О. И. Шелухин, Д. Ж. Сакалема, А. С. Филинова

ОБНАРУЖЕНИЕ ВТОРЖЕНИЙ В КОМПЬЮТЕРНЫЕ СЕТИ

[сетевые аномалии]

Горячая линия–Телеком

О. И. Шелухин
Д. Ж. Сакалема
А. С. Филинова

ОБНАРУЖЕНИЕ ВТОРЖЕНИЙ

В КОМПЬЮТЕРНЫЕ СЕТИ

[сетевые аномалии]

Даны основные определения и понятия в области систем обнаружения вторжений и компьютерных атак. Рассмотрены принципы построения и структура систем обнаружения вторжений. Анализируются способы развертывания, достоинства и недостатки существующих систем обнаружения вторжений.

Центральное место в книге уделено методам обнаружения сетевых аномалий. Рассмотрены методы кратномасштабного вейвлет- и мультифрактального анализа алгоритмов обнаружения аномальных вторжений. Проведен анализ статистических, интеллектуальных, иммунных, нейросетевых и других алгоритмов обнаружения аномалий.

Для студентов, обучающихся по направлению подготовки бакалавров и магистров 210700 – «Инфокоммуникационные технологии и системы связи», может быть полезно аспирантам и студентам, обучающимся по группе специальностей направления «Информационная безопасность» и специалистам в области защиты информации и безопасности инфокоммуникаций.

САЙТ ИЗДАТЕЛЬСТВА:
www.techbook.ru

ISBN 978-5-9912-0278-7

ПРЕДИСЛОВИЕ

Важнейшим атрибутом нашего времени является глобальная информационная интеграция, основанная на построении компьютерных сетей масштаба предприятия и их объединении посредством Интернета.

Сложность логической и физической организации современных сетей приводит к объективным трудностям при решении вопросов управления и защиты сетей. В процессе эксплуатации компьютерных сетей администраторам приходится решать две главные задачи:

- диагностировать работу сети и подключенных к ней серверов, рабочих станций и соответствующего программного обеспечения;
- защищать информационные ресурсы сети от несанкционированной деятельности хакеров, воздействий вирусов, сетевых червей и т. п., т. е. обеспечивать их конфиденциальность, целостность и доступность.

При решении задач, связанных с диагностикой и защитой сетевых ресурсов, центральным вопросом является оперативное обнаружение состояний сети, приводящих к потере полной или частичной ее работоспособности, уничтожению, искажению или утечке информации, являющихся следствием отказов, сбоев случайного характера или результатом получения злоумышленником несанкционированного доступа к сетевым ресурсам, проникновения сетевых червей, вирусов и других угроз информационной безопасности. Раннее обнаружение таких состояний позволит своевременно устранить их причину, а также предотвратит возможные катастрофические последствия.

Для их обнаружения используется большой спектр специализированных систем. Так, при решении проблем диагностики сетей применяются средства систем управления, анализаторы сетевых протоколов, системы нагрузочного тестирования, системы сетевого мониторинга. Проблемы защиты информационных ресурсов сетей решаются с помощью межсетевых экранов (firewall), антивирусов, систем обнаружения атак (СОВ) (Intrusion Detection System, IDS), систем контроля целостности, криптографических средств защиты.

Характерными особенностями использования этих систем является либо их периодическое и кратковременное применение для решения определенной проблемы, либо постоянное использование, но со статическими настройками. В результате методы анализа, используемые в современных системах, направлены на обнаружение известных и точно описанных типов воздействий, но зачастую оказываются не в состоянии обнаружить их модификации или новые типы, что делает их использование малоэффективным.

Таким образом, на сегодняшний день очень актуальной задачей является поиск более эффективных методов выявления недопустимых событий (аномалий) в работе сети, являющихся следствием технических сбоев или несанкционированных воздействий. Основным требованием к этим методам является возможность обнаружения произвольных типов аномалий, в том числе новых, а также воздействий, распределенных во времени.

Это направление научных исследований является очень молодым. Первые работы, посвященные данной проблеме, были опубликованы в 90-х годах прошлого столетия.

В настоящий момент исследования в этой области ведутся крупными зарубежными коммерческими компаниями. Общий подход, лежащий в основе этих исследований, заключается в поиске методов анализа, позволяющих выявлять аномальные состояния информационных ресурсов в виде отклонений от обычного («нормального») состояния. Эти отклонения могут являться результатами сбоев в работе аппаратного и программного обеспечения, а также следствиями сетевых атак хакеров. Такой подход теоретически позволит обнаруживать как известные, так и новые типы проблем. От эффективности и точности аппарата, определяющего «нормальное» состояние и фиксирующего отклонение, зависит в целом эффективность решения вопросов диагностики и защиты сетевых ресурсов. Особую важность на текущий момент представляет проблема обнаружения аномальных состояний в работе сети, имеющих распределенный во времени характер (APB). APB могут являться следствиями специально маскируемых сетевых атак злоумышленников, скрытых аппаратно-программных сбоев, новых вирусов и т. п.

В основу учебного пособия положен курс лекционных, практических и лабораторных занятий для студентов МТУСИ, обучающихся по магистерским программам «Программно-защищенные инфокоммуникации» в рамках направлений «Инфокоммуникационные технологии и системы связи», а также «Информатика и вычислительная техника».

1 КОМПЬЮТЕРНЫЕ АТАКИ

1.1. Основные определения и понятия

Атакой на информационную систему называются преднамеренные действия злоумышленника, использующие уязвимости информационной системы и приводящие к нарушению доступности, целостности и конфиденциальности обрабатываемой информации [1].

Устранение уязвимости информационной системы приводит к устранению и самой возможности реализации атак.

Существует три типа атак:

Разведка. Эти атаки включают ping sweeps, передачу DNS-зоны, разведку с помощью e-mail, сканирование TCP или UDP-портов и, возможно, анализ общественно доступных серверов с целью нахождения cgi-дыр.

