием человека, имеющего злой умысел), исключает присутствующий в определении угрозы элемент случайности, но, как показывает опыт, часто бывает невозможно различить преднамеренные и случайные действия, и хорошая система защиты должна адекватно реагировать на любое из них. В терминах компьютерной безопасности угроза раскрытия имеет место всякий раз, когда получен доступ к некоторой конфиденциальной информации, хранящейся в вычислительной системе или передаваемой от одной системы к другой. Иногда вместо слова «раскрытие» используются термины «кража» или «утечка».

Атаки с распределенным отказом в обслуживании – это реальная и растущая угроза, с которой сталкиваются компании во всем мире. Эти атаки реализуются большим количеством программных агентов, размещенных на хостах, которые злоумышленник скомпрометировал ранее. Реализация этих атак может привести не только к выходу из строя отдельных хостов и служб, но и остановить работу корневых DNS-серверов и вызвать частичное или полное прекращение работы Интернета. В связи с критичностью и нетривиальностью данного класса атак, построение эффективных средств защиты от них представляет собой сложную научно-техническую проблему. Но в целом проблема DDoS-атак на сегодняшний день по-прежнему очень

остро стоит для большинства организаций. Угроза отказа в обслуживании возникает всякий раз, когда в результате некоторых действий блокируется доступ к некоторому ресурсу вычислительной системы. Реально блокирование может быть постоянным, так чтобы запрашиваемый ресурс никогда не был получен, или оно может вызвать только задержку запрашиваемого ресурса, достаточно долгую для того, чтобы он стал бесполезным. В таких случаях говорят, что ресурс исчерпан.

Целью настоящего выпускной квалификационной работы является методика разработки обнаружения DDOS атак в инфокоммуникационных системах(ИКС).

Исходя из вышесказанного можно сделать вывод об актуальности выбранной темы выпускной работы.

Выпускная квалификационная работа состоит из введения, трёх разделов, заключения, списка используемой литературы и приложения.

Во введении обоснована актуальность темы выпускной квалификационной работы.

В первом разделе рассматриваются способы и виды защиты сетевых атак. Предлагаются подходы, основанные на стохастических контекстно-свободных формальных грамматиках, для моделирования атак и построения графов атак и на логическом программировании, для построения графов атак. Исследуется дерево атак, представляющим собой методологию описания угроз и мер противодействия для защиты систем. Используется агрегация подобных хостов как для улучшения наглядности графа атак, так и для повышения производительности. Предлагается архитектура топологического сканера безопасности. А также описываются методы обнаружение атак на сетевом и системном уровне.

Во втором разделе рассматривается анализ сетевого трафика, позволяющий изучить логику работы распределенной вычислительных систем, то есть получить взаимно однозначное соответствие событий,

происходящих в системе, и команд, пересылаемых друг другу ее объектами, в момент появления этих событий. Разрабатывается методика обнаружения DDoS-атак на основе система массового обслуживания, позволяющая получать адекватную оценку частоты потери заявок в сети в случае, если сеть массового обслуживания работает в стационарном режиме. Исследуется разработанной методики обнаружения DDoS-атак в инфокоммуникационных системах.

В третьем разделе рассматривается безопасность жизнедеятельности.

В заключении приведены основные выводы выпускной квалификационной работы.

1. Теоретическая часть. Проблемы и анализ сетевых атак в инфокоммуникационных системах

1.1. Способы и виды защиты атак в системе защиты информации

Современное общество уже не может обойтись без информационных технологий. Информационные технологии проникли во все сферы жизни человека. Их неотъемлемой частью является глобальная сеть Internet. Конечно же, одной из главных задач является обеспечение безопасности обращения информации внутри сети. Одной из опасностей для безопасности являются сетевые атаки. 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. На данный момент выделяют следующие атаки: mailbombing, переполнение буфера, использование специализированных программ (вирусов, снифферов, троянских коней, почтовых червей, rootkit-ов и т.д.), сетевая разведка, IP-спуфинг, man-in-the-middle, инъекция (SQL-инъекция, PHP-инъекция, межсайтовый скриптинг или XSS-атака, XPath-инъекция), отказ в обслуживании (DoS- и DDoS- атаки), phishing-атаки. Рассмотрим каждую из них.

Mailbombing. Суть данной атаки заключается в том, что на почтовый ящик посылается огромное количество писем на почтовый ящик пользователя. Эта атака может вызвать отказ работы почтового ящика или даже целого почтового сервера. Данная атака может проводиться любым хотя бы немного подготовленным противником. Простым примером программы, с помощью которой можно осуществить подобную атаку- The Unabomber. Достаточно знать адрес сервера, позволяющего анонимно отправлять почтовые сообщения, и адрес пользователя, которому эти

сообщения предназначены. Количество писем, которое можно отослать для этой программы равно 12 разрядному числу.

1. Давать адрес электронной почты только проверенным источникам.

2. В качестве преграды для mail bombing-а может выступать и Web-сайт провайдера, иногда настраиваемый таким образом, чтобы он автоматически определял почтовые атаки. В большинстве случаев они распознаются сервером посредством сравнения исходных IP-адресов входящих сообщений. Если количество сообщений из одного источника превышает некие разумные пределы, то все они автоматически поступают в Recycle Bin на сервере. Конечно же, ничто не мешает злоумышленнику фальсифицировать собственный IP-адрес.

Переполнение буфера(buffer overflows). Атака на переполнение буфера основывается на поиске программных или системных уязвимостей, способных вызвать нарушение границ памяти и аварийно завершить приложение или выполнить произвольный бинарный код от имени пользователя, под которым работала уязвимая программа.

Если программа работает под учетной записью администратора, то данная атака может позволить получить полный контроль над компьютером, на котором исполняется данная программа.

Реализации атаки требует решения двух подзадач:

1. Подготовка кода, который будет выполняться в контексте привилегированной программы.

2. Изменение последовательности выполнения программы с передачей управления подготовленному коду.

Таблица 1.1

Классификация атак по переполнению буфера

Подготовка кода Цель	Внедрение кода	Внедрение параметров	Не требуется

переполнения			
Искажение адреса возврата из функции	Атака «срыв стека»	Атака «срыв стека» с параметризацией	Атака «срыв стека» с передачей управления
Искажение указателей функций	Атака на указатели функций	Атака на указатели функций с параметризацией	Атака на указатели функций с передачей управления
Искажение таблиц переходов	Атака на таблицы переходов	Атака на таблицы переходов с параметризацией	Атака на таблицы переходов с передачей управления
Искажение указателей данных	Атака с искажением указателей данных	Атака с искажением указателей данных с параметризацией	Атака с искажением указателей данных с оригинальным кодом

Исходя из подзадач, реализацию которых требует атака, выделяют следующие способы борьбы с атаками подобного типа:

1. Корректировка исходных кодов программы для устранения уязвимостей. Переполнение буфера происходит, прежде всего, из-за неправильного алгоритма работы программы, который не предусматривает проверок выхода за границы буферов. Также возможно применение специальных утилит автоматического поиска уязвимостей в исходном коде программы. Указанные методы и средства позволяют создавать более защищенные программы, но не решают проблему в принципе, а лишь минимизируют число уязвимостей по переполнению буфера. Данный подход ориентирован непосредственно на разработчиков программного обеспечения и не является инструментом конечного пользователя или системного администратора.

2. Использование неисполнимых буферов. Суть метода заключается в запрещении исполнения кода в сегментах данных и стека, т.е. параметры сегментов данных и стека содержат только атрибуты записи и чтения, но не исполнения. Однако ограничение на исполнение данных приводит к проблеме несовместимости. Исполняемый стек необходим для работы многим программам, так как на его основе генерируется код компиляторами, реализуются системные функции операционных систем, реализуется автоматическая генерация кода. Защита с использованием неисполнимых буферов предотвратит только атаки с внедрением кода, но не поможет при других видах атак.

3. Применение проверок выхода за границы. В основе данного метода лежит выполнение проверок выхода за границы переменной при каждом обращении к ней. Это предотвращает все возможные атаки по переполнению буфера, так как полностью исключает само переполнение. Однако, у этого решения есть существенный недостаток - значительное (до 30 раз) снижение производительности программы.

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.

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).

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.

Rootkit - программа или набор программ для скрытия следов присутствия злоумышленника или вредоносной программы в системе. Большинство из реализаций современных rootkit могут прятать от пользователя файлы, папки и ключи реестра, скрывать запущенные программы, системные службы, драйверы и сетевые соединения. Т.е. злоумышленник имеет возможность создавать файлы и ключи реестра, запускать программы, работать с сетью и эта активность не будет обнаружена администратором. Кроме того, rootkits могут скрывать сетевую

активность путем модификации стека протоколов TCP/IP. Так, например rootkit Hacker Defender перехватывает вызовы Winsock и может обрабатывать сетевой трафик до того как он будет передан приложению. Т.е. если в системе установлен Web сервер, и соответственно открыт 80й порт, rootkit может использовать его для взаимодействия с взломщиком, в то время как другие пользователи будут без проблем работать по протоколу HTTP.

Выделяют следующие способы борьбы с этими видами атак:

1. Использование антивирусных средств и регулярное обновление их сигнатур. Может решить проблему с троянскими программами, вирусами, почтовыми червями, но не решит проблему снифферов и rootkit-ов.
2. Шифрование передаваемых данных. Проблема не решает полностью проблему снифферов, однако, противник перехватывает данные, которые нельзя свободно прочитать. Для их расшифровки требуется время.
3. Использование анти снифферов (Например, AntiSniff или PromiScan).
4. Использование межсетевых экранов.
5. Использование антируткитов.

Сетевой разведкой называется сбор информации о сети с помощью общедоступных данных и приложений. При подготовке атаки против какой-либо сети злоумышленник, 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, злоумышленник использует средства сканирования портов, чтобы составить полный список услуг, поддерживаемых этими хостами. И, наконец, злоумышленник анализирует

характеристики приложений, работающих на хостах. В результате добывается информация, которую можно использовать для взлома.

Способы борьбы с данной атакой:

1. Отключение эхо ICMP и эхо-ответ на периферийных маршрутизаторах. Это, однако, приведет к потере данных необходимых для диагностики сетевых сбоев.

2. Использование систем обнаружения вторжений (IDS).

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

Обычно IP-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 злоумышленник должен изменить все таблицы маршрутизации, чтобы направить трафик на ложный IP-адрес. Некоторые злоумышленники, однако, даже не пытаются получить ответ от приложений. Если главная задача состоит в получении от системы важного файла, ответы приложений не имеют значения[2].

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

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

1. 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. Если санкционированными являются и некоторые адреса внешней сети, данный метод становится неэффективным.

2. Фильтрация RFC 2827. Вы можете 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 провайдер. В результате отбраковывается весь трафик, который не имеет исходного адреса, ожидаемого на определенном интерфейсе.

3. Использование криптографической аутентификации.

Атака типа man-in-the-middle. Для атаки типа Man-in-the-Middle злоумышленнику 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.

Способы борьбы с данной атакой:

- Использование шифрования данных
- Инъекция
- SQL-инъекция

SQL-инъекция – это атака, в ходе которой изменяются параметры SQL-запросов к базе данных. В результате запрос приобретает совершенно иной смысл, и в случае недостаточной фильтрации входных данных способен не только произвести вывод конфиденциальной информации, но и изменить/удалить данные.

Способы защиты от данной атаки (используются исключительно администраторами ресурсов):

1. Для целых и дробных величин, перед их использованием в запросе достаточно привести величину к нужному типу.

$id=(int)$id; $total=(float)$total;

Вместо этого можно вставить систему слежения за тестированием на SQL инъекцию.

if((string)$id<>(string)(int)$id) {

die('ops');

}

2. Для строковых параметров, которые не используются в like, regexp и тд, экранируем кавычки.

$str=addslashes($str);

или, лучше,

mysql_escape_string($str)

3. В строках, которые предполагается использовать внутри like, regexp и тд, необходимо так же заэкранировать специальные символы, применяющиеся в этих операторах, если это необходимо. В противном случае, можно задокументировать использование этих символов.

PHP-инъекция. PHP-инъекция - один из способов взлома веб-сайтов, работающих на PHP. Он заключается в том, чтобы внедрить специально

сформированный злонамеренный сценарий в код веб-приложения на серверной стороне сайта, что приводит к выполнению произвольных команд.

Способы борьбы с данной атакой (используются исключительно администраторами ресурсов):

1. Проверять, не содержит ли переменная $name посторонние символы:

```
<?
...
$name = $_GET['name'];
if (strpbrk($name, '.?/:')) die('Blocked');
include $name. '.php';
...
?>
```

2. Проверять, что $name присвоено одно из допустимых значений:

```
<?
...
$name = $_GET['name'];
$arr = array('main', 'about', 'links', 'forum');
if (!in_array($name,$arr)) $name = $arr[0];
include $name . '.php';
...
?>
```

Отказ в обслуживании (DoS- и DDoS- атаки). DoS, без всякого сомнения, является наиболее известной формой атак. Кроме того, против атак такого типа труднее всего создать стопроцентную защиту. Даже среди злоумышленников 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.

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 (Internet Control Message Protocol). 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 через множество устройств, мы говорим о распределенной атаке DoS (DDoS - distributed DoS).

Phishing-атаки. Phishing (фишинг) - процесс обмана или социальная разработка клиентов организаций для последующего воровства их идентификационных данных и передачи их конфиденциальной информации для преступного использования[3]. Преступники для своего нападения используют spam или компьютеры-боты. При этом размер жертвы не имеет

значения; качество личной информации полученной преступниками в результате нападения, имеет значение само по себе.

Приведено пример фишинг-атаки:

Пользователь получает электронную почту от support@mybank.com <mailto:support@mybank.com> (адрес - подменен) со строкой сообщения "модификация защиты", в котором ее просят перейти по адресу www.mybank-validate.info <http://www.mybank-validate.info> (имя домена принадлежит нападавшему, а не банку) и ввести его банковский PIN-код.

Способы защиты от данной атаки:

1. Использовать только проверенные ресурсы и пути доступа к ним.
2. Использовать антивирусные средства и регулярно обновлять их сигнатуры.

1.2. Проблемы синтеза и анализа сетевых атак

Неформально, *граф атак* – это граф, представляющий всевозможные последовательности действий нарушителя для достижения угроз (целей). Такие последовательности действий называются *трассами* (*путями*) *атак*.

Можно выделить следующие виды графов атак:

state enumeration graph – в таких графах вершинам соответствуют тройки (s, d, a), где s – источник атаки, d – цель атаки, a – элементарная атака; дуги обозначают переходы из одного состояния в другое;

condition-orienteddependency graph – вершинам соответствуют результаты атак, а дугам – элементарные атаки, приводящие к таким результатам;

exploit dependency graph – вершины соответствуют результатом атак или элементарным атакам, дуги отображают зависимости между вершинами – условия, необходимые для выполнения атаки и следствие атаки. Например, атака RSH возможна, если нарушитель имеет привилегии супер пользователя

на хосте 1 и хост 3 доверяет хосту 1. В результате атаки нарушитель получает привилегии пользователя на хосте 3.

Под *элементарной атакой* (atomic attack) понимают использование нарушителем уязвимости. Примером элементарной атаки служит, например, переполнение буфера службы SSH, позволяющее удаленно получить права администратора системы.

При синтезе графа атак возникают следующие задачи: формализация понятия атаки, разработка формального языка моделирования атак и компьютерной системы (включающей нарушителя, его цели, сеть, средства защиты, отношение достижимости хостов и т.д.), выбор или разработка средств построения графа атаки и его визуализации, разработка средств автоматизации построения и анализа графа. Обычно такой анализ сводится к последовательному сканированию всех хостов сети на наличие известных уязвимостей. Результатом является отчет, содержащий перечень найденных уязвимостей и рекомендации по их устранению. В настоящее время постепенно внедряется другая парадигма анализа защищенности, учитывающая «топологию» компьютерной системы – взаимосвязь объектов компьютерной системы, их свойств и характеристик. Такой анализ защищенности называется топологическим, а средства, его выполняющие, топологическими сканерами безопасности. Топологический анализ защищенности предполагает построение графа атак на основе результатов сканирования сети, модели нарушителя и данных о конфигурации сети (фильтрации МЭ, маршрутизации, обнаружения атак, достижимости хостов и т.д.) и его анализ (вероятностный, минимизационный и т.д.).

Построенный граф содержит все известные сценарии атак для достижения нарушителем угроз. Результатом его анализа может являться:

- перечень успешных атак, не обнаруживаемых IDS;
- соотношение реализуемых мер безопасности и уровня защищенности сети;

- перечень наиболее критичных уязвимостей;
- перечень мер, позволяющих предотвратить использование уязвимостей в ПО, для которого отсутствуют обновления;
- наименьшее множество мер, реализация которых сделает сеть защищенной.

Графы атак также используются при расследовании компьютерных инцидентов, для анализа рисков и корреляций предупреждений систем обнаружения атак. Первоначально графы атак строили вручную, затем были предложены различные подходы к автоматизации данного процесса. Ключевой проблемой построения графа атак является масштабируемость – возможность построения графа атаки для сети с большим числом хостов и уязвимостей. В данной работе излагаются возможные подходы к построению и анализу графов атак и проблемы, возникающие при этом. Описываются также некоторые системы топологического анализа защищенности.