Эксплойт (exploit — использовать в своих интересах, злоупотреблять). Это компьютерная программа, фрагмент программного кода или последовательность команд, использующие уязвимости в программном обеспечении и применяемые для проведения атаки на вычислительную систему [9]. Целью атаки может быть как захват контроля над системой (повышение привилегий), так и нарушение её функционирования (DoS-атака). Нарушители будут использовать преимущества скрытых возможностей или ошибок для получения несанкционированного доступа к системе.

Отказ в обслуживании (Denial of Service, DoS). При такой атаке нарушитель пытается разрушить сервис (или компьютер), перегрузить сеть, перегрузить центральный процессор или переполнить диск.

Модели атак. Традиционная модель атаки строится по принципу «один к одному» или «один ко многим», т. е. атака исходит из одного источника. Разработчики сетевых средств защиты (межсетевых экранов, систем обнаружения атак и т. д.) ориентированы именно на традиционную модель атаки. В различных точках защищаемой сети устанавливаются агенты (сенсоры) системы защиты,

которые передают информацию на центральную консоль управления. Это облегчает масштабирование системы, обеспечивает простоту удаленного управления и т. д. Однако такая модель не справляется с распределенными атаками.

В модели распределенной атаки используются иные принципы. В отличие от традиционной модели в распределенной модели используются отношения «многие к одному» и «многие ко многим».

Распределенные атаки основаны на «классических» атаках типа «отказ в обслуживании», а точнее на их подмножестве, известном как Flood- или Storm-атаки (указанные термины можно перевести как «шторм», «наводнение» или «лавина»). Смысл данных атак заключается в посылке большого количества пакетов на атакуемый узел. Атакуемый узел может выйти из строя, поскольку он «захлебнется» в лавине посылаемых пакетов и не сможет обрабатывать запросы авторизованных пользователей. Однако в случае, если пропускная способность канала до атакуемого узла превышает пропускную способность атакующего или атакуемый узел некорректно сконфигурирован, к «успеху» такая атака не приведет. Но распределенная атака происходит уже не из одной точки Internet, а сразу из нескольких, что приводит к резкому возрастанию трафика и выведению атакуемого узла из строя.

Получили распространение следующие типы атак:

удаленное проникновение (remote penetration). Атаки, которые позволяют реализовать удаленное управление компьютером через сеть. Например, NetBus или BackOrifice;

локальное проникновение (local penetration). Атака, которая приводит к получению несанкционированного доступа к узлу, на котором она запущена. Например, GetAdmin;

удаленный отказ в обслуживании (remote denial of service). Атаки, которые позволяют нарушить функционирование или перегрузить компьютер через Internet. Например, Teardrop или trin00;

локальный отказ в обслуживании (local denial of service). Атаки, которые позволяют нарушить функционирование или перегрузить компьютер, на котором они реализуются. Примером такой атаки является «враждебный» апплет, который загружает центральный процессор бесконечным циклом, что приводит к невозможности обработки запросов других приложений.

1.2. Классификация атак

Существуют различные типа классификации атак. Например, деление на пассивные и активные, внешние и внутренние, умышлен-

Таблица 1.1

Основные типы аномалий в IP-сетях

Тип аномалии	Описание	Характеристики
Альфа-аномалия	Необычно высокий уровень трафика типа точка-точка	Выброс в представлении трафика байты/с, пакеты/с по одному доминирующему потоку источник — назначение. Небольшая продолжительность (до 10 минут)
DoS, DDoS атака	Распределенная атака типа отказ в обслуживании на одну жертву	Выброс в представлении трафика пакеты/с, потоки/с, от множества источников к одному адресу назначения
Перегрузка	Необычно высокий спрос на один сетевой ресурс или сервис	Скачок в трафике по потокам/с к одному доминирующему IP-адресу и доминирующему порту. Обычно кратковременная аномалия
Сканирование сети/портов	Сканирование сети по определенным открытым портам или сканирование одного хоста по всем портам с целью поиска уязвимостей	Скачок в трафике по потокам/с, с несколькими пакетами в потоках от одного доминирующего IP-адреса
Деятельность червей	Вредоносная программа, которая самостоятельно распространяется по сети и использует уязвимости операционных систем	Выброс в трафике без доминирующего адреса назначения, но всегда с одним или несколькими доминирующими портами назначения
Точка-мультиточка	Распространение контента от одного сервера многим пользователям	Выброс в пакетах, байтах от доминирующего источника к нескольким назначениям, все к одному (одним) хорошо известным портам
Отключения	Сетевые неполадки, которые вызывают падение в трафике между одной парой источник-назначение	Падение в трафике по пакетам, потокам и байтам обычно до нуля. Может быть долговременным и включать все потоки источник-назначение от или к одному маршрутизатору
Переключения потока	Необычное переключение потоков трафика с одного входящего маршрутизатора на другой	Падение в байтах или пакетах в одном потоке трафика и выброс в другом. Может затрагивать несколько потоков трафика

ные и неумышленные. Самым естественным и явным образом можно классифицировать существующие актуальные аномалии по типу источника или причины их возникновения. Пример такой классификации приведен на рис 1.1.

В табл. 1.1 представлены основные типы сетевых аномалий, их описание и основные характеристики. Приведенная систематизация данные об атаках и этапах их реализации дает необходимый базис для понимания технологий обнаружения атак.

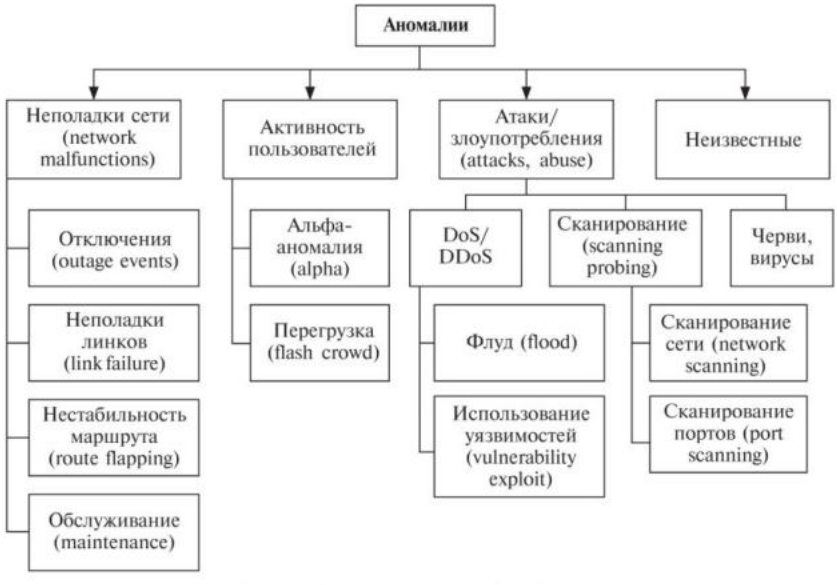

Рис. 1.1. Классификация аномалий в IP-сетях по типу

1.3. Этапы реализации атак

Можно выделить следующие этапы реализации атаки:

- предварительные действия перед атакой или «сбор информации»;
- собственно «реализация атаки»;
- завершение атаки.

Обычно, когда говорят об атаке, подразумевают именно второй этап, забывая о первом и последнем. Сбор информации и завершение атаки («заметание следов») в свою очередь также могут являться атакой и могут быть разделены на три этапа.

1.3.1. Сбор информации

Сбор информации — основной этап реализации атаки. Именно на данном этапе эффективность работы злоумышленника является залогом «успешности» атаки. Сначала выбирается цель атаки и собирается информация о ней:

- тип и версия операционной системы;
- открытые порты и запущенные сетевые сервисы;
- установленное системное и прикладное программное обеспечение и его конфигурация и т. д.).

Затем идентифицируются наиболее уязвимые места атакуемой системы, воздействие на которые приводит к нужному злоумыш-

леннику результату. Злоумышленник пытается выявить все каналы взаимодействия цели атаки с другими узлами. Это позволит не только выбрать тип реализуемой атаки, но и источник ее реализации. Например, атакуемый узел взаимодействует с двумя серверами под управлением ОС Unix и Windows NT. С одним сервером атакуемый узел имеет доверенные отношения, а с другим — нет. От того, через какой сервер злоумышленник будет реализовывать нападение, зависит, какая атака будет задействована, какое средство реализации будет выбрано и т. д. Затем, в зависимости от полученной информации и желаемого результата, выбирается атака, дающая наибольший эффект. Например, атака SYN-Flood. Сеансовый уровень отвечает за процедуру установления начала сеанса и подтверждение (квитирование) прихода каждого пакета от отправителя получателю. В Интернете протоколом сеансового уровня является протокол TCP (он занимает 4 и 5 уровни модели OSI). В отношении сеансового уровня очень широко распространена специфичная атака класса «отказ в сервисе», основанная на свойствах процедуры установления соединения в протоколе TCP. Она получила название SYN-Flood (flood — большой поток).

При попытке клиента подключиться к серверу, работающему по протоколу TCP (а его используют более 80 % информационных служб, в том числе HTTP, FTP, SMTP, POP3), он посылает серверу пакет без информации, но с битом SYN, установленным в 1 в служебной области пакета — запросом на соединение. По получении такого пакета сервер обязан выслать клиенту подтверждение приема запроса, после чего с третьего пакета начинается собственно диалог между клиентом и сервером. Одновременно сервер может поддерживать в зависимости от типа сервиса от 20 до нескольких тысяч клиентов.

При атаке типа SYN-Flood злоумышленник начинает на своей ЭВМ создавать пакеты, представляющие собой запросы на соединение (т. е. SYN-пакеты) от имени произвольных IP-адресов (возможно даже несуществующих) на имя атакуемого сервера по порту сервиса, который он хочет приостановить. Все пакеты будут доставляться получателю, поскольку при доставке анализируется только адрес назначения. Сервер, начиная соединение по каждому из этих запросов, резервирует под него место в своем буфере, отправляет пакет-подтверждение и начинает ожидать третьего пакета клиента в течение некоторого промежутка времени (1...5 секунд). Пакет-подтверждение уйдет по адресу, указанному в качестве ложного отправителя в произвольную точку Интернета и либо не найдет ад-

ресата вообще, либо чрезмерно «удивит» операционную систему на этом IP-адресе (поскольку она никаких запросов на данный сервер не посылала) и будет просто проигнорирован. А вот сервер, при достаточно небольшом потоке таких запросов, будет постоянно держать свой буфер заполненным ненужными ожиданиями соединений, и даже SYN-запросы от настоящих легальных пользователей не будут помещаться в буфер; сеансовый уровень просто не знает и не может узнать, какие из запросов фальшивые, а какие настоящие и могли бы иметь больший приоритет.

Традиционные средства защиты, такие, как межсетевые экраны или механизмы фильтрации в маршрутизаторах, вступают в действие лишь на втором этапе реализации атаки, совершенно «забывая» о первом и третьем. Это приводит к тому, что зачастую совершаемую атаку очень трудно остановить даже при наличии мощных и дорогих средств защиты. Пример тому — распределенные атаки. Логично было бы, чтобы средства защиты начинали работать еще на первом этапе, т. е. предотвращали бы возможность сбора информации об атакуемой системе. Это позволило бы если и не полностью предотвратить атаку, то хотя бы существенно усложнить работу злоумышленника. Традиционные средства также не позволяют обнаружить уже совершенные атаки и оценить ущерб после их реализации, т. е. не работают на третьем этапе реализации атаки. Следовательно, невозможно определить меры по предотвращению таких атак впредь.

В зависимости от желаемого результата нарушитель концентрируется на том или ином этапе реализации атаки. Например, для отказа в обслуживании подробно анализируется атакуемая сеть, в ней выискиваются лазейки и слабые места; для хищения информации основное внимание уделяется незаметному проникновению на атакуемые узлы при помощи обнаруженных ранее уязвимостей.