Основные подходы к синтезу графов атак. Проверка на модели (model checking) – это автоматический метод верификации систем с конечным числом состояний. Применение данного метода состоит из следующих этапов: *моделирование* – формальное описание *M* исследуемой системы; *спецификация* – выражение свойства *P*, которым должна обладать система, в формулах темп оральной логики, позволяющей описывать поведение системы во времени; *верификация* – проверка наличия у модели *M* заданного свойства *P*. Если модель *M* обладает свойством *P*, то возвращается “истина”; иначе приводится *трасса* – последовательность состояний системы, на которой возникает ошибка и, таким образом, не выполняется проверяемое свойство *P*.

Пусть *M* – модель сети, свойство *P* означает ”сеть безопасна”. Тогда, если при верификации окажется, что свойство *P* ложно, средство верификации выдаст трассу атаки, ведущую к нарушению свойства системы.

Основное достоинство данного подхода – формальность: если анализируемая система (а точнее, модель системы) признана безопасной, то это означает, что все варианты атак были рассмотрены, и вопросов касательно полноты анализа не возникает. Если же система не обладает данным свойством, то в процессе проверки всегда будет построен контр пример, представляющий собой последовательность действий нарушителя для достижения угрозы[4]. Недостаток – при большом числе хостов сети число возможных состояний становится необозримым, а анализ – невозможным. Например, применение последнего средства к сети, состоящей из 10 хостов и 5 уязвимостей, позволило получить граф атак, содержащий порядка 10 миллионов ребер. В настоящее время факт неадекватности методов верификации для построения графов атак является общепризнанным.

В настоящее время следующие направления исследований, связанные с подходом model checking, являются актуальными:

1. Использование логик LTL, CTL* и μ-исчисления для задания спецификаций системы.
2. Использование различных систем верификаций моделей (например, SPIN и SVC).

Интерес представляет также возможность применения подходов автоматического доказательства теорем (например, HOL) для построения графов атак.

Формальные грамматики. Котенко и Городецкий предложили подход, основанный на стохастических контекстно-свободных формальных грамматиках, для моделирования атак и построения графов атак. В их работах описывается онтология сетевых атак – систематизация и классификация всевозможных угроз и атак. Формально, *онтология* есть дерево, на множестве узлов которого заданы отношения: «часть», «вид», «последовательность» и «пример». Две вершины дерева соединены ребром,

если и только если они принадлежат одному из отношений. Всякую угрозу можно представить последовательностью символов. Эти последовательности рассматриваются как слова языка, сгенерированного формальной грамматикой. Соединение двух узлов в дереве (онтологии) определяется операцией подстановки, где терминальные символы родительского узла рассматриваются как аксиомы формальной грамматики, соответствующей дочернему узлу.

Логический подход. Предложен подход, основанный на логическом программировании, для построения графов атак. Логический граф атак состоит из вершин вывода (derivation node) и вершин фактов (facts node). Всякой *вершине факта* соответствует логическое высказывание в форме предиката $p(t_1, \ldots, t_k)$, истинное или ложное в зависимости от его аргументов. *Вершине вывода* соответствует правило вывода вида: $L_0 \vdash L_1, \ldots, L_n$. Дуги в графе обозначают отношение зависимости.

В нем предикаты networkService, vulExists, remoteExploit являются примитивными, а предикаты ecexCode и netAccess выводимыми. Правило определяет предусловия и постусловия для атаки: если сервис Program запущен с правами пользователя User на хосте Host, использует порт Port, протокол Protocol и имеет уязвимость с идентификатором VulnID, позволяющую повысить привилегии, и нарушитель имеет сетевой доступ к службе, то он может выполнить код на машине Host от имени User.

Для вычисления всех возможных трасс атак используется модифицированная система доказательств MulVAL, сохраняющая найденные трассы атак. Для этого в правила вывода MulVAL добавляется специальная функция, которая сохраняет вывод всякого истинного правила. По этим сохраненным данным строится граф атак [5]. Доказано, что для построения графа атак сети из N хостов необходимо время между $O(N^2)$ и $O(N^3)$. Данным средством был построен граф атак для сети из 1000 хостов (Pentium 4 CPU, 1 GB RAM).

Graph-based подходы. Дерево атак представляет собой методологию описания угроз и мер противодействия для защиты систем. Дерево атак определяется следующим образом: корню дерева соответствует цель нарушителя, потомкам всякой вершины (*подцелям*) – действия нарушителя для достижения цели. Действия комбинируются с помощью логических И и ИЛИ. Вершинам и ребрам могут назначаться различные числовые значения, которые характеризуют вероятность успеха, сложность, стоимость действий нарушителя. Всякий путь, ведущий от листа к корню – трасса атаки (рис. 1.1).

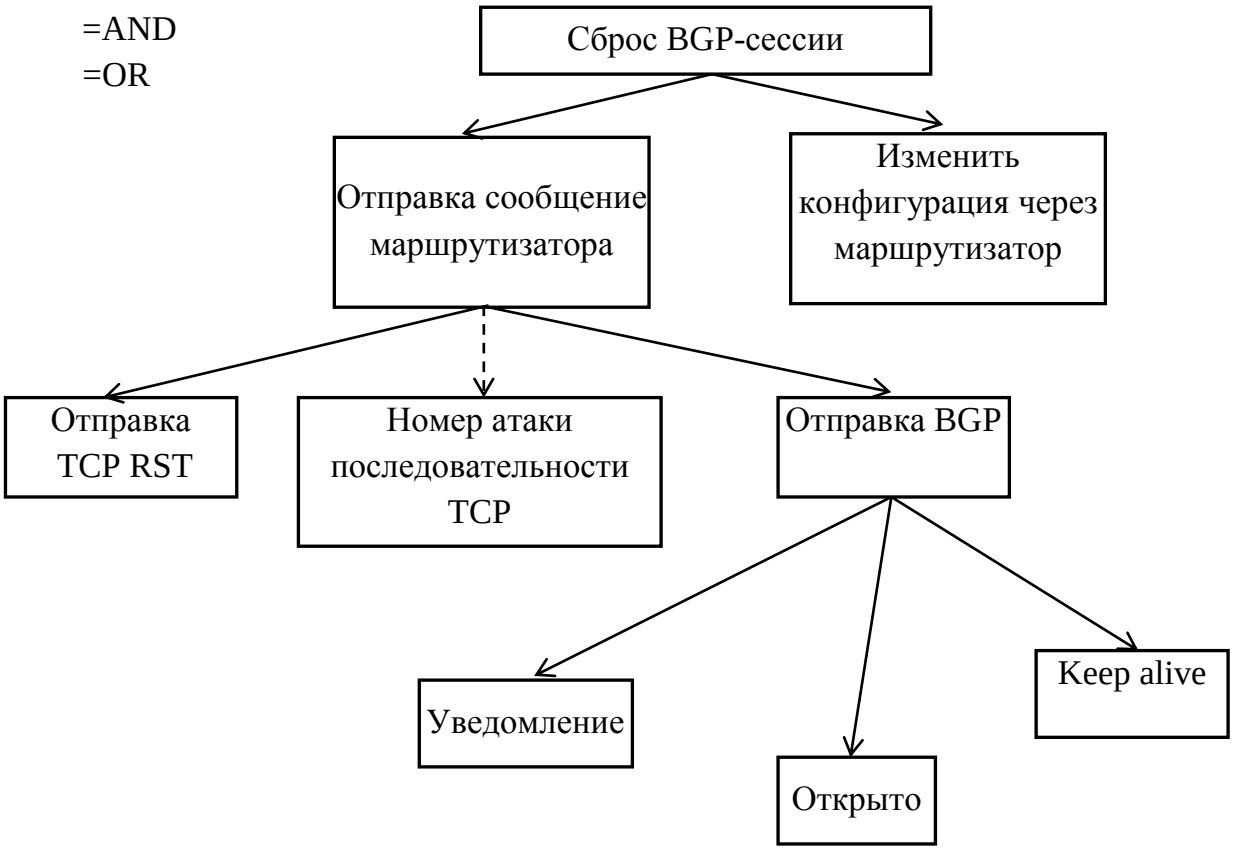

Рис.1.1. Трасса атаки

Определяется расширенное дерево атак введением двух атрибутов: времени жизни *TTL* и степени уверенности *PC*. Первый атрибут отражает временные зависимости между этапами атаки и позволяет уменьшить число ложных срабатываний системы обнаружения атак (СОА). Второй характеризует вероятность достижения цели при достигнутых подцелях.

Затем, используя, строится автомат расширенного дерева атаки. Обнаружение атак заключается в обходе расширенного дерева атак. Параллельный автомат представляет собой формальное объединение автоматов и применяется для обнаружения атак. В качестве примера разбирается обнаружение комплексных атак в беспроводных сетях для стандарта 802.11. Прототип СОА успешно обнаруживает сценарии атак с небольшим числом ложных срабатываний.

Основные проблемы синтеза графов атак. Эффективное моделирование подразумевает автоматизацию процесса построения модели сети. При этом необходимы данные об имеющихся уязвимостях, политики маршрутизации, внедренной политики безопасности и т.д. Обычно во всех топологических сканерах безопасности присутствуют подсистемы, взаимодействующие со сканерами безопасности, МЭ, СОА, маршрутизаторами и другими средствами для построения адекватной модели сети.

Кроме этого, необходимо определить достижимость между всеми хостами, учитывая, например, МЭ и маршрутизаторы. Во многих работах граф строится в предположении наличия таких данных. Простейший подход вычисления доступности требует дополнительных N^2 шагов перед построением графа и заключается в построении матрицы достижимости A размера $N \times N$, где a[i][j] = 1 тогда и только тогда, когда хост j доступен хосту i. Определение достижимости устройств в крупной сети является трудной задачей. Не всегда возможно определить достижимость устройства, используя только сканеры безопасности. Точное определение достижимости между хостами требует анализа конфигурационных правил МЭ, маршрутизаторов, коммутаторов, VPN-шлюзов, персональных МЭ и других устройств.

Применимость алгоритмов для реальных сетей. Многие алгоритмы построения графов не могут быть применены для реальных сетей. Так, все

подходы, основанные на использовании полных графов, являются не эффективными. Например, полный граф атак часто не мог быть вычислен в даже при наличии 13 уязвимостей в файловой системе UNIX-хоста.

Для повышения эффективности решений может использоваться агрегация подобных хостов как для улучшения наглядности графа атак, так и для повышения производительности.

Простейшая агрегация заключается в замене группы идентичных хостов на один хост[6]. Другой метод заключается в построении ограниченного графа, используемого исключительно для ответа на следующие вопросы: какие хосты могут быть скомпрометированы нарушителем с некоторого хоста и какое минимальное множество эксплойтов позволит нарушителю достигнуть его цели? Также неизвестным параметрам системы можно присваивать некоторые значения по умолчанию.

Общая архитектура топологического сканера безопасности. На основе приведенных выше архитектур можно предложить общую архитектуру топологического сканера безопасности. Схематически она представлена на рис.1.2.

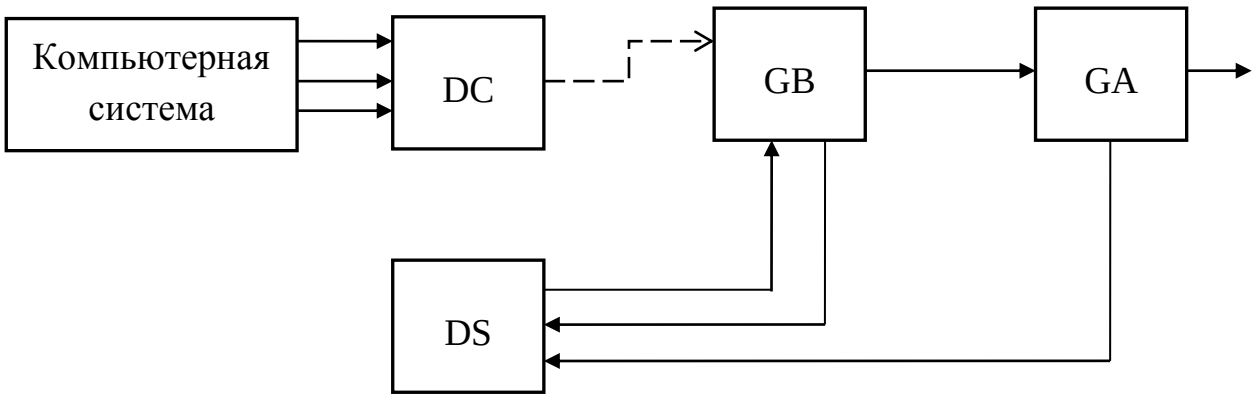

Рис.1.2. Общая архитектура топологического сканера безопасности

Система сбора данных (Data Collector) включает сетевые и хостовые сенсоры и предназначена для сбора данных, необходимых для моделирования компьютерной системы. Хостовые сенсоры устанавливаются на МЭ, маршрутизаторах, серверах, рабочих станциях и служат для проверок

наличия уязвимостей, анализа конфигурационных файлов, таблиц маршрутизации и т.п. Сетевой сенсор устанавливается на отдельную систему и осуществляет сбор сведений и проверки на наличие уязвимостей по сети, выполняя или моделируя атаки. Система хранения данных (Data Storage) состоит из баз данных и предназначена для хранения данных, собранных сенсорами, топологии сети, моделей компьютерных систем, графов атак и отчетов. Система построения графов атак (Graph Builder) предназначена для синтеза графа атак по имеющимся данным. Система анализа (Graph analyzer) служит для проведения анализа защищенности путем исследования графа атак и формирования отчета. В ходе анализа графа система вычисляет всевозможные сценарии нападения, наиболее значимые угрозы и пути их достижения, минимизирует и визуализирует граф атаки.

1.3. Обнаружение атак на сетевом и системном уровне

Системы обнаружения атак сетевого уровня используют в качестве источника данных для анализа необработанные (raw) сетевые пакеты. Как правило, IDS сетевого уровня используют сетевой адаптер, функционирующий в режиме «прослушивания» (promiscuous), и анализируют трафик в реальном масштабе времени по мере его прохождения через сегмент сети. Модуль распознавания атак использует четыре широко известных метода для распознавания сигнатуры атаки:

- соответствие трафика шаблону (сигнатуре), выражению или байткоду, характеризующих об атаке или подозрительном действии;
- контроль частоты событий или превышение пороговой величины;
- корреляция нескольких событий с низким приоритетом;
- обнаружение статистических аномалий.

Как только атака обнаружена, модуль реагирования предоставляет широкий набор вариантов уведомления, выдачи сигнала тревоги и

реализации контрмер в ответ на атаку[7]. Эти варианты изменяются от системы к системе, но, как правило, включают в себя: уведомление администратора через консоль или по электронной почте, завершение соединения с атакующим узлом и/или запись сессии для последующего анализа и сбора доказательств.

Обнаружение атак на системном уровне. Современные системы обнаружения атак системного уровня остаются мощным инструментом для понимания уже осуществленных атак и определения соответствующих методов для устранения возможностей их будущего применения. Современные IDS системного уровня по-прежнему используют журналы регистрации, но они стали более автоматизированными и включают сложнейшие методы обнаружения, основанные на новейших исследованиях в области математики. Как правило, IDS системного уровня контролируют систему, события и журналы регистрации событий безопасности (security log или syslog) в сетях, работающих под управлением Windows NT или Unix. Когда какой-либо из этих файлов изменяется, IDS сравнивает новые записи с сигнатурами атак, чтобы проверить, есть ли соответствие. Если такое соответствие найдено, то система посылает администратору сигнал тревоги или приводит в действие другие заданные механизмы реагирования.

IDS системного уровня постоянно развиваются, постепенно включая все новые и новые методы обнаружения. Один их таких популярных методов заключается в проверке контрольных сумм ключевых системных и исполняемых файлов через регулярные интервалы времени на предмет несанкционированных изменений. Своевременность реагирования непосредственно связана с частотой опроса. Некоторые продукты прослушивают активные порты и уведомляют администратора, когда кто-то пытается получить к ним доступ. Такой тип обнаружения вносит в операционную среду элементарный уровень обнаружения атак на сетевом уровне.

Достоинства систем обнаружения атак на сетевом уровне. IDS сетевого уровня имеют много достоинств, которые отсутствуют в системах обнаружения атак на системном уровне. В действительности, многие покупатели используют систему обнаружения атак сетевого уровня из-за ее низкой стоимости и своевременного реагирования. Ниже представлены основные причины, которые делают систему обнаружение атак на сетевом уровне наиболее важным компонентом эффективной реализации политики безопасности.

1. *Низкая стоимость эксплуатации.* IDS сетевого уровня необходимо устанавливать в наиболее важных местах сети для контроля трафика, циркулирующего между многочисленных систем. Системы сетевого уровня не требуют, чтобы на каждом хосте устанавливалось программное обеспечение системы обнаружения атак. Поскольку для контроля всей сети число мест, в которых установлены IDS невелико, то стоимость их эксплуатации в сети предприятия ниже, чем стоимость эксплуатации систем обнаружения атак на системном уровне.

2. *Обнаружение атак, которые пропускаются на системном уровне.* IDS сетевого уровня изучают заголовки сетевых пакетов на наличие подозрительной или враждебной деятельности. IDS системного уровня не работают с заголовками пакетов, следовательно, они не могут определять эти типы атак. Например, многие сетевые атаки типа «отказ в обслуживании» («denial-of-service») и «фрагментированный пакет» (Tear Drop) могут быть идентифицированы только путем анализа заголовков пакетов, по мере того, как они проходят через сеть. Этот тип атак может быть быстро идентифицирован с помощью IDS сетевого уровня, которая просматривает трафик в реальном масштабе времени. IDS сетевого уровня могут исследовать содержание тела данных пакета, отыскивая команды или определенный синтаксис, используемые в конкретных атаках. Например, когда хакер пытается использовать программу Back Orifice на системах,

которые пока еще не поражены ею, то этот факт может быть обнаружен путем исследования именно содержания тела данных пакета. Как говорилось выше, системы системного уровня не работают на сетевом уровне, и поэтому не способны распознавать такие атаки[8].

3. *Для хакера более трудно удалить следы своего присутствия.* IDS сетевого уровня используют «живой» трафик при обнаружении атак в реальном масштабе времени. Таким образом, хакер не может удалить следы своего присутствия. Анализируемые данные включают не только информацию о методе атаки, но и информацию, которая может помочь при идентификации злоумышленника и доказательстве в суде. Поскольку многие хакеры хорошо знакомы с журналами регистрации, они знают, как манипулировать этими файлами для скрытия следов своей деятельности, снижая эффективность систем системного уровня, которым требуется эта информация для того, чтобы обнаружить атаку.

4. *Обнаружение и реагирование в реальном масштабе времени*. IDS сетевого уровня обнаруживают подозрительные и враждебные атаки по мере того, как они происходят, и поэтому обеспечивают гораздо более быстрое уведомление и реагирование, чем IDS системного уровня. Например, хакер, инициирующий атаку сетевого уровня типа «отказ в обслуживании» на основе протокола TCP, может быть остановлен IDS сетевого уровня, посылающей установленный флаг Reset в заголовке TCP-пакета для завершения соединения с атакующим узлом, прежде чем атака вызовет разрушения или повреждения атакуемого хоста. IDS системного уровня, как правило, не распознают атаки до момента соответствующей записи в журнал и предпринимают ответные действия уже после того, как была сделана запись. К этому моменту наиболее важные системы или ресурсы уже могут быть скомпрометированы или нарушена работоспособность системы, запускающей IDS системного уровня. Уведомление в реальном масштабе времени позволяет быстро среагировать в соответствии с предварительно

определенными параметрами. Диапазон этих реакций изменяется от разрешения проникновения в режиме наблюдения для того, чтобы собрать информацию об атаке и атакующем, до немедленного завершения атаки.

5. *Обнаружение неудавшихся атак или подозрительных намерений.* IDS сетевого уровня, установленная с наружной стороны межсетевого экрана (МСЭ), может обнаруживать атаки, нацеленные на ресурсы за МСЭ, даже несмотря на то, что МСЭ, возможно, отразит эти попытки. Системы системного уровня не видят отраженных атак, которые не достигают хоста за МСЭ. Эта потерянная информация может быть наиболее важной при оценке и совершенствовании политики безопасности.

6. *Независимость от ОС.* IDS сетевого уровня не зависят от операционных систем, установленных в корпоративной сети. Системы обнаружения атак на системном уровне требуют конкретных ОС для правильного функционирования и генерации необходимых результатов.

Достоинства систем обнаружения атак системного уровня. И хотя системы обнаружения атак системного уровня не столь быстры, как их аналоги сетевого уровня, они предлагают преимущества, которых не имеют последние. К этим достоинствам можно отнести более строгий анализ, пристальное внимание к данным о событии на конкретном хосте и более низкая стоимость внедрения.

1. *Подтверждают успех или отказ атаки.* Поскольку IDS системного уровня используют журналы регистрации, содержащие данные о событиях, которые действительно имели место, то IDS этого класса могут с высокой точностью определять – действительно ли атака была успешной или нет. В этом отношении IDS системного уровня обеспечивают превосходное дополнение к системам обнаружения атак сетевого уровня. Такое объединение обеспечивает раннее предупреждение при помощи сетевого компонента и «успешность» атаки при помощи системного компонента.

2. *Контролирует деятельность конкретного узла.* IDS системного уровня контролирует деятельность пользователя, доступ к файлам, изменения прав доступа к файлам, попытки установки новых программ и/или попытки получить доступ к привилегированным сервисам. Например, IDS системного уровня может контролировать всю logon- и logoff-деятельность пользователя, а также действия, выполняемые каждым пользователем при подключении к сети. Для системы сетевого уровня очень трудно обеспечить такой уровень детализации событий. Технология обнаружения атак на системном уровне может также контролировать деятельность, которая обычно ведется только администратором. Операционные системы регистрируют любое событие, при котором добавляются, удаляются или изменяются учетные записи пользователей. IDS системного уровня могут обнаруживать соответствующее изменение сразу, как только оно происходит. IDS системного уровня могут также проводить аудит изменений политики безопасности, которые влияют на то, как системы осуществляют отслеживание в своих журналах регистрации и т.д.

В конечном итоге системы обнаружения атак на системном уровне могут контролировать изменения в ключевых системных файлах или исполняемых файлах. Попытки перезаписать такие файлы или инсталлировать «троянских коней» могут быть обнаружены и пресечены. Системы сетевого уровня иногда упускают такой тип деятельности.

3. *Обнаружение атак, которые упускают системы сетевого уровня.* IDS системного уровня могут обнаруживать атаки, которые не могут быть обнаружены средствами сетевого уровня. Например, атаки, осуществляемые с самого атакуемого сервера, не могут быть обнаружены системами обнаружения атак сетевого уровня.

4. *Хорошо подходит для сетей с шифрованием и коммутацией.* Поскольку IDS системного уровня устанавливается на различных хостах сети предприятия, она может преодолеть некоторые из проблем, возникающие

при эксплуатации систем сетевого уровня в сетях с коммутацией и шифрованием.

Определенные типы шифрования также представляют проблемы для систем обнаружения атак сетевого уровня. В зависимости от того, где осуществляется шифрование (канальное или абонентское), IDS сетевого уровня может остаться «слепой» к определенным атакам. IDS системного уровня не имеют этого ограничения. К тому же ОС, и, следовательно, IDS системного уровня, анализирует расшифрованный входящий трафик.

5. *Обнаружение и реагирование почти в реальном масштабе времени.* Хотя обнаружение атак на системном уровне не обеспечивает реагирования в действительно реальном масштабе времени, оно, при правильной реализации, может быть осуществлено почти в реальном масштабе. В отличие от устаревших систем, которые проверяют статус и содержания журналов регистрации через заранее определенные интервалы, многие современные IDS системного уровня получают прерывание от ОС, как только появляется новая запись в журнале регистрации. Эта новая запись может быть обработана сразу же, значительно уменьшая время между распознаванием атаки и реагированием на нее. Остается задержка между моментом записи операционной системой события в журнал регистрации и моментом распознавания ее системой обнаружения атак, но во многих случаях злоумышленник может быть обнаружен и остановлен прежде, чем нанесет какой-либо ущерб.

6. *Не требуют дополнительных аппаратных средств.* Системы обнаружения атак на системном уровне устанавливаются на существующую сетевую инфраструктуру, включая файловые сервера, Web-сервера и другие используемые ресурсы. Такая возможность может сделать IDS системного уровня очень эффективными по стоимости, потому что они не требуют еще одного узла в сети, которому необходимо уделять внимание, осуществлять техническое обслуживание и управлять им[9].

Необходимость в обеих системах обнаружения атак сетевого и системного уровней. IDS и сетевого, и системного уровней имеют свои достоинства и преимущества, которые эффективно дополняют друг друга. Следующее поколение IDS, таким образом, должно включать в себя интегрированные системные и сетевые компоненты. Комбинирование этих двух технологий значительно улучшит сопротивление сети к атакам и злоупотреблениям, позволит ужесточить политику безопасности и внести большую гибкость в процесс эксплуатации сетевых ресурсов.

1.4. Методы защиты от DDoS-атак в компьютерных сетях

Борьба с распределенными DDoS-атаками – дело достаточно непростое. Во-первых, очень трудно установить организатора атаки, а пользователи, чьи компьютеры генерируют паразитический трафик, как правило, даже не подозревают, что их машины стали инструментом в руках злоумышленников. Во-вторых, практически невозможно отличить вредоносный трафик от легитимного, поскольку по сути это те же самые запросы, что и от обычных пользователей, но в неизмеримо большем количестве.

Анализ аномалий в сетевом трафике – единственный эффективный метод обнаружения DDoS-атаки. С точки зрения защиты, DDoS-атаки являются одной из самых сложных сетевых угроз, поэтому принятие эффективных мер противодействия является исключительно сложной задачей для организаций, деятельность которых зависит от интернета. DDoS-атаку очень сложно выявить и предотвратить, поскольку «вредоносные» пакеты неотличимы от «легитимных». Сетевые устройства и традиционные технические решения для обеспечения безопасности сетевого периметра, такие как межсетевые экраны, маршрутизация в «черные дыры» и системы обнаружения вторжений (IDS), являются важными компонентами общей

стратегии сетевой безопасности, однако одни эти устройства не обеспечивают полной защиты от DDoS-атак.

Маршрутизация в «черные дыры». Процесс маршрутизации в «черные дыры» применяется провайдером услуг для блокировки всего трафика, адресованного на целевой объект, в как можно более ранней точке[10]. «Снятый с маршрута» трафик маршрутизируется в «черную дыру» для защиты сети провайдера и других его клиентов. Маршрутизацию в «черные дыры» нельзя назвать удачным решением, поскольку вместе со злоумышленным трафиком атаки отбраковываются и благонадежные пакеты. Жертвы полностью лишаются своего трафика, и хакер празднует победу.

Списки контроля доступа. Многие полагают, что маршрутизаторы, на которых применяются списки контроля доступа (ACL) для фильтрации «нежелательного» трафика, обеспечивают защиту от атак DDoS. Действительно, списки ACL могут защитить от простых и известных атак DDoS, например, от ICMP-атак, фильтрация второстепенных, неиспользуемых протоколов.

Однако на сегодняшний день в атаках DDoS, как правило, используются корректные действующие протоколы, которые необходимы для присутствия в сети Интернет, и поэтому фильтрация протоколов становится менее эффективным средством защиты. Маршрутизаторы также могут блокировать зоны с некорректными IP-адресами, однако хакеры, чтобы их не обнаружили, обычно подделывают корректные IP-адреса. В целом, хотя списки ACL на маршрутизаторах служат первой линией обороны от базовых атак, они не оптимизированы для защиты от следующих сложных атак DDoS:

- SYN, SYN-ACK, FIN и другие лавинные атаки. Списки ACL не могут заблокировать атаку SYN с произвольным выбором объектов спуфинга (проставление в поле обратного адреса IP-пакета неверного адреса) или атаки ACK и RST на 80-й порт Веб-сервера, при которых поддельные IP-адреса

источника постоянно меняются, поскольку для этого потребовалось бы вручную отследить и идентифицировать каждый подделанный источник, а эта задача практически невыполнима. Единственный возможный вариант здесь состоит в том, что заблокировать весь сервер, а именно в этом и состоит задача хакера;

- Proxy – Поскольку списки ACL не могут отличить друг от друга «благонадежные» и «злоумышленные» SYN, поступающие из одного исходного IP или Proxy, то, пытаясь остановить сфокусированную атаку со спуфингом, они вынуждены, по определению, блокировать весь трафик клиентов жертвы, поступающий из определенного исходного IP или Proxy (модуля доступа);
- DNS или протокол пограничного шлюза (Border Gateway Protocol, BGP) – Когда запускаются атаки с произвольным выбором объектов спуфинга на сервер DNS или на маршрутизатор BGP, списки ACL, как и в случае с лавинными атаками SYN, не могут отследить быстро меняющийся объем трафика с произвольно выбранными объектами спуфинга. Кроме этого, списки ACL не в состоянии отличить поддельные адреса от корректных;
- атаки на уровне приложений (клиентские) – Хотя списки ACL теоретически могут блокировать клиентские атаки, например, атаки с ошибочными соединениями HTTP и с полуоткрытыми соединениями HTTP (при условии, что есть возможность точно идентифицировать источник атаки и конкретные не подделанные источники), пользователям потребуется конфигурировать сотни, а в некоторых случаях и тысячи списков ACL для каждой потенциальной жертвы.

Межсетевые экраны. Хотя межсетевые экраны играют исключительно важную роль в системе безопасности любой компании, они не созданы именно как инструмент предотвращения атак DDoS. Фактически, у межсетевых экранов есть ряд исходных свойств, которые не позволяют им

обеспечить полную защиту от самых изощренных современных атак DDoS. Прежде всего, это отсутствие механизма выявления аномалий. Межсетевые экраны в первую очередь предназначены для контроля доступа в частные сети, и они отлично справляются с этой задачей. Один из путей выполнения этой задачи – отслеживание сеансов, которые инициированы изнутри (на «чистой» стороне) и адресованы на внешний сервис, и последующий прием только особых откликов от ожидаемых источников на внешней стороне. Однако такая схема не действует применительно к таким сервисам как Web, DNS, и к другим сервисам, которые должны быть открыты для общего доступа, чтобы была обеспечена возможность принимать запросы. В подобных случаях межсетевые экраны выполняют операцию, которая называется «открыванием канала»: они пропускают трафик HTTP на IP-адрес веб-сервера. Хотя такой подход и обеспечивает некоторую защиту, поскольку разрешены лишь определенные протоколы, адресуемые на определенные адреса, он не слишком эффективен в борьбе с атаками DDoS, поскольку хакеры могут без труда воспользоваться «разрешенным» протоколом (в данном случае HTTP) для переноса трафика атаки. Отсутствие возможностей для выявления аномалий означает, что межсетевые экраны не могут распознать ситуацию, в которой носителем атаки служат корректные разрешенные протоколы.

Также в межсетевых экранах отсутствуют ресурсы борьбы со спуфингом. Если выявлена атака DDoS, межсетевые экраны могут заблокировать конкретный поток трафика, связанный с атакой, но не могут применить меры антиспуфинга, чтобы отделить хороший, «благонадежный» трафик от плохого, а именно эта операция важна для защиты от атак, в которых используется большой объем подделанных IP-адресов.

Реакция на атаки DDoS-атаки. Процедуры защиты от атак DDoS, инициируемые вручную, можно охарактеризовать словами «слишком мало, слишком поздно». Первая реакция жертвы на атаку DDoS, как правило,

заключается в том, что он просит ближайшего предшествующего провайдера услуг соединения (это может быть провайдер Интернет-услуг, провайдер услуг хостинга или магистральный) попытаться идентифицировать источник. Если адреса подделаны или их слишком много, этот процесс может оказаться долгим и трудным, и для его реализации будет необходимо объединить усилия многих провайдеров. Хотя источник, возможно, и будет идентифицирован, блокировка этого источника выльется в блокировку всего трафика – и плохого, и хорошего.

Анализ аномалий в сети. Во время обнаружить DDoS-атаку — в этом и заключается главная проблема, если мы не хотим бороться с ней по факту падения ресурсов сети. Наиболее эффективный способ обнаружить DDoS-атаку основан на накоплении статистических данных о прохождении трафика в сети. Составив картину нормального состояния сети, всегда можно отследить возникновение какой-либо аномалии в сетевом трафике. В качестве источника данных для статистики можно использовать сам трафик или некоторую статистическую информацию о нем. Для этого используются либо дополнительные сенсоры, устанавливаемые на сеть, либо информация, которую могут предоставить существующие сетевые элементы. В случае снятия такой информации непосредственно с маршрутизаторов обычно используется протокол Netflow. Этот протокол был в свое время разработан для оптимизации работы маршрутизаторов, его задача заключалась в том, чтобы не обрабатывать каждый пакет, а перенаправлять его как можно быстрее, если он соответствовал требованиям потока. Протокол оказался неэффективным для решения основной задачи, но очень пригодился для борьбы с DDoS-атаками и шире — для анализа работы сети в целом. Такие способности протоколу дает заложенная в него возможность формировать таблицу, в которой в динамическом режиме прописываются все статистические данные по пришедшим потокам и пакетам: откуда пришел пакет, куда он направляется, какой у него протокол, порт, какое количество

данных передано. Причем имеется возможность экспорта статистических данных во внешние системы для последующего анализа.

Ситуация, при которой текущий трафик на защищаемый ресурс резко отличается от нормального, считается DDoS-атакой. Стоит подчеркнуть, что система распознает только отклонение от трафика, а чем он вызван — всплеском легитимных обращений к ресурсам (выложили новый патч, прошла рекламная кампания) или DDoS-атакой — может определить только владелец ресурса, ожидал ли он такой объем обращений или нет [11].

После обнаружения факта аномалии происходит ее классификация и определяется, насколько она серьезна. Если DDoS-атака не грозит возникновением проблем в сети, то лучше наблюдать и ничего не предпринимать, так как возникает вероятность не пустить на ресурс законного пользователя.