1.3.2. Основные механизмы реализации атак

Первый этап реализации атак — это сбор информации об атакуемой системе или узле. Он включает такие действия как определение сетевой топологии, типа и версии операционной системы атакуемого узла, а также доступных сетевых и иных сервисов и т. п. Эти действия реализуются различными методами.

Изучение окружения. На этом этапе нападающий исследует сетевое окружение вокруг предполагаемой цели атаки. К таким областям, например, относятся узлы Internet-провайдера «жертвы» или узлы удаленного офиса атакуемой компании. На этом этапе

злоумышленник может пытаться определить адреса «доверенных» систем (например, сеть партнера) и узлов, которые напрямую соединены с целью атаки (например, маршрутизатор ISP) и т. д. Такие действия достаточно трудно обнаружить, поскольку они выполняются в течение достаточно длительного периода времени и снаружи области, контролируемой средствами защиты (межсетевыми экранами, системами обнаружения атак и т. п.).

Идентификация топологии сети. Существует два основных метода определения топологии сети, используемых злоумышленниками:

- изменение TTL (TTL modulation);
- запись маршрута (record route).

По первому методу работают программы traceroute для Unix и tracert для Windows. Они используют поле Time to Live («время жизни») в заголовке IP-пакета, которое изменяется в зависимости от числа пройденных сетевым пакетом маршрутизаторов. Для записи маршрута ICMP-пакета может быть использована утилита ping. Зачастую сетевую топологию можно выяснить при помощи протокола SNMP, установленного на многих сетевых устройствах, защита которых неверно сконфигурирована. При помощи протокола RIP можно попытаться получить информацию о таблице маршрутизации в сети и т. д.

Многие из этих методов используются современными системами управления (например, HP OpenView, Cabletron SPECTRUM, MS Visio и т. д.) для построения карт сети. И эти же методы могут быть с успехом применены злоумышленниками для построения карты атакуемой сети.

Идентификация узлов. Идентификация узла, как правило, осуществляется путем посылки при помощи утилиты ping команды ECHO_REQUEST протокола ICMP. Ответное сообщение ECHO_REPLY говорит о том, что узел доступен. Существуют свободно распространяемые программы, которые автоматизируют и ускоряют процесс параллельной идентификации большого числа узлов, например fping или nmap. Опасность данного метода в том, что стандартными средствами узла запросы ECHO_REQUEST не фиксируются. Для этого необходимо применять средства анализа трафика, межсетевые экраны или системы обнаружения атак.

Это самый простой метод идентификации узлов. Однако он имеет два недостатка.

1. Многие сетевые устройства и программы блокируют ICMP-пакеты и не пропускают их во внутреннюю сеть (или наоборот не

пропускают их наружу). Например, MS Proxy Server 2.0 не разрешает прохождение пакетов по протоколу ICMP. В результате возникает неполная картина. С другой стороны, блокировка ICMP-пакета говорит злоумышленнику о наличии «первой линии обороны» — маршрутизаторов, межсетевых экранов и т. д.

2. Использование ICMP-запросов позволяет с легкостью обнаружить их источник, что, разумеется, не может входить в задачу злоумышленника.

Существует еще один метод идентификации узлов — использование «смешанного» режима сетевой карты, который позволяет определить различные узлы в сегменте сети. Но он не применим в тех случаях, в которых трафик сегмента сети недоступен нападающему со своего узла, т. е. этот метод применим только в локальных сетях. Другим способом идентификации узлов сети является так называемая разведка DNS, которая позволяет идентифицировать узлы корпоративной сети при помощи обращения к серверу службы имен.

Идентификация сервисов или сканирование портов. Идентификация сервисов, как правило, осуществляется путем обнаружения открытых портов (port scanning). Такие порты очень часто связаны с сервисами, основанными на протоколах TCP или UDP. Например: открытый 80-й порт подразумевает наличие Web-сервера, 25-й порт — почтового SMTP-сервера, 31337-й — серверной части троянского коня BackOrifice, 12345-й или 12346-й — серверной части троянского коня NetBus и т. д.

Для идентификации сервисов и сканирования портов могут использоваться различные программы, в том числе и свободно распространяемые, например nmap или netcat.

Идентификация операционной системы. Основной механизм удаленного определения ОС — анализ ответов на запросы, учитывающие различные реализации TCP/IP-стека в различных операционных системах. В каждой ОС по-своему реализован стек протоколов TCP/IP, что позволяет при помощи специальных запросов и ответов на них определить, какая ОС установлена на удаленном узле.

Другой, менее эффективный и крайне ограниченный, способ идентификации ОС узлов — анализ сетевых сервисов, обнаруженных на предыдущем этапе. Например, открытый 139-й порт позволяет сделать вывод, что удаленный узел, вероятнее всего, работает под управлением ОС семейства Windows. Для определения ОС могут быть использованы различные программы, например nmap или queso.

Определение роли узла. Предпоследним шагом на этапе сбора информации об атакуемом узле является определение его роли, например выполнении функций межсетевого экрана или Web-сервера. Выполняется этот шаг на основе уже собранной информации об активных сервисах, именах узлов, топологии сети и т. п. Например, открытый 80-й порт может указывать на наличие Web-сервера, блокировка ICMP-пакета указывает на потенциальное наличие межсетевого экрана, а DNS-имя узла proxy.domain.ru или fw.domain.ru говорит само за себя.

Определение уязвимостей узла. Последний шаг — поиск уязвимостей. На этом шаге злоумышленник при помощи различных автоматизированных средств или вручную определяет уязвимости, которые могут быть использованы для реализации атаки.

1.3.3. Реализация атак

С этого момента начинается попытка доступа к атакуемому узлу. При этом доступ может быть как непосредственный, т. е. проникновение на узел, так и опосредованный, например при реализации атаки типа «отказ в обслуживании». Реализация атак в случае непосредственного доступа также может быть разделена на два этапа:

- проникновение;
- установление контроля.