Архитектура защиты от DDoS-атак. Общая схема противодействия DDoS-атакам представлена на рис.1.3

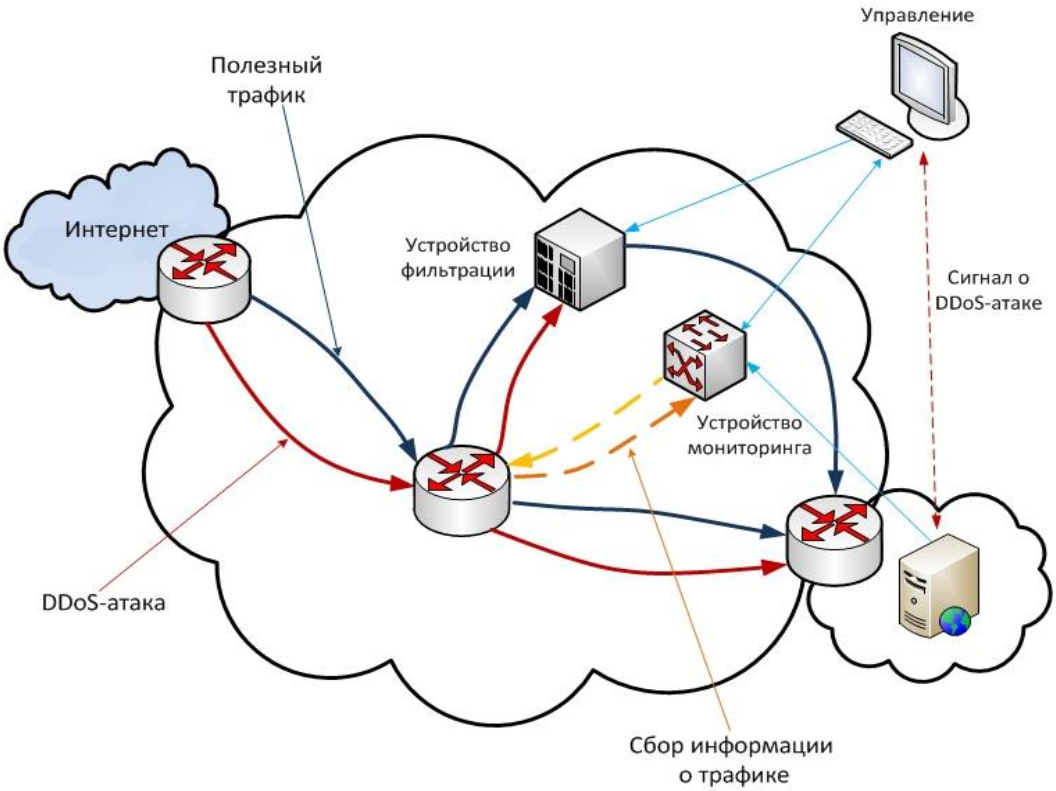

Рис.1.3. Схема противодействия DDoS-атакам

Техническая реализация данного решения предполагает наличие в сети двух дополнительных устройств, одно из которых осуществляет мониторинг входящего трафика и выявляет проведение DDoS-атаки, а второе фильтрует (очищает) поступающий извне трафик. В нормальном режиме работы данные устройства не должны оказывать никакого влияния на проходящий трафик. В случае же атаки устройство «очистки» задерживает трафик, идентифицируемый как DDoS-пакеты, не допуская его попадания в относительно узкополосные клиентские каналы и на клиентские ресурсы, тем самым не прерывая предоставление клиенту основной услуги.

2. Основная часть. Разработки методики обнаружения DDoS атак в компьютерных сетях

2.1. Механизмы реализации типовых удаленных атак

Исследования и анализ информационной безопасности различных распределенных ВС, проводимые в течение последних лет, наглядно продемонстрировали тот факт, что, независимо от используемых сетевых протоколов, топологии, инфраструктуры исследуемых распределенных ВС, механизмы реализации удаленных воздействий на РВС инвариантны по отношению к особенностям конкретной системы. Это объясняется тем, что распределенные ВС проектируются на основе одних и тех же принципов, а, следовательно, имеют практически одинаковые проблемы безопасности; Поэтому оказывается, что причины успеха удаленных атак на различные РВС одинаковы. Таким образом, появляется возможность ввести понятие типовой удаленной атаки. *Типовая удаленная атака*- это удаленное информационное разрушающее воздействие, программно осуществляемое по каналам связи и характерное для любой распределенной ВС. Введение этого понятия в совокупности с описанием механизмов реализации типовых УА позволяет предложить методику исследования безопасности, инвариантную по отношению к виду распределенной ВС. Методика заключается в последовательном осуществлении всех типовых удаленных воздействий в соответствии с предложенным далее их описанием и характеристиками. При этом основным элементом исследования безопасности РВС является анализ сетевого трафика. Как пояснение последнего утверждения рассмотрим следующую аналогию: отладчик - основное средство для хакера, соответственно анализатор сетевого трафика - основное средство для сетевого хакера. Анализатор сетевого трафика по своей сути является сетевым отладчиком. Итак, в качестве методики исследования информационной безопасности распределенной ВС предлагается выполнение

ряда тестовых задач, оценивающих защищенность системы по отношению к типовым удаленным воздействиям. Рассмотрено в следующих пунктах типовые удаленные атаки и механизмы их реализации.

Анализ сетевого трафика. Как уже отмечалось, основной особенностью распределенной ВС является то, что ее объекты распределены в пространстве и связь между ними физически осуществляется по сетевым соединениями программно - при помощи механизма сообщений. При этом все управляющие сообщения и данные, пересылаемые между объектами РВС, передаются по сетевым соединениям в виде пакетов обмена. Эта особенность привела к появлению специфичного для распределенных ВС типового удаленного воздействия, заключающегося в прослушивании канала связи. Назовем данное типовое удаленное воздействие *анализом сетевого трафика* (или, сокращенно, сетевым анализом).

Анализ сетевого трафика позволяет, во-первых, изучить логику работы распределенной ВС, то есть получить взаимно однозначное соответствие событий, происходящих в системе, и команд, пересылаемых друг другу ее объектами, в момент появления этих событий[12]. Это достигается путем перехвата и анализа пакетов обмена на канальном уровне. Знание логики работы распределенной ВС позволяет на практике моделировать и осуществлять типовые удаленные атаки, рассмотренные в следующих пунктах на примере конкретных распределенных ВС.

Во-вторых, анализ сетевого трафика позволяет перехватить поток данных, которыми обмениваются объекты распределенной ВС. Таким образом, удаленная атака данного типа заключается в получении на удаленном объекте несанкционированного доступа к информации, которой обмениваются два сетевых абонента. Отметим, что при этом отсутствует возможность модификации трафика и сам анализ возможен только внутри одного сегмента сети. Примером перехваченной при помощи данной типовой

удаленной атаки информации могут служить имя и пароль пользователя, пересылаемые вне зашифрованном виде по сети.

Подмена доверенного объекта или субъекта распределенной ВС. Одной из проблем безопасности распределенной ВС является недостаточная идентификация и аутентификация ее удаленных друг от друга объектов. Основная трудность заключается в осуществлении однозначной идентификации сообщений, передаваемых между субъектами и объектами взаимодействия. Обычно в распределенных ВС эта проблема решается следующим образом: в процессе создания виртуального канала объекты РВС обмениваются определенной информацией, уникально идентифицирующей данный канал. Такой обмен обычно называется "рукопожатием" (handshake). Однако, отметим, что не всегда для связи двух удаленных объектов в РВС создается виртуальный канал. Практика показывает, что зачастую, особенно для служебных сообщений (**!?**) (например, от маршрутизаторов) используется передача одиночных сообщений, не требующих подтверждения.

Как известно, для адресации сообщений в распределенных ВС используется сетевой адрес, который уникален для каждого объекта системы (на канальном уровне модели OSI - это аппаратный адрес сетевого адаптера, на сетевом уровне - адрес определяется в зависимости от используемого протокола сетевого уровня (например, IP-адрес).Сетевой адрес также может использоваться для идентификации объектов распределенной ВС. Однако сетевой адрес достаточно просто подделывается и поэтому использовать его в качестве единственного средства идентификации объектов недопустимо.

В том случае, когда распределенная ВС использует нестойкие алгоритмы идентификации удаленных объектов, то оказывается возможной типовая удаленная атака, заключающаяся в передаче по каналам связи сообщений от имени произвольного объекта или субъекта РВС. При этом существуют две разновидности данной типовой удаленной атаки:

- атака при установленном виртуальном канале;
- атака без установленного виртуального канала.

В случае установленного виртуального соединения атака будет заключаться в присвоении прав доверенного субъекта взаимодействия, легально подключившегося к объекту системы, что позволит атакующему вести сеанс работы с объектом распределенной системы от имени доверенного субъекта. Реализация удаленных атак данного типа обычно состоит в передаче пакетов обмена с атакующего объекта на цель атаки от имени доверенного субъекта взаимодействия (при этом переданные сообщения будут восприняты системой как корректные). Для осуществления атаки данного типа необходимо преодолеть систему идентификации и аутентификации сообщений, которая, в принципе, может использовать контрольную сумму, вычисляемую с помощью открытого ключа, случайные много битные счетчики пакетов и сетевые адреса станций. Как было замечено выше, для служебных сообщений в распределенных ВС часто используется передача одиночных сообщений, не требующих подтверждения, то есть не требуется создание виртуального соединения. Атака без установленного виртуального соединения заключается в передаче служебных сообщений от имени сетевых управляющих устройств, например, от имени маршрутизаторов.

Очевидно, что в этом случае для идентификации пакетов возможно лишь использование статических ключей, определенных заранее, что довольно неудобно и требует сложной системы управления ключами. Однако, при отказе от такой системы идентификация пакетов без установленного виртуального канала будет возможна лишь по сетевому адресу отправителя, который легко подделать[13].

Посылка ложных управляющих сообщений может привести к серьезным нарушениям работы распределенной ВС(например, к изменению ее конфигурации.

Ложный объект распределенной ВС. В том случае, если в распределенной ВС недостаточно надежно решены проблемы идентификации сетевых управляющих устройств (например, маршрутизаторов), возникающие при взаимодействии последних с объектами системы, то подобная распределенная система может подвергнуться типовой удаленной атаке, связанной с изменением маршрутизации и внедрением в систему ложного объекта. В том случае, если инфраструктура сети такова, что для взаимодействия объектов необходимо использование алгоритмов удаленного поиска, то это так же позволяет внедрить в систему «Ложный объект РВС».

Отказ в обслуживании. Одной из основных задач, возлагаемых на сетевую ОС, функционирующую на каждом из объектов распределенной ВС, является обеспечение надежного удаленного доступа с любого объекта сети к данному объекту. Нарушение работоспособности соответствующей службы предоставления удаленного доступа, то есть невозможность получения удаленного доступа с других объектов РВС – «Отказ в обслуживании».

Таблица 2.1

Классификация типовых удаленных атак на распределенные ВС

Типовая удаленная атака	Характер воздействия		Цель воздействия			Условие начала осуществления воздействия			Наличие обратной связи с атакуемым объектом		Расположение субъекта атаки относительно атакуемого объекта		Уровень модели OSI						
Класс воздействия	1.1	1.2	2.1	2.2	2.3	3.1	3.2	3.3	4.1	4.2	5.1	5.2	6.1	6.2	6.3	6.4	6.5	6.6	6.7

Анализ сетевого трафика	+	-	+	-	-	-	-	+	-	+	+	-	-	+	-	-	-	-	-
Подмена доверенного объекта РВС	-	+	+	+	-	-	+	-	+	+	+	+	-	-	+	+	-	-	-
Внедрение в РВС ложного объекта путем навязывания ложного маршрута	-	+	+	+	+	-	-	+	+	+	+	+	-	-	+	-	-	-	-
Внедрение в РВС ложного объекта путем использования недостатков алгоритмов удаленного поиска	-	+	+	+	-	+	-	+	+	-	+	+	-	+	+	+	-	-	-

Отказ в обслуживании	-	+	-	-	+	-	-	+	-	+	+	+	-	+	+	+	+	+	+

2.2. Методы анализа обнаружение DDOS-атак при учете сезонности

В DDOS-атаке в роли атакующего выступает так называемая бот-сеть, или зомби-сеть. Зомби- сеть может насчитывать от нескольких десятков до тысяч хостов. Обычно это нейтральные компьютеры, которые в силу каких-то причин (отсутствие файрвола, устаревшие базы антивируса и т.д.), были заражены, вредоносными программами. Программы, работая в фоновом режиме, непрерывно посылают запросы на атакуемый сервер, выводя его таким образом из строя [14].

В настоящий момент не существует какого-то универсального средства для противодействия DDOS-атакам. Даже такие крупные компании, как Microsoft, eBay, Amazon, Yahoo, страдают от DDOS-атак и не всегда могут с ними справиться. Для противодействия распределенным атакам, направленным на отказ в обслуживании, требуется выполнение двух основных задач.

1. Диагностировать DDOS-атаку на самых ранних стадиях. Чем раньше будет обнаружена DDOS-атака, тем раньше сможет включиться в игру сетевой администратор и тем раньше можно будет начать проводить анти DDOS-мероприятия. Кроме того, при обнаружении DDOS-атаки можно будет, не дожидаясь реагирования администратора, автоматически запустить мероприятия по противодействию: задействовать резервные каналы связи, включить фильтры и т.д.

2. Вторая задача связана с разделением общего потока трафика на вредоносный и обычный. Поняв, какие из клиентских запросов являются результатом DDOS-атаки, можно будет создать соответствующие правила

для межсетевого экрана или ACL правила для маршрутизатора или же, в случае масштабной атаки, передать эти данные на вышестоящие маршрутизаторы.

Первая из этих задач является достаточно новой. Несколько лет назад основной являлась именно задача по «сортировке» трафика. Однако злоумышленники постоянно совершенствуют способы проведения атак такого типа. И современные атаки отличаются сложностью и наличием этапа подготовки. Во время подготовительного этапа злоумышленник пытается выявить наиболее уязвимые для атаки места. Например, для web-сервера такими местами могут быть определенные скрипты, которые совершают большое количество запросов к базе данных или чрезмерно используют процессорное время. Для выявления этих мест злоумышленник может совершать серию мини-DDOS- атак на различные скрипты, отслеживая при этом время ответа сервера и время выполнения скрипта. Найдя уязвимое место, злоумышленник сможет парализовать работу сервера, используя бот-сеть меньшего размера. С другой стороны, если диагностировать атаку удастся уже на этом этапе, можно будет задействовать автоматические средства предотвращения атаки, а у системного администратора будет время подготовиться - оптимизировать скрипты, чрезмерно загружающие ресурсы компьютера, создать фильтры и т.д. Для обнаружения DDOS-атак и создания специальных фильтров для отсечения вредоносного трафика применяются разнообразные методы и подходы. Среди основных методов можно выделить методы, базирующиеся на статистическом анализе. Это количественный анализ, анализ среднеквадратичных отклонений, кластерный анализ и т.д. Все эти виды анализа могут оценивать различные параметры сетевой активности и диагностировать начало атаки либо определять вредоносный трафик.

Основными параметрами, по которым проводится анализ, могут быть:

- количество запросов за определенный период;

- скорость поступления запросов;
- количество запросов с определенного источника или из определенной сети;
- количество запросов к определенному пункту назначения (для web-сервера это конкретный скрипт);
- время между запросами;
- другие различные параметры сетевой активности.

С помощью среднеквадратичного отклонения можно рассчитать допустимую границу для одного из параметров сетевой активности, например, для количества запросов за какой-то период времени. В случае если граница будет нарушена, это станет свидетельством начала атаки. Так как в разное время нагрузка на сетевой ресурс, так же может быть разной, то для раннего обнаружения атаки необходим постоянный мониторинг и пересчет границ для каждого временного шага. Постоянный мониторинг позволит определить атаку, если она начнется в период небольшой сетевой активности, или, если злоумышленник ищет потенциально уязвимые места на сервере, проводя мини-DDOS- атаки и изучая поведения сервера. В случае если верхняя граница задана строго и злоумышленник проводит мини-атаки в период наименьшей сетевой активности, он может не нарушать заданную границу, и его действия будут не обнаружены. Атака будет обнаружена тогда, когда злоумышленник найдет потенциально уязвимое место, и предпримет на него атаку. Постоянный мониторинг активности и перерасчет допустимых границ позволяет этого избежать. В период меньшей сетевой активности верхняя граница снизится. Однако и этот метод имеет ряд минусов.

Во-первых, злоумышленник может начать атаку постепенно. Показатели активности на каждом шаге будут плавно повышаться, но при этом не будут нарушать границ. Так как при расчете среднеквадратичного отклонения используются последние n интервалов, в том числе и те, которые

уже содержат данные атаки, то злоумышленник, постепенно увеличивая интенсивность атаки, будет отодвигать границу.

Во-вторых, выбор размера периода n для расчета среднеквадратичного отклонения, не является однозначным.

Если n будет слишком велико, полученная граница будет слишком высоко, если используется малое значение n, то возможны частые срабатывания. Выбрать же оптимально значение n в этой ситуации будет невозможно, так как любое его значение может захватывать два разных периода. Например, даже малое значение n, рассчитываемое в начале рабочего дня, будет захватывать данные и ночного периода, связанного с низкой активностью, и дневного периода, который характеризуется большей нагрузкой на сетевые ресурсы.

Для того чтобы предотвратить ложное срабатывание, связанное с началом рабочего дня, потребуется использовать такое значение n, в котором будут данные за несколько дней. В качестве гипотезы можно предположить, что более высокую точность в этих случаях может дать учет различных периодов активности и сравнение сходных между собой периодов.

Пусть X_i количество запросов к серверу за один час. Сервер испытывает стабильную суточную нагрузку. Количество суточных периодов n. Тогда запросы к серверу можно записать в виде матрицы:

$$x_{11}, x_{12}, x_{13} \dots x_{1\ 24},$$

$$x_{21}, x_{22}, x_{23} \dots x_{2\ 24},$$

$$\dots\ \dots\ \dots\ \dots\ \dots\ \dots\ \dots\ \dots\ \dots\ \dots$$

$$x_{n1}, x_{n2}, x_{n3} \dots x_{n\ 24},$$

Каждая строка матрицы включает в себя суточные данные о количестве запросов. Первая строка отражает данные текущих суток, в этой связи она может быть заполнена не до конца. Расчет среднеквадратичного отклонения в этом случае может проводиться двумя способами:

• обычным способом с учетом определенного числа последних значений, например так:

$$x_{21}, x_{22}, x_{23} \dots x_{2\ 24}, x_{11}, x_{12}, x_{13}, x_{14}$$

Значения берутся из строк матрицы.

• с учетом сезонности. Расчет проводится по столбцам:

$$x_{n1} \dots x_{21}, x_{11}$$

Если мы находимся в i-м периоде, можно рассчитать границу для $(i+1)-$го периода, используя значение $(i+1)-$го столбца. Если сетевой ресурс испытывает нагрузку, связанную с недельными или суточными циклами, то необходимо исключить строки, которые соответствуют праздничным и выходным дням [15]. Или даже использовать только каждую седьмую строку, т.е. сравнивать, например, только период с 11:00 до 12:00, для каждого понедельника.

Проверка гипотезы. Апробация данной гипотезы проведена на реальных данных, полученных из лог-файлов различных web-сайтов, которые содержат в себе нормальные данные и данные, соответствующие DDOS-атакам. Лог-файл представляет собой стандартный файл access_log web-сервера Apache. Предварительно данные из лог-файлов были обработаны вручную и проанализированы. В результате были выделены сезонные периоды, а также точно обозначено время начала атак.

Диагностирование DDOS-атаки проводилось различными методами:

- анализ сходных сезонных периодов;
- анализ последних n периодов, при различных значениях n.
- анализ последних n периодов, различной размерности (минуты, часы и т.д.), для разных значений n.

В связи с тем, что лог-файлы имеют свой специфичный формат, их анализ стандартными методами является затруднительным. Для проведения анализа был создан скрипт, извлекающий из лог- файла необходимые данные и экспортирующий их в базу данных.

Метод анализа с учетом сезонности показал более высокую точность обнаружения DDOS- атаки и более короткое время, которое прошло с момента начала атаки до её диагностирования.

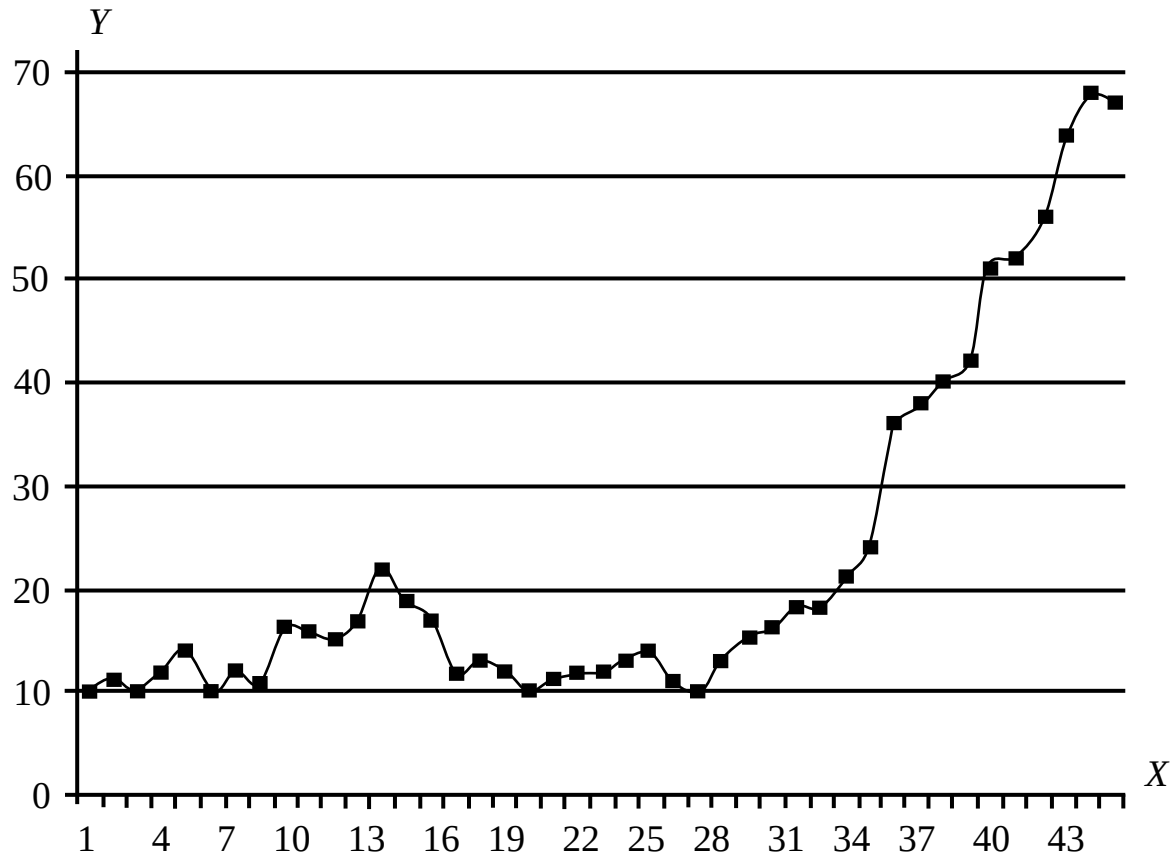

Рис.2.1. Количество запросов в период начала DDOS-атаки, 10-минутный интервал

На графике отражается количество запросов к серверу за секунду, соответствующие периоду начала DDOS-атаки. Ось *X* - временной интервал. Одно деление соответствует 10 мин. Ось *Y* – количеству запросов к серверу за секунду. На основании IP-адресов, принадлежащих компьютерам бот-сети, которые были выявлены при анализе лог-файлов, удалось точно установить момент начала атаки. На графике он соответствует 24-му периоду. Учет сезонности помог выявить DDOS-атаку уже в 31-м периоде. Другие методы показали худшие результаты. При слишком больших значениях *n* атаку удалось диагностировать только на 42-м периоде, при малых происходило

ложно срабатывание в 14-м периоде. В среднем по всем тестам время обнаружения DDOS-атаки методами с учетом сезонности, сократилось в 4 раза, так же сократилось число ложных срабатываний. Достаточно большой сложностью, возникающей при использовании данного метода, является правильный выбор сходных между собой периодов. Для апробации были выбраны данные с таких серверов, периоды работы которых однозначно определялись и не вызывали сомнения.

2.3. Разработки методики обнаружения DDoS-атак на основе системы массового обслуживания

Для распознавания атаки типа «отказ в обслуживании» оценивается вероятность потери произвольной заявки при ее прохождении по сети. Поскольку атаки прикладного уровня на различные сетевые службы происходят независимо, в рамках каждой службы для моделирования узлов СеМО можно использовать одноканальную систему массового обслуживания (СМО) с очередью длины *m*.

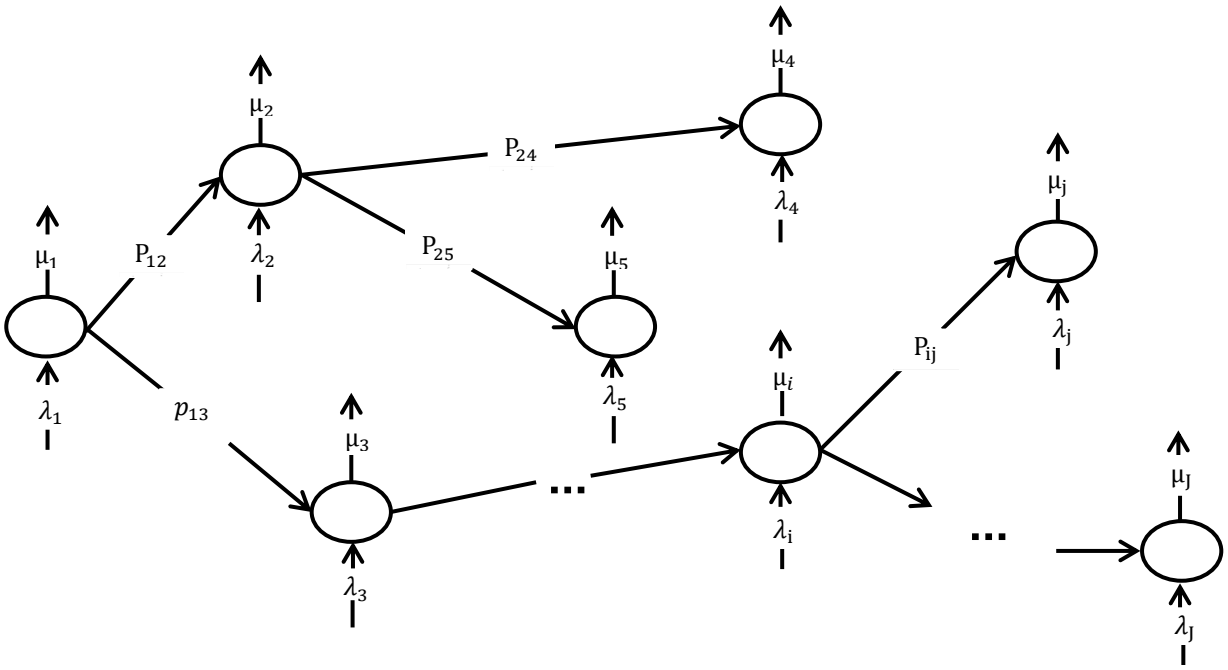

Рис.2.2. Исходная сеть массового обслуживания

Предположив, что вся сеть в целом и выбранный узел в частности функционирует в стационарном режиме, и что этот режим существует. Узел содержит одно устройство для обслуживания заявок, для которого задана интенсивность m обработки заявок. После обработки заявка покидает узел. Если во время поступления заявки обслуживающее устройство занято обработкой другой заявки, то входящая заявка становится в очередь, максимальная длина очереди т. Если заявка поступает, и очередь полностью заполнена, заявка теряется.

Исходящий из узла поток успешно обработанных заявок является пуассоновским с параметром:

$$\lambda_R = \mu(1 - P_0),\ P_0 = \frac{1}{\sum_{k=0}^{m} \left(\frac{\lambda}{\mu}\right)^k}.$$

Поток, управляющий потерями заявок в узле в стационарном режиме, также является пуассоновским с параметром:

$$\lambda_F = P_F\lambda,\ P_F = \frac{\left(\frac{\lambda}{\mu}\right)^m}{\sum_{k=0}^{m} \left(\frac{\lambda}{\mu}\right)^k}.$$

Здесь p_0 - вероятность того, что при функционировании в стационарном режиме очередь узла будет пуста, а p_F - вероятность того, что в стационарном режиме очередь узла будет полна, p_0 и p_F можно найти с помощью второй формулы Эрланга.

Далее рассматривается СеМО без потерь заявок (СеМО Джексона), состоящая из J узлов. Каждый узел содержит одно устройство для обработки заявок, время обработки одной заявки в узле j имеет экспоненциальное распределение с параметром μ_j. Значения μ_j вектора $\vec{\mu}$ положительны.

Очередь для поступающих в узел заявок не ограничена по длине. Также известна субстохастическая матрица маршрутизации P (сумма элементов по строке меньше или равна 1). Её элемент p_{ij} задает вероятность, что заявка,

которая успешно завершила обслуживание в узле номер i**,** отправится в узел j**.** Матрица P неприводима. С матрицей маршрутизации связан ориентированный граф, в котором из узла i в узел j есть ребро только в случае, когда p_{ij} отлично от нуля (рис.2.2).

Величина:

$$P_i^* = \sum_{k=1}^{J} P_{ik}\,.$$

задает вероятность, что после обработки в узле заявка покинет сеть. На вход узла сети j поступает пуассоновский поток заявок извне с параметром λ_j. Значения элементов λ_j вектора $\vec{\lambda}$ неотрицательны. Также вводится вектор $\vec{p}$.

Величины p_j – интенсивности суммарных входящих в узлы потоков. В общем случае суммарный поток не обязан быть пуассоновским, но при сделанных предположениях о сети он таковым является. Если матрица (I - P) обратима (это ограничение эквивалентно требованию ||P||<1), то:

$$\vec{p} = \vec{\lambda}(I - P)^{-1}$$

Рассмотрено узел сети массового обслуживания j, не передающий заявок другим узлам. Этот узел заменяется узлом сети Джексона с аналогичными макро характеристиками в стационарном режиме (рис.2.3).

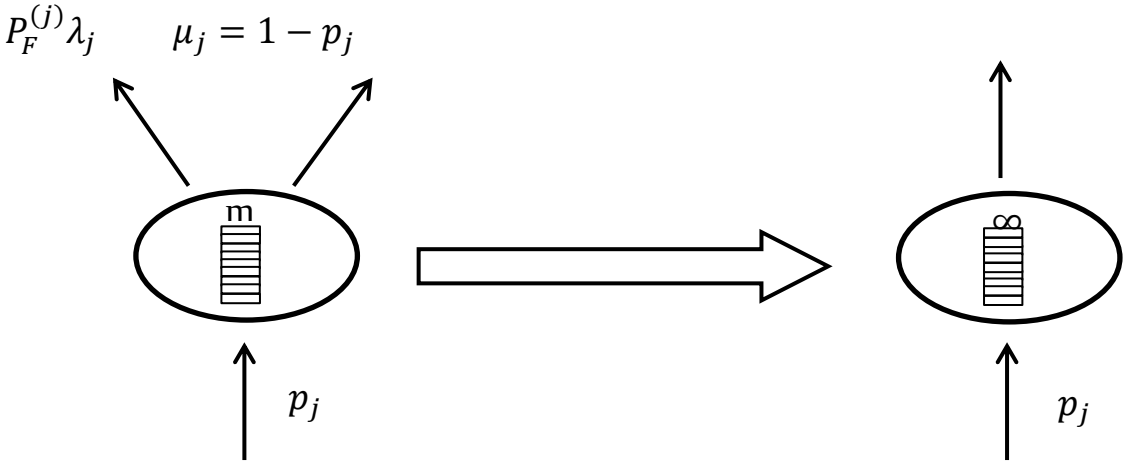

Рис.2.3. Замена узла исходной СеМО узлом сети Джексона

Т.е. принимается предположение, что успешно обработанные заявки объединяются в один поток с потерянными заявками и образуют общий поток успешно обработанных заявок в узле после замены [16].

Далее рассматривается узел исходной сети, в котором заявки не только завершают работу, но и передаются другим узлам. Вероятности перехода $\tilde{p}_{ji}$ у узла с неограниченной очередью после замены выбираются меньше (учитывается, что часть заявок выходного потока узла после замены на самом деле потеряны и не передаются в другие узлы), а вероятность успешной обработки $\tilde{p}_j^*$ - больше соответствующей вероятности p_j^*. Потенциальный путь поступившей в исходный узел заявки отображен на следующей схеме (рис.2.4):

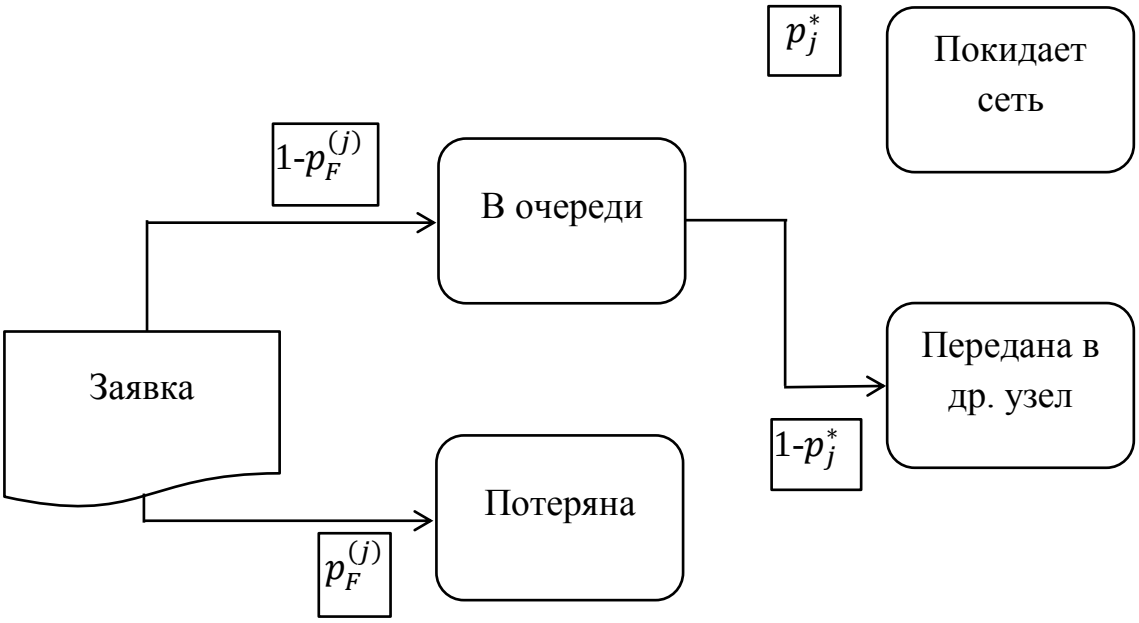

Рис.2.4. Потенциальный путь поступившей в исходный узел заявки

Далее необходимо рассмотреть вероятность $\tilde{p}_j^*$ «успешного» обработки в узле после замены (которое включает в себя и потерю заявки). Событие, отвечающее этой вероятности, состоит из потери заявки при заполненной очереди и успешной обработки и освобождения в случае постановки в очередь:

$$\tilde{p}_j^* = p_{j,F} + \left(1 - p_{j,F}\right)p_j^* = p_j^* + (1 - p_j^*)p_{j,F}$$

Далее определяются вероятности переходов в другие узлы $\tilde{p}_{ji}$. Их можно рассчитать как:

$$\tilde{p}_{ji} = \beta_j p_j i$$

где коэффициент β_j можно найти из условия нормировки:

$$\tilde{p}_{j1} + \tilde{p}_{j2} + \cdots + \tilde{p}_{jJ} + \tilde{p}^* = 1$$

$$\beta_j\left(p_{j1} + p_{j2} + \cdots p_{jJ}\right) + \tilde{p}^* = 1$$

$$\beta_j = \frac{1 - \tilde{p}^*}{p_{j1} + p_{j2} + \cdots + p_{jJ}} = \frac{1 - \tilde{p}^*}{1 - p_j^*}$$

Тогда:

$$\beta_{j=} 1 - p_{j,F}$$

После чего пересчитываются параметры стационарного режима, и происходит замена следующего узла.