Проникновение. Проникновение подразумевает под собой преодоление средств защиты периметра (например, межсетевого экрана). Реализовываться это может быть различными путями. Например, использование уязвимости сервиса компьютера, «смотрящего» наружу или путем передачи враждебного содержания по электронной почте (макровирусы) или через апплеты Java. Такое содержание может использовать так называемые «туннели» в межсетевом экране (не путать с туннелями VPN), через которые затем и проникает злоумышленник. К этому же этапу можно отнести подбор пароля администратора или иного пользователя при помощи специализированной утилиты, например L0phtCrack или Crack.

Установление контроля. После проникновения злоумышленник устанавливает контроль над атакуемым узлом. Это может быть осуществлено путем внедрения программы типа «троянский конь» (например, NetBus или BackOrifice). После установки контроля над нужным узлом и «заметания» следов злоумышленник может осуществлять все необходимые несанкционированные действия дистанционно без ведома владельца атакованного компьютера. При этом установление контроля над узлом корпоративной сети должно сохраняться и после перезагрузки операционной системы. Это может

быть реализовано путем замены одного из загрузочных файлов или вставка ссылки на враждебный код в файлы автозагрузки или системный реестр. Известен случай, когда злоумышленник смог перепрограммировать EEPROM сетевой карты и даже после переустановки ОС он смог повторно реализовать несанкционированные действия. Более простой модификацией этого примера является внедрение необходимого кода или фрагмента в сценарий сетевой загрузки (например, для ОС Novell Netware).

Цели реализации атак. Необходимо отметить, что злоумышленник на втором этапе может преследовать две цели. Во-первых, получение несанкционированного доступа к самому узлу и содержащейся на нем информации. Во-вторых, получение несанкционированного доступа к узлу для осуществления дальнейших атак на другие узлы. Первая цель, как правило, осуществляется только после реализации второй. То есть сначала злоумышленник создает себе базу для дальнейших атак и только после этого проникает на другие узлы. Это необходимо для того, чтобы скрыть или существенно затруднить нахождение источника атаки.

1.3.4. Завершение атаки

Этапом завершения атаки является «заметание следов» со стороны злоумышленника. Обычно это реализуется путем удаления соответствующих записей из журналов регистрации узла и других действий, возвращающих атакованную систему в исходное, «предатакованное» состояние.

2 ПРИНЦИПЫ ПОСТРОЕНИЯ СИСТЕМ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ

Системами обнаружения вторжений (СОВ) называют множество различных программных и аппаратных средств, объединяемых одним общим свойством — они занимаются анализом использования вверенных им ресурсов и, в случае обнаружения каких-либо подозрительных или просто нетипичных событий, способны предпринимать некоторые самостоятельные действия по обнаружению, идентификации и устранению их причин.

Но системы обнаружения вторжений лишь один из инструментов защитного арсенала и он не должен рассматриваться как замена для любого из других защитных механизмов. Защита информации наиболее эффективна, когда в интрасети поддерживается многоуровневая защита. Она складывается из следующих компонентов:

- политика безопасности интрасети организации;
- система защиты хостов в сети;
- сетевой аудит;
- защита на основе маршрутизаторов;
- межсетевые экраны;
- системы обнаружения вторжений;
- план реагирования на выявленные атаки.

Следовательно, для полной защиты целостности сети необходима реализация всех вышеперечисленных компонентов защиты, и использование многоуровневой защиты является наиболее эффективным методом предотвращения несанкционированного использования компьютерных систем и сетевых сервисов. Таким образом, система обнаружения вторжений — это одна из компонент обеспечения безопасности сети в многоуровневой стратегии её защиты.

2.1. Классификация СОВ

Существует большое число различных классификаций систем обнаружения атак, однако самой распространенной является классификация по принципу реализации:

- Host-based, т. е. обнаруживающие атаки, направленные на конкретный узел сети;
- Network-based, т. е. обнаруживающие атаки, направленные на всю сеть или сегмент сети.

Системы обнаружения атак, контролирующие отдельный компьютер, как правило, собирают и анализируют информацию из журналов регистрации операционной системы и различных приложений (Web-сервер, СУБД и т. д.). По такому принципу функционирует RealSecure OS Sensor.

Система проводит аудит системных журналов на предмет «неправильного поведения», например множественных попыток подключения к сети или попыток изменения атрибутов файлов. В указанные моменты времени может выполняться проверка контрольных сумм важных файлов для выявления фактов их изменений. Таким образом, основная задача «агента» централизованной СОВ — отслеживать внутренние процессы и сообщать о критических событиях. Агент изначально устанавливается в только что развернутой системе, имея своей целью контролировать неизменность установок хоста, и записывает важные атрибуты системных файлов. Соответственно, если на «охраняемых» станциях работают разные операционные системы (Microsoft Windows NT/2000, Sun Solaris или Linux), администратору придется устанавливать СОВ на каждую из платформ. Рекомендуемый объем оперативной памяти и производительность процессора у разных СОВ различаются.

Однако в последнее время стали получать распространение системы, тесно интегрированные с ядром ОС, тем самым предоставляя более эффективный способ обнаружения нарушений политики безопасности. Причем такая интеграция может быть реализовано двояко. Во-первых, могут контролироваться все системные вызовы ОС (так работает Entercept) или весь входящий/исходящий сетевой трафик (так работает RealSecure Server Sensor). В последнем случае система обнаружения атак захватывает весь сетевой трафик напрямую с сетевой карты, минуя операционную систему, что позволяет уменьшить зависимость от нее и тем самым повысить защищенность системы обнаружения атак.

Системы обнаружения атак уровня сети собирают информацию из самой сети, то есть из сетевого трафика. «Сетевые» СОВ располагаются в локальной сети предприятия и производят мониторинг внутрисетевого трафика в режиме реального времени на предмет соответствия происходящих процессов заранее определенным «шаблонам» (сигнатурам) атак.