В работе предложена обобщенная итерационная процедура построения скорректированной матрицы сети. Пусть задана субстохастическая матрица P**,** интенсивности входных потоков λ_j и обработки заявок в узлах μ_j. В результате итерационной процедуры строится матрица $\tilde{P} = \vec{\beta} * P$ (данное обозначение соответствует операции, в ходе которой i-ая строка матрицы P умножается на i-ый элемент вектора P). В качестве начального приближения выбирается

$$\tilde{\beta}^{(0)} = (1,1,1, \ldots ,1)$$

Шаг итерации:

1. Используя формулу (1.1) и матрицу $\tilde{P} = \vec{\beta} * P$, рассчитываются текущие интенсивности входящих потоков в узлы:

$$\vec{\rho}^{(k)} = \vec{\lambda}(I - \beta \vec{}\wedge k * P)^{-1} \quad (2.1)$$

2. Рассчитываются элементы вектора вероятностей потерь $\tilde{p}_F$:

$$p_{j,F}^{(k)} = \frac{\left(\frac{\rho_j^{(k)}}{\mu_j}\right)^{m_j}\left(1-\frac{\rho_j^{(k)}}{\mu_j}\right)}{\left(1-\frac{\rho_j^{(k)}}{\mu_j}\right)^{m_j}}$$

3. Определяется результат итерации по формуле (2.2):

$$\beta_j^{(k+1)} = 1 - p_{j,F}^{(k)}$$

4. Переход на следующую итерацию.

Итерации продолжаются, пока $\|\vec{\beta}^{(k+1)} = 1 - \vec{\beta}^{(k)}\| > \varepsilon$, где ε - заданная точность. В работе доказана сходимость предложенной итерационной процедуры.

Пусть теперь известен вектор интенсивностей входящих в узлы исходной сети потоков $\vec{\rho}$ (суммарные потоки извне и из других узлов), полученный с помощью расчета стационарного распределения соответствующей сети Джексона. Также задан вектор интенсивностей входящих потоков заявок извне $\vec{\lambda}$, вектор интенсивностей обработки заявок $\vec{\mu}$, и субстохастическая матрица маршрутизации после коррекции $\widetilde{\mathrm{P}}$. В работе предложена методика построения цепи Маркова с дискретным временем, соответствующая пути произвольной заявки по узлам, для оценки вероятности потерь заявок в сети.

Для этого вводятся расщепленные состояния этой цепи, т.е. состоянием называется упорядоченная пара чисел (id)=s. Первое число соответствует номеру узла, в котором находится заявка (меняется в пределах от 1 до J), а второе - количеству занятых мест в очереди узла (меняется в пределах от 1 до m_i+1). Состояния вида $(i, m_i + 1)$, соответствуют переполненной очереди в узле i . Начальное распределение P можно рассчитать как:

$$\hat{p}\{s = (i,d)\} = p_{\lambda_i} . p_{\pi,i,d-1} \quad (2.3)$$

Здесь Pi - вероятность того, что заявка пришла извне именно в узел i, принимаем ее равной:

$$p_{\lambda_i} = \frac{\lambda_i}{\sum_{j=1}^{J} \lambda_j} \qquad (2.4)$$

$p_{\lambda_i}.p_{\pi,i,d}$ - вероятность того, что в очереди узла i находятся d заявок, эту вероятность можно рассчитать, используя формулу (2.5), заменяя λ_i на ρ_i:

$$p_{\pi,i,d} = \frac{\frac{\rho_i^{d-1}}{\mu_i^{d-1}}\left(1-\frac{\rho_i}{\mu_i}\right)}{1-\frac{\rho_i^{m_i}}{\mu_i^{m_i}}} \qquad (2.5)$$

Подставляя (2.5) и (2.4) в (2.3), можно получить выражение для начального распределения:

$$\hat{p}\{s=(i,d)\} = \frac{\lambda_i}{\sum_{j=1}^{J} \lambda_j} \cdot \frac{\frac{\rho_i^{d-1}}{\mu_i^{d-1}}\left(1-\frac{\rho_i}{\mu_i}\right)}{1-\frac{\rho_i^{m_i}}{\mu_i^{m_i}}} \qquad (2.6)$$

Кроме того, вводятся два дополнительных состояния (*S*) и (*F*). Первое соответствует успешной обработке заявки, а второе - потере заявки. Начальные вероятности этих состояний равны 0.

Далее необходимо определить вероятности переходов (матрица $\hat{P}$). Прежде всего, надо отметить, что состояния (S) и (F) не сообщаются. Цепь, попав в одно из этих состояний, уже из него не выходит. Вероятность перехода из состояния (i,d) в состояние (j,w) можно рассчитать следующим образом:

$$p\{(i,d) \to (j,w)\} = \hat{p}_{ij} p_{\pi,j,w-1},$$

$$p\{(i,d) \to (j,w)\} = \hat{p}_{ij} \frac{\frac{\rho_i^{w-1}}{\mu_i^{w-1}}\left(1-\frac{\rho_i}{\mu_i}\right)}{1-\frac{\rho_i^{m_i}}{\mu_i^{m_i}}}$$

Вероятность успешной обработки заявки в узле равна:

$$p\{(i,d) \to (S)\} = \tilde{p}_i^*$$

Если заявка находится в переполненной очереди, вероятность потери (перехода в состояние) *(F*) будет равна 1:

$$p\{(i, m_i + 1) \to (F)\} = 1$$

Все остальные вероятности равны 0.

Если возвести матрицу P в степень k и умножить справа начальное распределение (6) на результат возведения в степень:

$$\hat{p}^{(k)} = \hat{p}(\hat{p})^k$$

то член вектора $\hat{p}^{(k)}\{(F)\}$, соответствующий состоянию (F), будет являться оценкой вероятности потери заявки при стационарном режиме работы сети.

В случае, когда сеть имеет произвольную структуру, установка параметра k происходит с помощью дополнительной итерационной процедуры. На первом шаге процедуры устанавливается некоторое начальное значение k , строится цепь Маркова и рассчитывается распределение ее состояний после k переходов заявки по узлам[17]. По результатам расчёта вычисляется $\hat{\mathrm{p}}_{\mathrm{N}}^{(\mathrm{k})}$ - вероятность того, что заявка всё еще находится в сети, т.е. не потеряна и не обработана полностью:

$$\hat{p}_N^{(k)} 1 - \hat{p}^{(k)}\{(F)\} - \hat{p}^{(k)}\{(S)\}$$

Если в результате вычислений $\hat{\mathrm{p}}_{\mathrm{N}}^{(\mathrm{k})}$ превышает некоторую наперед заданную точность, k увеличивается, и расчет повторяется до тех пор, пока не будет достигнута заданная точность указанной вероятности.

2.4. Исследование разработанной методики обнаружения DDoS-атак

Моделирующий модуль содержит объекты двух видов - источники заявок и узлы сети. Источник заявок генерирует новые объекты заявок и добавляет их в очередь соответствующего узла. Начиная с момента запуска, узел проверяет наличие в очереди заявок. В процессе обработки заявки сохраняется вся информация о ее маршруте по сети массового обслуживания,

что позволяет по завершению процесса моделирования точно рассчитать наблюдаемую частоту потери заявок. Функционирование объектов модели СМО реализовано в параллельном режиме на основе библиотеки Open MP, генерация псевдослучайных чисел осуществляется при помощи библиотеки Boost. Отдельно оценивалась теоретическая вероятность потери заявки, с помощью описанной выше итерационной процедуры и построения цепи Маркова, соответствующей пути заявки в системе. По итогам реализации программно-вычислительного комплекса для имитационного моделирования и расчёта характеристик сетей массового обслуживания была произведена серия вычислительных экспериментов для различных логических сетевых топологий. Ниже приводится описание эксперимента для топологии «произвольный ориентированный граф», исходные параметры СеМО, граф (рис.2.6), определяемый матрицей переходов, рассчитанные теоретические оценки параметров СеМО, а также практические результаты моделирования.

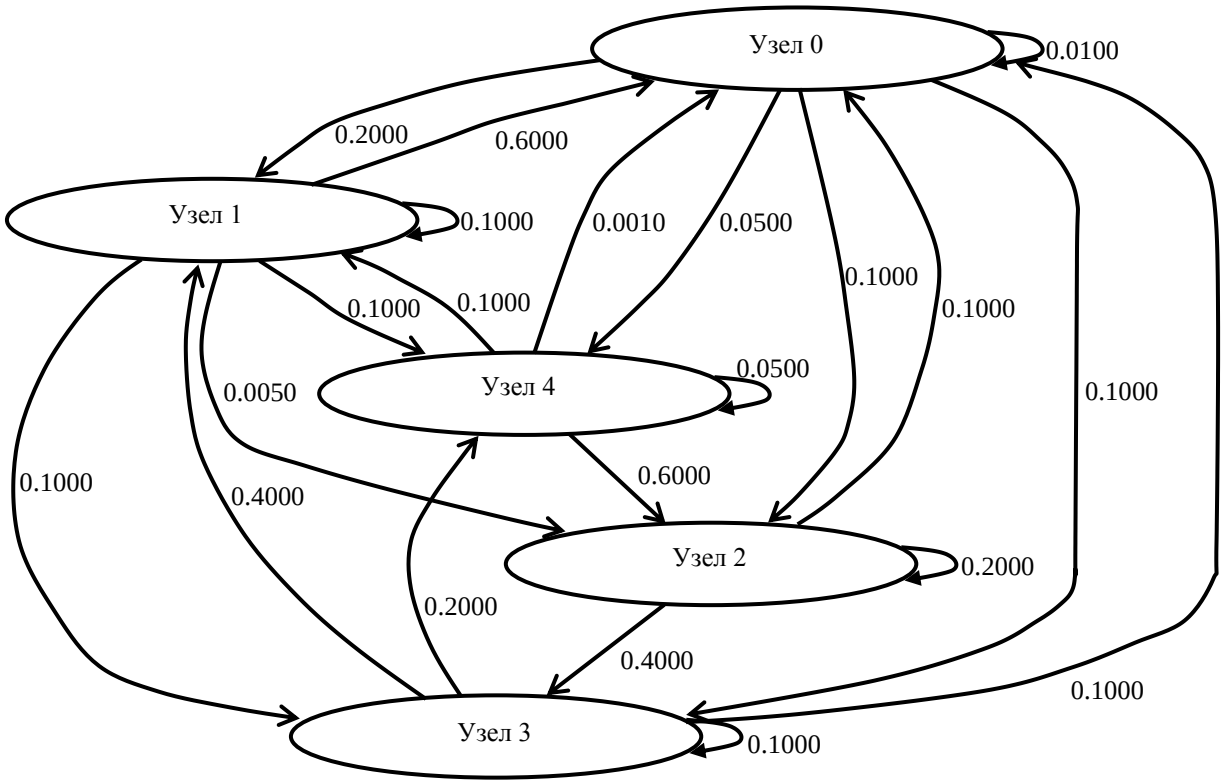

Рис.2.5. Моделируемая сеть типа «произвольный ориентированный граф»

В рамках каждого эксперимента производилось 10 сеансов имитационного моделирования сетевого взаимодействия.

Матрица P и вектора $\vec{m}$,$\vec{\lambda}$, $\vec{\mu}$, соответствующие этой сети, имеют вид:

$$P = \begin{pmatrix} 0.01 & 0.2 & 0.1 & 0.1 & 0.05 \\ 0.6 & 0.1 & 0.005 & 0.1 & 0.1 \\ 0.1 & 0 & 0.2 & 0.4 & 0 \\ 0.1 & 0.4 & 0 & 0.1 & 0.2 \\ 0.001 & 0.1 & 0.6 & 0 & 0.005 \end{pmatrix}, \quad \vec{m} = \begin{pmatrix} 5 \\ 4 \\ 8 \\ 9 \\ 5 \end{pmatrix}; \quad \vec{\lambda} = \begin{pmatrix} 2.8 \\ 0.9 \\ 2.1 \\ 1.0 \\ 0.5 \end{pmatrix}; \quad \vec{\mu} = \begin{pmatrix} 8.1 \\ 10.1 \\ 8.1 \\ 7.3 \\ 9.3 \end{pmatrix};$$

По результатам наблюдений можно отметить, что средняя ошибка рассчитанной оценки вероятности потерь заявок в СеМО не превышает среднеквадратичное отклонение наблюдаемой частоты потерь в серии экспериментов (табл.2.2).

Таблица 2.2

Результаты имитационного моделирования заданной сети

Количество сеансов моделирования	q	10
Ср. знач. наблюдаемой частоты потерь заявок	$p_{\text{ср.}}$	9,42E-2
Среднеквадратичное отклонение частоты потерь	$\sigma(p)$	1,08E-3
Рассчитанная оценка вероятности потерь заявок	$p_{рассч.}$	9,38E-2
Средняя ошибка	$\Delta_{\text{ср.}}(p)$	4,0E-4

На рис.2.6. представлен график зависимости теоретической оценки вероятности потери произвольной заявки в СеМО от количества шагов предложенной итерационной процедуры. Кроме того, на графике отмечена наблюдаемая частота потерь заявок в моделируемой сети. Представленный график позволяет сделать вывод о необходимом количестве итераций для обеспечения заданной точности [18].

Разработанная методика позволяет получать адекватную оценку частоты потери заявок в сети в случае, если сеть массового обслуживания работает в стационарном режиме. Во время возникновения DDoS-атаки один или несколько узлов СеМО выходят из стационарного режима на некоторое

время, после чего устанавливается стационарный режим с другими параметрами. На время перехода между режимами рассматриваемая методика неприменима. Таким образом, оценка времени перехода между режимами имеет определяющее значение. Очевидно, что время перехода сильно зависит от топологии сети и параметров узлов.

Рис.2.6. Зависимость теоретической оценки вероятности потери заявки от количества шагов итерационной процедуры

В течение сеанса моделирования СеМО, спустя некоторое время после запуска один из узлов производил атаку типа «отказ в обслуживании» на заполнение пропускной способности канала. Общий интервал времени моделирования был разбит на 30 равных интервалов, для каждого из которых оценивалась теоретическая вероятность потери заявок и наблюдаемая

частота потерь заявок. При максимальном размере заявки (пакета) 8760 байт (размер окна, максимальное количество данных, которое может быть отправлено по протоколу TCP/IP без подтверждения) и пропускной способности каналов 100 Мбит/сек, время перехода между стационарными режимами варьируется от 10 минут (при средней загруженности сети на уровне 30%) до 35 минут (при средней загруженности сети на уровне 10%).

Оценка эффективности разработанной методики обнаружения DDoS-атак в компьютерных сетях и её сравнительный анализ с другими подходами, методами и системами представляет сложную задачу. Основные трудности оценки и анализа заключаются в следующем:

- эффективность обнаружения атак типа «отказ в обслуживании» напрямую зависит от параметров работы сети в штатном режиме (загрузка сети, среднее значение потерь пакетов/запросов), при этом воспроизведение параметров работы для двух различных экспериментов не всегда является возможным;

- разрабатываемые методы и системы обладают рядом настраиваемых индивидуальных параметров (точность вычислений, значение порога принятия решения об обнаружении атаки), существенно влияющих на эффективность обнаружения атак и количество ложных срабатываний;

- кроме того, для каждой разновидности DDoS-атаки существует множество различных модификаций и параметров (интенсивность атаки, уникальные идентификаторы), которые также непосредственно влияют на эффективность обнаружения атаки;

- все вышеперечисленные факторы обуславливают отсутствие единого стандарта экспериментальных условий для оценки эффективности систем и методов обнаружения атак типа «отказ в обслуживании».

В ходе поставленного эксперимента ботнет, образованный группой из 4 узлов нарушителей в одной подсети, производит атаки типа «отказ в обслуживании» (SYN-флуд, ICMP-флуд, UDP-флуд) на узел-жертву в другой

подсети (рис.2.7). В задачи эксперимента входила оценка точности обнаружения атак типа «отказ в обслуживании» с помощью разработанной методики.

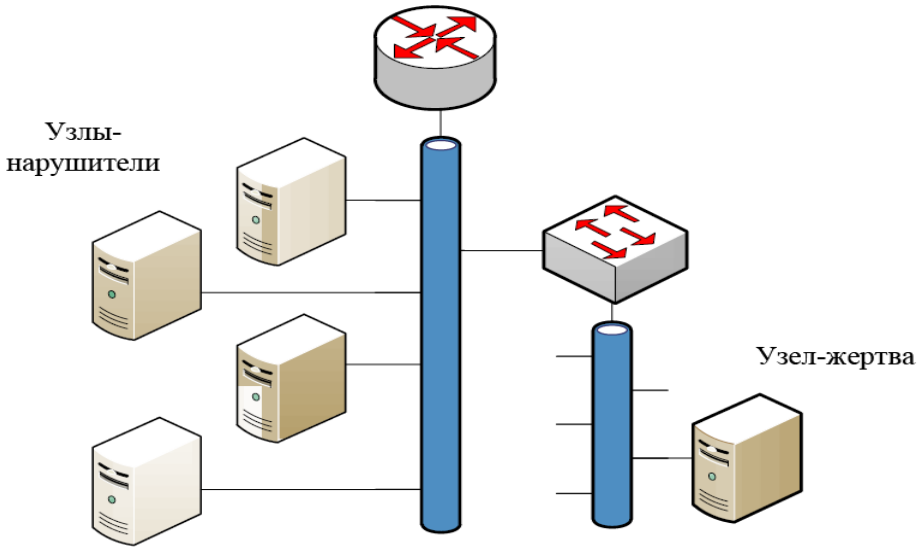

Рис.2.7. Топология сети для проведения экспериментальных исследований

Для решения поставленной задачи ровно 1000 раз в течение заданного временного интервала (10 сек.) происходило накопление передаваемых пакетов в штатном режиме работы сети, после чего ровно 1000 раз происходило накопление передаваемых пакетов на узле под воздействием различных атак типа «отказ в обслуживании». Интенсивность атак варьировалась случайным образом в интервале от 500 пакетов/сек. до 15000 пакетов/сек. В результате обозначенного эксперимента для разработанной методики обнаружения DDoS-атак были получены следующие результаты для ошибок первого рода (количество ложных срабатываний, 11,6%) и ошибок второго рода (количество необнаруженных атак, 3,4%).

Таблица 2.3

Значения ошибок обнаружения DDoS-атак первого рода и второго рода для различных систем

Система	Ошибка первого рода, %	Ошибка второго рода, %

Kaspersky Anti-Hacker 1.8.180	0	10,6
Snort	4,8	10,5
AIDS	3,5	7,24

Таким образом, разработанная методика демонстрирует более высокий процент обнаружения атак типа «отказ в обслуживании» по сравнению с уже исследованными системами, что представляет наибольшую практическую важность. Вместе с тем процент ложных срабатываний для разрабатываемой методики также максимален, что в определенной степени является её недостатком.

3. Безопасность жизнедеятельности

3.1. Защита людей от поражения электрическим током при работе на оборудовании и с электрооборудованием

Электрическими называются установки, в которых вырабатывается, преобразуется и распределяется электрическая энергия. Электроустановки могут быть напряжением до 1000 В и выше, постоянными и временными, размещаться на открытом воздухе или в закрытых помещениях.

Эксплуатация и ремонт электроустановок связаны с опасностью поражения работающих электрическим током, а также возникновения взрывов и пожаров. При эксплуатации и ремонте электрического оборудования человек может оказаться под напряжением электрического тока. Поражение человека электрическим током происходит при соприкосновении с оголенными токоведущими частями электроустановок (повреждение изоляции) или при неправильном переключении в них.

Тело человека является проводником электрического тока и имеет переменное сопротивление, которое зависит от состояния организма, физиологических факторов, параметров электрической цепи и характера окружающей среды. Проходя через тело человека, электрический ток оказывает термическое, электрическое и биологическое действие.

Термическое действие тока выражается в ожогах отдельных участков тела, нагреве до высокой температуры кровеносных сосудов, сердца, нервов, мозга и других органов, что вызывает в них серьезные функциональные расстройства. Электролитическое действие тока проявляется в разложении крови и других органических жидкостей, которое сопровождается значительными нарушениями их физико-химического состава. Биологическое действие тока выражается в раздражении и возбуждении живых тканей организма, что сопровождается непроизвольными судорожными сокращениями мышц, в том числе мышц

сердца и легких. В результате могут возникнуть различные нарушения в организме, в том числе нарушение и даже полное прекращение деятельности органов дыхания и кровообращения. Раздражающее действие тока на ткани живого организма (а следовательно, и обусловленные им непроизвольные судорожные сокращения мышц) может быть прямым, когда ток проходит непосредственно по этим тканям, а в некоторых случаях — рефлекторным, т. е. через центральную нервную систему, когда путь тока лежит вне этих тканей. Все многообразие действий электрического тока на организм вызывает два вида поражения: электрические травмы и электрические удары.

Электрические травмы - это ярко выраженные местные повреждения тканей организма, вызванные действием электрического тока или электрической дуги. Различают следующие электрические травмы: электрические ожоги, электрические знаки, металлизация кожи, механические повреждения и электроофтальмия.

Электрические ожоги могут быть вызваны протеканием тока непосредственно через тело человека при соприкосновении его с токоведущей частью (контактный ожог), а также воздействием электрической дуги на тело (дуговой ожог). В первом случае ожог является следствием преобразования энергии электрического тока, протекающего через пораженный участок тела, в тепловую. Такой вид ожога возникает редко и характеризуется обычно I или II степенями, т. е. является сравнительно легким (покраснение кожи, образование пузырей). Ожоги, вызванные электрической дугой,- наиболее распространенный вид электротравмы. Они обусловлены высоким напряжением и высокой температурой дуги (свыше 3500°) и носят, как правило, тяжелый характер: III и IV степень (омертвение всей толщи кожи и обугливание тканей).

Электрические знаки представляют собой четко очерченные пятна серого или бледно-желтого цвета на поверхности тела человека, подвергшегося действию тока. Обычно знаки имеют круглую или овальную

форму с углублением в центре и размеры 1—5 мм. Бывают знаки в виде царапин, небольших ран, порезов или ушибов, Пораженный участок кожи затвердевает подобно мозоли.

Как правило, электрические знаки безболезненны и излечиваются благополучно: с течением времени верхний слой кожи сходит, и пораженное место приобретает первоначальный цвет, эластичность и чувствительность.

Металлизация кожи - проникновение в верхние слои кожи мельчайших частичек металла, расплавившегося под действием электрической дуги. Это может произойти при коротких замыканиях, отключениях разъединителей и рубильников под нагрузкой и пр. Обычно с течением времени больная кожа сходит, пораженный участок принимает нормальный вид, исчезают и болезненные ощущения. Механические повреждения являются следствием резких непроизвольных судорожных сокращений мышц под действием тока, проходящего через человека. В результате могут произойти разрывы кожи, сухожилий, кровеносных сосудов и нервной ткани, а также вывихи суставов и даже переломы костей. Механические повреждения являются серьезными травмами, требующими длительного лечения. Они возникают примерно у 0,5% лиц, пострадавших от электрического тока вследствие относительно длительного нахождения под напряжением.

Электроофтальмия - воспаление наружных оболочек глаз (покраснение и воспаление кожи и слизистых оболочек век, слезотечение, гнойные выделения из глаз, а в тяжелых случаях - нарушение прозрачности роговой оболочки) под воздействием электрической дуги.

Электрический удар - возбуждение живых тканей организма проходящим через него электрическим током, сопровождающееся непроизвольными судорожными сокращениями мышц. В зависимости от исхода воздействия электрического тока на организм человека электрические удары можно условно разделить на четыре степени:

I — судорожное сокращение мышц без потери сознания;

II — судорожное сокращение мышц с потерей сознания, но с сохранившимся дыханием и работой сердца;

III — потеря сознания и нарушение сердечной деятельности или дыхания (либо того и другого вместе);

IV— клиническая смерть, т. е. отсутствие дыхания и кровообращения.

Клиническая («мнимая») смерть - переходное состояние организма от жизни к смерти, наступающее с момента прекращения деятельности сердца и легких. У человека, находящегося в состоянии клинической смерти, отсутствуют все признаки жизни: он не дышит, сердце его не работает, болевые раздражения не вызывают никаких реакций, зрачки глаз расширены и не реагируют на свет. Однако в этот период в организме еще полностью жизнь не угасла, ибо ткани его отмирают не все сразу и не сразу угасают функции различных органов. В первый момент почти во всех тканях организма продолжаются обменные процессы, хотя и на очень низком уровне и резко отличающиеся от обычных, но достаточный для поддержания минимальной жизнедеятельности. Первыми начинают погибать очень чувствительные к кислородному голоданию клетки коры головного мозга, с деятельностью которых связаны сознание и мышление. Поэтому длительность клинической смерти определяется временем с момента прекращения сердечной деятельности и дыхания до начала гибели клеток головного мозга. В большинстве случаев она составляет 4—6 мин, а при гибели здорового человека от случайной причины, например от электрического тока, примерно 7—8 мин.

Биологическая (истинная) смерть - необратимое явление, характеризующееся прекращением биологических процессов в клетках и тканях организма и распадом белковых структур. Она наступает по истечении периода клинической смерти.

Эти обстоятельства позволяют, воздействуя на более стойкие жизненные функции организма, восстановить угасающие или только угасшие функции, т. е. оживить умирающий организм.

Степень воздействия электрического тока на организм человека различна и зависит от ряда факторов: рода, частоты, силы тока и длительности его воздействия, напряжения сети, сопротивления тела человека и индивидуальных особенностей его организма.

По степени опасности поражения людей напряжение, применяемое в электрических установках, классифицируется на три вида: низковольтное - 12 и 42 В, низкое - от 42 до 1000 В и высокое - свыше 1000 В.

Условно безопасным напряжением считается низковольтное, которое в зависимости от характера среды тоже может представлять опасность.

Величина (сила) тока, протекающего через тело человека, является главным фактором, от которого зависит исход поражения: чем больше ток, тем опаснее его воздействие. Человек начинает ощущать протекающий через него ток с частотой 50 Гц и относительно малого значения: 0,5—1,5 мА. Этот ток называется пороговым током. Он не может вызвать поражения человека и в этом смысле является безопасным. Однако такой ток может стать косвенной причиной несчастного случая, поскольку человек, почувствовав воздействие тока, теряет уверенность в своей безопасности и может допустить неправильные действия.

При увеличении тока возрастают болезненные ощущения. Ток в 10-15 мА (при 50 Гц) вызывает непроизвольные болезненные сокращения мышц, рук (судороги), которые человек не в состоянии преодолеть, т. е. он не может разжать руку, прикасающуюся к токоведущей части, отбросить провод от себя и оказывается как бы прикованным к токоведущей части. Такой ток называется пороговым неотпускающим. Сам по себе он не угрожает жизни, но если человек немедленно не будет освобожден от токоведущих частей, то

с течением времени по мере увеличения тока вследствие понижения сопротивления тела человек погибнет.

При 25-50 мА действие тока распространяется и на мышцы грудной клетки, что приводит к затруднению и даже прекращению дыхания. Одновременно происходит сужение кровеносных сосудов и, как следствие этого, повышение артериального давления и ослабление деятельности сердца. Ток в 100 мА оказывает непосредственное воздействие на мышцу сердца, вызывая его остановку или фибрилляцию, т. е. быстрые хаотические и разновременные сокращения волокон сердечной мышцы (фибрилл), при которых сердце перестает работать как насос. В результате прекращается кровообращение в организме, и наступает клиническая смерть.

Если пораженному в течение ближайших 5—7 минут не будет оказана своевременная медицинская помощь (искусственное дыхание), он погибнет вследствие кислородного голодания, т. е. наступит биологическая смерть.

Электрическое сопротивление различных тканей тела человека неодинаково. Так, кожа, точнее ее наружный слой, называемым эпидермисом, имеет толщину от 0,1 до 0,5 мм, состоит в основном из мертвых ороговевших клеток, обладает большим сопротивлением (сухая, чистая, неповрежденная), которое и определяет общее сопротивление тела человека - от 3000 до 100000 Ом. Сопротивление внутренних тканей тела человека незначительно - примерно 300—500 Ом.

При увлажнении и загрязнении кожи снижается ее сопротивление и возрастает опасность поражения электрическим током; при повреждении кожи сопротивление тела оказывается наименьшим —300—500 Ом. Сопротивление тела уменьшается с увеличением силы тока и времени его протекания вследствие увеличения потовыделения и других факторов. При расчетах среднее сопротивление тела человека принимается равным 1000 Ом.

Род и частота тока в значительной степени определяют степень поражения. Сопротивление тела человека постоянному току больше, чем

переменному. Переменный ток с частотой от 20 до 1000 Гц наиболее опасен. При частоте меньше или выше 1000 Гц опасность тока снижается.

При постоянном токе порог ощущения повышается до 6—7 мА, а неотпускающий ток — до 60—70 мА. Токи частотой более 500000 Гц не оказывают раздражающего действия на ткани и потому не вызывают электрического удара, однако они опасны тем, что могут вызвать термические ожоги. Индивидуальные особенности человека - состояние здоровья, подготовленность к работе в электрической установке и другие факторы (повышенная температура окружающего воздуха - до 30-45°С) - также влияют на исход поражения. Поэтому обслуживание электроустановок поручается лицам, прошедшим специальное обучение и медицинский осмотр.

Меры защиты от поражения электрическим током. Анализ несчастных случаев в промышленности, сопровождающихся временной утратой трудоспособности, показывает, что число травм, вызванных электрическим током, сравнительно невелико и составляет 0.5—1% от общего числа несчастных случаев на производстве.

Совершенно иная картина наблюдается, если рассматривать только смертельные несчастные случаи. При этом оказывается, что из общего числа смертельных несчастных случаев на производстве 20—40% их происходит в результате поражения электрическим током, что, как правило, больше, чем по какой-либо другой причине. Вот почему вопросам электробезопасности на производстве необходимо уделять большое внимание. 75—80% смертельных поражений электрическим током происходит в электроустановках напряжением до 1000 В и, в первую очередь, в установках 127 и 220 В. Объясняется это весьма широким распространением этих установок на производстве. Основными мерами защиты от поражения электрическим током являются:

- обеспечение недоступности токоведущих частей, находящихся под напряжением, для случайного прикосновения;
- устранение опасности поражения электрическим током при появлении напряжения на корпусах, кожухах и других частях электрооборудования путем защитного заземления, зануления, защитного отключения, применения малых напряжений, защитного разделения сетей, применения двойной изоляции, выравнивания потенциала и др.;
- применение индивидуальных защитных средств при электротехнических работах.

Недоступность токоведущих частей электроустановок для случайного прикосновения может быть обеспечена их изоляцией, размещением на недоступной высоте, ограждением и др.

3.2. Пожарная безопасность

Противопожарная защита имеет своей целью изыскание наиболее эффективных, экономически целесообразных и технически обоснованных способов и средств предупреждения пожаров и их ликвидации с минимальным ущербом при наиболее рациональном использовании сил и технических средств тушения.

Пожарная безопасность – это состояние объекта, при котором исключается возможность пожара, а в случае его возникновения используются необходимые меры по устранению негативного влияния опасных факторов пожара на людей, сооружения и материальных ценностей

Пожарная безопасность может быть обеспечена мерами пожарной профилактики и активной пожарной защиты. Пожарная профилактика включает комплекс мероприятий, направленных на предупреждение пожара или уменьшение его последствий. Активная пожарная защита – меры, обеспечивающие успешную борьбу с пожарами или взрывоопасной

ситуацией. Пожар – это горение вне специального очага, которое не контролируется и может привести к массовому поражению и гибели людей, а также к нанесению экологического, материального и другого вреда.

Горение – это химическая реакция окисления, сопровождающаяся выделением теплоты и света. Для возникновения горения требуется наличие трех факторов: горючего вещества, окислителя и источника загорания. Окислителями могут быть кислород, хлор, фтор, бром, йод, окиси азота и другие. Кроме того, необходимо чтобы горючее вещество было нагрето до определенной температуры и находилось в определенном количественном соотношении с окислителем, а источник загорания имел определенную энергию. Наибольшая скорость горения наблюдается в чистом кислороде. При уменьшении содержания кислорода в воздухе горение прекращается. Горение при достаточной и над мерной концентрации окислителя называется полным, а при его нехватке – неполным.Выделяют три основных вида самоускорения химической реакции при горении: тепловой, цепной и цепочно-тепловой. Тепловой механизм связан с экзотермичностью процесса окисления и возрастанием скорости химической реакции с повышением температуры. Цепное ускорение реакции связано с катализом превращений, которое осуществляют промежуточные продукты превращений. Реальные процессы горения осуществляются, как правило, по комбинированному (цепочно-тепловой) механизму. Процесс возникновения горения подразделяется на несколько видов:

1. Вспышка – быстрое сгорание горючей смеси, не сопровождающееся образованием сжатых газов.

2. Возгорание – возникновение горения под воздействием источника зажигания.

3. Воспламенение – возгорание, сопровождающееся появлением пламени.

4. Самовозгорание – явление резкого увеличения скорости экзотермических реакций, приводящее к возникновению горения вещества при отсутствии источника зажигания.

5. Различают несколько видов самовозгорания:

6. Химическое – от воздействия на горючие вещества кислорода, воздуха, воды или взаимодействия веществ;

7. Микробиологическое – происходит при определенной влажности и температуры в растительных продуктах (самовозгорание зерна);

8. Тепловое – вследствие долговременного воздействия незначительных источников тепла (например, при температуре 100 С тирса, ДВП и другие склоны к самовозгоранию).

Температура воспламенения – температура горения вещества, при которой оно выделяет горючие пары и газы с такой скоростью, что после воспламенения их от источника зажигание возникает устойчивое горение.