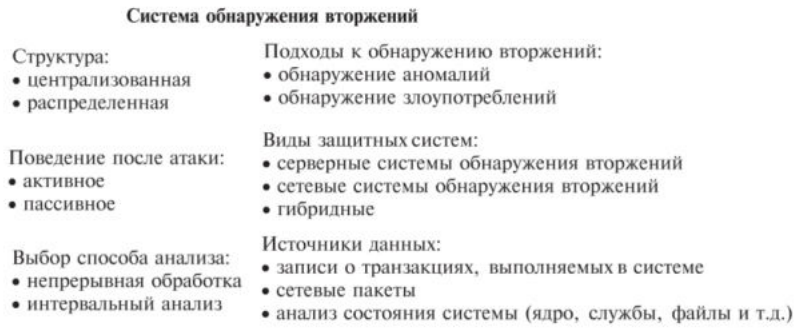

Рис. 2.1. Классификация систем обнаружения вторжений

Выполняться эти системы могут на обычных компьютерах, на специализированных компьютерах или быть интегрированными в маршрутизаторы или коммутаторы (например, CiscoSecure IOS Integrated Software или Cisco Catalyst 6000 IDS Module). В первых двух случаях анализируемая информация собирается посредством захвата и анализа пакетов, используя сетевые интерфейсы в беспорядочном (promiscuous) режиме. В последнем случае захват трафика осуществляется с шины сетевого оборудования.

Анализ сетевых атак, который показал, что наиболее целесообразно классифицировать их относительно характера воздействия. Такой подход позволил провести исследование данных атак относительно деструктивного воздействия на компьютерные сети и системы, выделяя следующие основные группы:

- несанкционированный удаленный доступ;
- несанкционированное получение привилегированных прав доступа;
- отказ в обслуживании;
- сканирование.

Получили распространение системы обнаружения сетевых вторжений, классификация и основные особенности которых представлены на рис. 2.1.

Основным сдерживающим фактором применения всех существующих методов является их ограниченное признаковое пространство, которое включает в себя четыре группы параметров:

1) основные параметры отдельных TCP-соединений: IP-адреса, порты, протоколы, количество байтов, продолжительность, количество пакетов;

2) параметры, основанные на контексте, например количество SYN пакетов;

Рис. 2.2. Характеристики систем обнаружения вторжений

3) параметры, связанные со временем, т. е. различные условные комбинации параметров в последние T секунд;

4) параметры, определяюшие соединения, т. е. различные условные комбинации параметров в последние N соединений.

Следует отметить, что эти параметры описывают только сетевую и транспортную части протокола. Хотя ряд алгоритмов используют дополнительные характеристики из прикладной части, но этого недостаточно для эффективного обнаружения аномалий протокола.

Для проведения классификации СОВ необходимо учесть несколько факторов (рис. 2.2).

Метод обнаружения описывает характеристики анализатора. Когда СОВ использует информацию о нормальном поведении контролируемой системы, она называется поведенческой. Когда СОВ работает с информацией об атаках, она называется интеллектуальной.

Поведение после обнаружения указывает на реакцию СОВ на атаки. Реакция может быть активной — СОВ предпринимает корректирующие (устраняет лазейки) или действительно активные (закрывает доступ для возможных нарушителей, делая недоступными сервисы) действия. Если СОВ только выдаёт предупреждения, её называют пассивной.

Расположение источников результата аудита подразделяет СОВ в зависимости от вида исходной информации, которую они анализируют. Входными данными для них могут быть результаты аудита, системные регистрационные файлы или сетевые пакеты.

Частота использования отражает либо непрерывный мониторинг контролируемой системы со стороны СОВ, либо соответствующие периодическим запускам СОВ для проведения анализа.

Рис. 2.3. Классификация систем обнаружения вторжений

Классифицировать СОВ можно по нескольким параметрам (рис. 2.3). По способам реагирования различают статические и динамические СОВ. *Статические* средства делают «снимки» (snapshot) среды и осуществляют их анализ, разыскивая уязвимое ПО, ошибки в конфигурациях и т. д. Статические СОВ проверяют версии работающих в системе приложений на наличие известных уязвимостей и слабых паролей, проверяют содержимое специальных файлов в директориях пользователей или проверяют конфигурацию открытых сетевых сервисов. Статические СОВ обнаруживают следы вторжения.

Динамические СОВ осуществляют мониторинг в реальном времени всех действий, происходящих в системе, просматривая файлы аудита или сетевые пакеты, передаваемые за определённый промежуток времени. Динамические СОВ реализуют анализ в реальном времени и позволяют постоянно следить за безопасностью системы.

По способу сбора информации различают *сетевые* и *системные* СОВ. Сетевые СОВ (network intrusion detection system, NIDS) контролируют пакеты в сетевом окружении и обнаруживают попытки злоумышленника проникнуть внутрь защищаемой системы или реализовать атаку «отказ в обслуживании». Эти СОВ работают с сетевыми потоками данных. Типичный пример NIDS — система, которая контролирует большое число TPC-запросов на соединение (SYN) со многими портами на выбранном компьютере, обнаруживая, таким образом, что кто-то пытается осуществить сканирование TCP-портов. Сетевая СОВ может запускаться либо на отдельном компьютере, который контролирует свой собственный трафик, либо на выделенном компьютере, прозрачно просматривающим весь трафик в сети (концентратор, маршрутизатор). Сетевые СОВ контролируют много компьютеров, тогда как другие СОВ контролируют только один.

Среди преимуществ использования NIDS можно выделить следующие моменты:

- NIDS можно полностью скрыть в сети таким образом, что злоумышленник не будет знать о том, что за ним ведется наблюдение;
- одна система NIDS может использоваться для мониторинга трафика с большим числом потенциальных систем-целей;
- NIDS может осуществлять перехват содержимого всех пакетов, направляющихся на систему-цель.