Температурные пределы воспламенения – температуры, при которых насыщенные пары вещества образуют в данной окислительной среде концентрации, равные соответственно нижнему и верхнему концентрационным пределам воспламенения жидкостей. Горючими называются вещества, способные самостоятельно гореть после изъятия источника загорания. По степени горючести вещества делятся на: горючие (сгораемые), трудногорючие (трудносгораемые) и негорючие (несгораемые).

К горючим относятся такие вещества, которые при воспламенении посторонним источником продолжают гореть и после его удаления. К трудногорючим относятся такие вещества, которые не способны распространять пламя и горят лишь в месте воздействия источника зажигания. Негорючими являются вещества, не воспламеняющиеся даже при воздействии достаточно мощных источников зажигания (импульсов).

Горючие вещества могут быть в трех агрегатных состояниях: жидком, твердом и газообразном. Большинство горючих веществ независимо от

агрегатного состояния при нагревании образует газообразные продукты, которые при смешении с воздухом, содержащим определенное количество кислорода, образуют горючую среду. Горючая среда может образоваться при тонкодисперсном распылении твердых и жидких веществ. Пожар на предприятии наносит большой материальный ущерб народному хозяйству и очень часто сопровождается несчастными случаями с людьми.

Основными причинами, способствующими возникновению и развитию пожара, являются:

- нарушение правил применения и эксплуатации приборов и оборудования с низкой противопожарной защитой;
- использование при строительстве в ряде случаев материалов, не отвечающих требованиям пожарной безопасности;
- отсутствие на многих объектах народного хозяйства и в подразделениях пожарной охраны эффективных средств борьбы с огнем.

Мероприятия по пожарной профилактике разделяются на организационные, технические, режимные, строительно-планировочные и эксплуатационные. Организационные мероприятия: предусматривают правильную эксплуатацию машин и внутризаводского транспорта, правильное содержание зданий, территории, противопожарный инструктаж и тому подобное.

Режимные мероприятия – запрещение курения в неустановленных местах, запрещение сварочных и других огневых работ в пожароопасных помещениях и тому подобное.

Эксплуатационные мероприятия – своевременная профилактика, осмотры, ремонты и испытание технологического оборудования.

Строительно-планировочные определяются огнестойкостью зданий и сооружений (выбор материалов конструкций: сгораемые, несгораемые, трудносгораемые) и предел огнестойкости — это количество времени, в

течение которого под воздействием огня не нарушается несущая способность строительных конструкций вплоть до появления первой трещины.

Все строительные конструкции по пределу огнестойкости подразделяются на 8 степеней от 1/7 ч до 2ч.

В зависимости от степени огнестойкости наибольшие дополнительные расстояния от выходов для эвакуации при пожарах

В практике тушения пожаров наибольшее распространение получили следующие принципы прекращения горения:

- изоляция очага горения от воздуха или снижение концентрации кислорода путем разбавления воздуха негорючими газами (углеводы CO2 < 12–14%).
- охлаждение очага горения ниже определенных температур;
- интенсивное торможение (ингибирование) скорости химической реакции в пламени; механический срыв пламени струей газа или воды;
- создание условий огнепреграждения (условий, когда пламя распространяется через узкие каналы).

Вещества, которые создают условия, при которых прекращается горение, называются огнегасящими. Они должны быть дешевыми и безопасными в эксплуатации не приносить вреда материалам и объектам.

Вода является хорошим огнегасящим средством, обладающим следующими достоинствами: охлаждающее действие, разбавление горючей смеси паром (при испарении воды ее объем увеличивается в 1700 раз), механическое воздействие на пламя, доступность и низкая стоимость, химическая нейтральность.

Недостатки: нефтепродукты всплывают и продолжают гореть на поверхности воды; вода обладает высокой электропроводностью, поэтому ее нельзя применять для тушения пожаров на электроустановках под напряжением.

Тушение пожаров водой производят установками водяного пожаротушения, пожарными автомашинами и водяными стволами. Для подачи воды в эти установки используют водопроводы. К установкам водяного пожаротушения относят спринклерные и дренчерные установки.

Спринклерная установка представляет собой разветвленную систему труб, заполненную водой и оборудованную спринклерными головками. Выходные отверстия спринклерных головок закрываются легкоплавкими замками, которые распаиваются при воздействии определенных температур (345, 366, 414 и 455 К). Вода из системы под давлением выходит из отверстия головки и орошает конструкции помещения и оборудование. Дренчерные установки представляют собой систему трубопроводов, на которых расположены специальные головки–дренчеры с открытыми выходными отверстиями диаметром 8, 10 и 12,7 мм лопастного или розеточного типа, рассчитанные на орошение до 12 м2 площади пола. Дренчерные установки могут быть ручного и автоматического действия. После приведения в действие вода заполняет систему и выливается через отверстия в дренчерных головках. Пар применяют в условиях ограниченного воздухообмена, а также в закрытых помещениях с наиболее опасными технологическими процессами. Гашение пожара паром осуществляется за счет изоляции поверхности горения от окружающей среды. При гашении необходимо создать концентрацию пара приблизительно 35 %. Пены применяют для тушения твердых и жидких веществ, не вступающих во взаимодействие с водой. Огнегасящий эффект при этом достигается за счет изоляции поверхности горючего вещества от окружающего воздуха. Огнетушащие свойства пены определяются ее кратностью – отношением объема пены к объему ее жидкой фазы, стойкостью дисперсностью, вязкостью. В зависимости от способа получения пены делят на химические и воздушно-механические.

Химическая пена образуется при взаимодействии растворов кислот и щелочей в присутствии пенообразующего вещества и представляет собой концентрированную эмульсию двуокиси углерода в водном реакторе минеральных солей. Применение химических солей сложно и дорого, поэтому их применение сокращается. Воздушно-механическую пену низкой (до 20), средней (до 200) и высокой (свыше 200) кратности получают с помощью специальной аппаратуры и пенообразователей ПО–1, ПО–1Д, ПО–6К и т.д. Инертные газообразные разбавители: двуокись углерода, азот, дымовые и отработавшие газы, пар, аргон и другие.

Порошковые составы несмотря на их высокую стоимость, сложность в эксплуатации и хранении, широко применяют для прекращения горения твердых, жидких и газообразных горючих материалов. Они являются единственным средством гашения пожаров щелочных металлов и металлоорганических соединений. Для гашения пожаров используется также песок, грунт, флюсы. Порошковые составы не обладают электропроводимостью, не коррозируют металлы и практически не токсичны. Широко используются составы на основе карбонатов и бикарбонатов натрия и калия. Аппараты пожаротушения: передвижные (пожарные автомобили), стационарные установки, огнетушители.

Стационарные установки предназначены для тушения пожаров в начальной стадии их возникновения без участия человека. Подразделяются на водяные, пенные, газовые, порошковые, паровые. Могут быть автоматическими и ручными с дистанционным управлением. Огнетушители – устройства для гашения пожаров огнегасящим веществом, которое он выпускает после приведения его в действие, используется для ликвидации небольших пожаров. Как огнетушащие вещества в них используют химическую или воздухо-механическую пену, диоксид углерода (жидком состоянии), аэрозоли и порошки, в состав которых входит бром.

Заключение

Основные результаты ВКР могут быть сформулированы в следующем виде:

1. Рассмотрены способы и виды защиты сетевых атак.
2. Предложены подходы, основанные на стохастических контекстно-свободных формальных грамматиках, для моделирования атак и построения графов атак и на логическом программировании, для построения графов атак.
3. Исследовано дерево атак, представляющим собой методологию описания угроз и мер противодействия для защиты систем.
4. Использованы агрегация подобных хостов как для улучшения наглядности графа атак, так и для повышения производительности.
5. Предложена архитектура топологического сканера безопасности.
6. Описаны методы обнаружение атак на сетевом и системном уровне.
7. Произведен анализ и классификация существующих атак типа «отказ в обслуживании», обобщены и систематизированы основные подходы к обнаружению атак данного типа в распределенных компьютерных сетях.
8. Разработана итерационная процедура аппроксимации сети массового обслуживания с конечными очередями сетью Джексона для вычисления интенсивностей входящих в узлы потоков заявок (для различных топологий исходной сети).
9. Предложена методика построения цепи Маркова на основе характеристик сети массового обслуживания для оценки вероятности потерь заявок в сети.
10. Разработана методика обнаружения низкоактивных атак типа «отказ в обслуживании» на основе их моделирования в сетях массового обслуживания и оценки вероятности потерь в стационарном режиме функционирования сети.
11. Произведена экспериментальная оценка эффективности и сравнительный анализ разработанного программно-аналитического

комплекса для обнаружения атак типа «отказ в обслуживании» в компьютерных сетях.

Использованные литературы

1. Постановление Президента Республики Узбекистан «О дополнительных мерах по дальнейшему развитию информационно-коммуникационных технологий». от 21 марта 2012 года, ПП – 1730.
2. Лукацкий А. В. Обнаружение атак. 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2012. – 608 с.
3. Лукацкий А.В. «Системы обнаружения атак на сетевом уровне» PC Week/RE № (207)33`1999 от 7.9.1999 стр. 14
4. Зима В. М. Молдовян А. А., Молдовян Н. А. Безопасность глобальных сетевых технологий. 2-е изд. – СПб.: БХВ-Петербург, 2011. – 368 с.
5. Danforth M. Models for Threat Assessment in Networks. – http://www.cs.ucdavis.edu/research/tech- reports/2006/CSE-2006-13.pdf
6. Stephenson P. Using formal methods for forensic analysis of intrusion events – a preliminary examination. – http://www.imfgroup.com/Document Library.html.
7. Amenaza. A Quick Tour of Attack Tree Based Risk Analysis Using. http:// www.amenaza.com.
8. Cuppens F. Alert Correlation in a Cooperative Intrusion Detection Framework // Proc. of the 2002 IEEE Symposium on Security and Privacy. – 2009.
9. Соколов А. В., Степанюк О. М. Методы и средства защиты объектов и компьютерных сетей – М.: ООО «Фирма «Издательство АСТ», 2010. – 272 с.
10. Устройства для защиты объектов и информации./В.И.Андрианов, А.В.Соколов СПб.: ООО «Издательство Полигон», 2005.- 256 с.
11. В. Г. Олифер, Н. А. Олифер. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 2-е изд. – СПб.: Питер, 2003. – 864 с.

12. DDOS-атаки [Электронный ресурс].-Режим доступа: http://localname.ru/soft/ataki-tipa-otkaz- v-obsluzhivanii-dos-i-raspredelennyiy-otkaz-v-obsluzhivanii-ddos.html, свободный (дата обращения: 24.04.2012).

13. Предотвращение атак с распределенным отказом в обслуживании (DDoS) Официальный сайт компании Cisco [Электронный ресурс]. - Режим доступа:http://www.cisco.com/web/RU/products/ps5887/products_white_paper0900aecd8011e927_.html, свободный (дата обращения: 24.04.2012).

14. Методы защиты от DDOS нападений [Электронный ресурс]. - Режим доступа: http://www.securitylab.ru/analytics/216251.php, свободный (дата обращения: 24.04.2012).

15. Щерба М.В. Методика разработки системы защиты информации комплекса муниципальных информационных систем / М.В. Щерба // Информационные технологии моделирования и управления. – 2009. – Выпуск 6(58). – С. 850-854.

16. Терновой О.С. Раннее обнаружение DDOS-атак методами статистического анализа / Перспективы развития информационных технологий. - Новосибирск: Сибпринт, 2012. - С. 201-212.

17. Боровков А.А. Математическая статистика. Оценка параметров проверки гипотез. - М.: Наука. 1984. - 280 с.

18. Бенкен Е.С. PHP, MySQL, XML. Программирование для Интернета. - СПб.: БХВ-Петербург, 2011. - С. 336.

19. www.opennet.ru

20. Вишняков Я.Д., Вагин В.И., Овчинников В.В., Стародубец А.Н. Безопасность жизнедеятельности. Защита населения и территорий в чрезвычайных ситуациях. Уч. Пособие. Москва. 2007.

21. Безопасность жизнедеятельности. Учебник для ВУЗов. С.В. Белов, А.В. Ипьницкая, А.Ф. Козьяков и др. Под общей ред. С.В. Белова. М. Высшая школа. 1999 г.

Приложение

```
unit Unit Main;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, ActnList, StdActns, Menus, StdCtrls, ComCtrls, ExtCtrls, Mask,
  UnitThreadSniffer, UnitPileTrame, UnitGlobal;
type
  TMainForm = class(TForm)
    PanelBottom: TPanel;
    Btn_StartStop: TButton;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Ecr: TRichEdit;
    FiltreACK: TCheckBox;
    FiltrePSH: TCheckBox;
    FiltreRST: TCheckBox;
    FiltreSYN: TCheckBox;
    FiltreFIN: TCheckBox;
    procedure FormCreate(Sender: TObject);
    procedure Btn_StartStopClick(Sender: TObject);
    procedure TimerInDataTimer(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
  end;
end;
procedure TMainForm.Btn_StartStopClick(Sender: TObject);
begin
  Start:= not Start;
```

```
        if Start then
        begin
          if TousLesPorts.Checked then
            FThreadSniffer:=
TThreadSniffer.Create(FPileTrameTCP,EditIP.Text,$1,$FFFF)
          else
            FThreadSniffer:=
TThreadSniffer.Create(FPileTrameTCP,EditIP.Text,strtoint(StartPort.Text),
       strtoint(EndPort.Text));
        end
        else FThreadSniffer.Terminate;
      end;
      begin
        result:= str;
        if length(result)<NbCar then result:= DupeString(' ', Nbcar-
Length(result))+result;
      end;
      // On Timer
      procedure TMainForm.TimerInDataTimer(Sender: TObject);
      var
        TrameTcp    : PTrameTCP;
        Str         : string;
        StrDate     : string;
        StrIp       : string;
        StrDrapeaux : string;
        FiltreOk    : boolean;
      begin
        // Tant que la pile des messages TCP n'est pas vide
        while FPileTrameTCP.Count > 0 do
```

```
begin
  // On рйсурире le message TCP formatй par nos soins
  TrameTcp:= FPileTrameTCP.Pop;
  // Si messsage existe !!! en principe n'est jamais a nil
  if TrameTcp<>nil then
  begin
    StrDrapeaux:= '';
    FiltreOk:= false;
    // Construction chaine drapeaux TCP pour affichage
    // et teste filtre par la meme occasion
    // URG
    if drp_URG in TrameTcp^.EtatDrapeaux then
    begin
      StrDrapeaux:= StrDrapeaux + 'URG ';
      FiltreOk:= FiltreOk or FiltreUrg.Checked;
    end
    else StrDrapeaux:= StrDrapeaux + '    ';
    // ACK
    if drp_ACK in TrameTcp^.EtatDrapeaux then
    begin
      StrDrapeaux:= StrDrapeaux + 'ACK ';
      FiltreOk:= FiltreOk or FiltreACK.Checked;
    end
    else StrDrapeaux:= StrDrapeaux + '    ';
    // PSH
    if drp_PSH in TrameTcp^.EtatDrapeaux then
    begin
      StrDrapeaux:= StrDrapeaux + 'PSH ';
      FiltreOk:= FiltreOk or FiltrePSH.Checked;
```

```
end
else StrDrapeaux:= StrDrapeaux + '    ';
// RST
if drp_RST in TrameTcp^.EtatDrapeaux then
begin
  StrDrapeaux:= StrDrapeaux + 'RST ';
  FiltreOk:= FiltreOk or FiltreRST.Checked;
end
else StrDrapeaux:= StrDrapeaux + '    ';
// SYN
if drp_SYN in TrameTcp^.EtatDrapeaux then
begin
  StrDrapeaux:= StrDrapeaux + 'SYN ';
  FiltreOk:= FiltreOk or FiltreSYN.Checked;
end
else StrDrapeaux:= StrDrapeaux + '    ';
// FIN
if drp_FIN in TrameTcp^.EtatDrapeaux then
begin
  StrDrapeaux:= StrDrapeaux + 'FIN ';
  FiltreOk:= FiltreOk or FiltreFIN.Checked;
end
else StrDrapeaux:= StrDrapeaux + '    ';
// Si Filtre sur drapeaux TCP est actif on construit le reste
if FiltreOk then
begin
  // Construction chaine finale visu
  str:= TrameTcp^.Data;
  // cr lf pour affichage
```

```
            if length(str)>1 then
              if copy(Str,length(Str)-1,2)=#13#10 then
                str:= copy(Str,1, length(Str)-2);
            // Formate date time
            if VoirDate.Checked
              then StrDate:= FormatDateTime('dd/mm/yyyy hh:mm:ss ',
TrameTcp^.TimeStamp)
              else StrDate:='';
            // Formate ip et port
            if VoirIpPort.Checked then
            begin
           TrameTcp^.Source.Port,
          VisuStr(TrameTcp^.Destination.Ip,15),
          TrameTcp^.Destination.Port]);
            end
            else StrIp:='';
            // Formate Drapeaux TCP
            if not VoirDrapeaux.Checked then StrDrapeaux:= '';
        procedure TMainForm.StartPortKeyPress(Sender: TObject; var Key: Char);
        begin
          // Traite backspace
          if Key=#8 then exit;
          // Autres touches
          if (Key<'0') or (Key >'9') then Key:= #0;
        end;
        procedure TMainForm.TouslesportsClick(Sender: TObject);
        begin
          StartPort.Enabled := false;
          EndPort.Enabled   := false;
```

```
end;
procedure TMainForm.PortStartPortEndClick(Sender: TObject);
begin
  end.
```