Среди недостатков данной системы необходимо отметить следующие аспекты:

- NIDS может только выдавать сигнал тревоги, если трафик соответствует предустановленным правилам или признакам;
- NIDS может упустить нужный интересуемый трафик из-за использования широкой полосы пропускания или альтернативных маршрутов;
- NIDS не может определить, была ли атака успешной;
- NIDS не может просматривать зашифрованный трафик;
- в коммутируемых сетях (в отличие от сетей с общими носителями) требуются специальные конфигурации, без которых NIDS будет проверять не весь трафик.

СОВ, которые устанавливаются на хосте и обнаруживают злонамеренные действия на нём, называются *хостовыми* или *системными* СОВ (Host-based intrusion detection system, HIDS). Примерами хостовых СОВ могут быть системы контроля целостности файлов (СКЦФ), которые проверяют системные файлы с целью определения, когда в них были внесены изменения. Мониторы регистрационных файлов (Log-file monitors, LFM) контролируют регистрационные файлы, создаваемые сетевыми сервисами и службами. Цель обманных систем, работающих с псевдосервисами, заключается в воспроизведении хорошо известных уязвимостей для обмана злоумышленников.

Узловые СОВ представляют собой систему датчиков, загружаемых на различные сервера организации и управляемых центральным диспетчером. Датчики отслеживают различные типы событий (более детальное рассмотрение этих событий приводится в следующем разделе) и предпринимают определенные действия на сервере либо передают уведомления. Датчики HIDS отслеживают события, связанные с сервером, на котором они загружены. Сенсор HIDS позволяет определить, была ли атака успешной, если атака имела место на той же платформе, на которой установлен датчик.

Различные типы датчиков HIDS позволяют выполнять различные типы задач по обнаружению вторжений. Не каждый тип дат-

чиков может использоваться в организации, и даже для различных серверов внутри одной организации могут понадобиться разные датчики. Следует заметить, что система HIDS, как правило, стоит дороже, чем сетевая система, так как в этом случае каждый сервер должен иметь лицензию на датчик (датчики дешевле для одного сервера, однако общая стоимость датчиков больше по сравнению со стоимостью использования сетевых СОВ).

С использованием систем HIDS связан еще один вопрос, заключающийся в возможностях процессора на сервере. Процесс анализа датчика на сервере может занимать 5...15 % общего процессорного времени. Если датчик работает на активно используемой системе, его присутствие отрицательно скажется на производительности и, таким образом, придется приобретать более производительную систему.

Анализаторы журналов. Анализатор журнала представляет собой именно то, что отражает само название датчика. Процесс выполняется на сервере и отслеживает соответствующие файлы журналов в системе. Если встречается запись журнала, соответствующая некоторому критерию в процессе датчика HIDS, предпринимается установленное действие.

Большая часть анализаторов журналов настроена на отслеживание записей журналов, которые могут означать событие, связанное с безопасностью системы. Администратор системы, как правило, может определить другие записи журнала, представляющие определенный интерес.

Анализаторы журналов по своей природе являются реактивными системами. Иными словами, они реагируют на событие уже после того, как оно произошло. Таким образом, журнал будет содержать сведения о том, что проникновение в систему выполнено. В большинстве случаев анализаторы журналов не способны предотвратить осуществляемую атаку на систему.

Анализаторы журналов, в частности, хорошо адаптированы для отслеживания активности авторизованных пользователей на внутренних системах. Таким образом, если в организации уделяется внимание контролю за деятельностью системных администраторов или других пользователей системы, можно использовать анализатор журнала для отслеживания активности и перемещения записи об этой активности в область, недосягаемую для администратора или пользователя.

Датчики признаков. Датчики этого типа представляют собой наборы определенных признаков событий безопасности, сопоставля-

емых с входящим трафиком или записями журнала. Различие между датчиками признаков и анализаторами журналов заключается в возможности анализа входящего трафика.

Системы, основанные на сопоставлении признаков, обеспечивают возможность отслеживания атак во время их выполнения в системе, поэтому они могут выдавать дополнительные уведомления о проведении злоумышленных действий. Тем не менее, атака будет успешно или безуспешно завершена перед вступлением в действие датчика HIDS, поэтому датчики этого типа считаются реактивными. Датчик признаков HIDS является полезным при отслеживании авторизованных пользователей внутри информационных систем.

Анализаторы системных вызовов. Анализаторы системных вызовов осуществляют анализ вызовов между приложениями и операционной системой для идентификации событий, связанных с безопасностью. Датчики HIDS данного типа размещают программную спайку между операционной системой и приложениями. Когда приложению требуется выполнить действие, его вызов операционной системы анализируется и сопоставляется с базой данных признаков. Эти признаки являются примерами различных типов поведения, которые являют собой атакующие действия, или объектом интереса для администратора СОВ.

Анализаторы системных вызовов отличаются от анализаторов журналов и датчиков признаков HIDS тем, что они могут предотвращать действия. Если приложение генерирует вызов, соответствующий, например, признаку атаки на переполнение буфера, датчик позволяет предотвратить этот вызов и сохранить систему в безопасности.

Анализаторы поведения приложений. Анализаторы поведения приложений аналогичны анализаторам системных вызовов в том, что они применяются в виде программной спайки между приложениями и операционной системой. В анализаторах поведения датчик проверяет вызов на предмет того, разрешено ли приложению выполнять данное действие, вместо определения соответствие вызова признакам атак. Например, веб-серверу обычно разрешается принимать сетевые соединения через порт 80, считывать файлы в веб-каталоге и передавать эти файлы по соединениям через порт 80. Если веб-сервер попытается записать или считать файлы из другого места или открыть новые сетевые соединения, датчик обнаружит несоответствующее норме поведение сервера и заблокирует действие.

При конфигурировании таких датчиков необходимо создавать список действий, разрешенных для выполнения каждым приложе-

нием. Поставщики датчиков данного типа предоставляют шаблоны для наиболее широко используемых приложений. Любые «доморощенные» приложения должны анализироваться на предмет того, какие действия им разрешается выполнять, и выполнение этой задачи должно быть программно реализовано в датчике.

Контролеры целостности файлов. Контролеры целостности файлов отслеживают изменения в файлах. Это осуществляется посредством использования криптографической контрольной суммы или цифровой подписи файла. Конечная цифровая подпись файла будет изменена, если произойдет изменение хотя бы малой части исходного файла (это могут быть атрибуты файла, такие как время и дата создания). Алгоритмы, используемые для выполнения этого процесса, разрабатывались с целью максимального снижения возможности для внесения изменений в файл с сохранением прежней подписи.

При изначальной конфигурации датчика каждый файл, подлежащий мониторингу, подвергается обработке алгоритмом для создания начальной подписи. Полученное число сохраняется в безопасном месте. Периодически для каждого файла эта подпись пересчитывается и сопоставляется с оригиналом. Если подписи совпадают, это означает, что файл не был изменен. Если соответствия нет, значит, в файл были внесены изменения.

По *методам анализа* СОВ делят на две группы: СОВ, которые сравнивают информацию с предустановленной базой сигнатур атак и СОВ, контролирующие частоту событий или обнаружение статистических аномалий.

Анализ сигнатур был первым методом, примененным для обнаружения вторжений. Он базируется на простом понятии совпадения последовательности с образцом. Во входящем пакете просматривается байт за байтом и сравнивается с сигнатурой (подписью) — характерной строкой программы, указывающей на характеристику вредного трафика. Такая подпись может содержать ключевую фразу или команду, которая связана с нападением. Если совпадение найдено, объявляется тревога.

Второй метод анализа состоит в рассмотрении строго форматированных данных трафика сети, известных как протоколы. Каждый пакет сопровождается различными протоколами. Авторы СОВ, зная это, внедрили инструменты, которые разворачивают и осматривают эти протоколы, согласно стандартам. Каждый протокол имеет несколько полей с ожидаемыми или нормальными значениями. Если что-нибудь нарушает эти стандарты, то вероятна зло-

намеренность. СОВ просматривает каждое поле всех протоколов входящих пакетов: IP, TCP, и UDP. Если имеются нарушения протокола, например если он содержит неожиданное значение в одном из полей, объявляется тревога.

Системы анализа сигнатуры имеют несколько важных сильных сторон. Во-первых, они очень быстры, так как полный анализ пакета — относительно тяжелая задача. Правила легко написать, понять и настроить. Кроме того, имеется просто фантастическая поддержка компьютерного сообщества в быстром производстве сигнатур для новых опасностей. Эти системы превосходят все другие при отлове хакеров на первичном этапе: простые атаки имеют привычку использовать некие предварительные действия, которые легко распознать. Наконец, анализ, основанный на сигнатуре, точно и быстро сообщает, что в системе все нормально (если это действительно так), поскольку должны произойти некие особые события для объявления тревоги.

С другой стороны СОВ, основанная только на анализе сигнатур, имеет определенные слабости. Являясь первоначально очень быстрой, со временем скорость ее работы будет замедляться, поскольку возрастает число проверяемых сигнатур. Это существенная проблема, поскольку число проверяемых сигнатур может расти очень быстро. Фактически, каждая новая атака или действие, придуманное атакующим, увеличивает список проверяемых сигнатур. Не помогут даже эффективные методы работы с данными и пакетами: огромное количество слегка измененных атак могут проскользнуть через такую систему.

Имеется и другая сторона проблемы: так как система работает, сравнивая список имеющихся сигнатур с данными пакета, такая СОВ может выявить только уже известные атаки, сигнатуры которых имеются.

Но необходимо отметить, что согласно статистике 80 % атак происходит по давно известным сценариям. Наличие в системе обнаружения сигнатур известных атак даёт высокий процент обнаружения вторжений.

В случае анализа протоколов тоже имеются свои достоинства и недостатки. Из-за предпроцессов, требующих тщательной экспертизы протоколов, анализ протокола может быть довольно медленным. Кроме того, правила проверки для системы протокола трудно написать и понять. Можно даже сказать, что в этом случае приходится уповать на добросовестность производителя программы, так

как правила относительно сложны и трудны для самостоятельной настройки.

На первый взгляд, СОВ на основе анализа протокола работают медленнее, чем системы на основе сигнатуры, они более «основательны» в смысле масштабности и результатов. Кроме того, эти системы ищут «генетические нарушения» и часто могут отлавливать свежайшие «эксплоиты нулевого дня».

СОВ можно разбить на следующие категории:

системы обнаружения атак на сетевом уровне (Network IDS, NIDS) контролирует пакеты в сетевом окружении и обнаруживают попытки злоумышленника проникнуть внутрь защищаемой системы (или реализовать атаку типа «отказ в обслуживании»);

системы контроля целостности (System integrity verifiers, SIV) проверяют системные файлы для того, чтобы определить, когда злоумышленник внес в них изменения;

мониторы регистрационных файлов (Log-file monitors, LFM) контролируют регистрационные файлы, создаваемые сетевыми сервисами и службами.

2.2. Архитектура СОВ

У систем обнаружения вторжений целесообразно различать локальную и глобальную архитектуру. В рамках локальной архитектуры реализуются элементарные составляющие, которые затем могут быть объединены для обслуживания корпоративных систем [1–3].

Основные элементы *локальной архитектуры* и связи между ними показаны на рис. 2.4. Первичный сбор данных осуществляют агенты, называемые также сенсорами. Регистрационная информация может извлекаться из системных или прикладных журналов (технически несложно получать ее и напрямую от ядра ОС) либо добываться из сети с помощью соответствующих механизмов активного сетевого оборудования или путем перехвата пакетов посредством установленной в режим мониторинга сетевой карты.

На уровне агентов (сенсоров) может выполняться фильтрация данных с целью уменьшения их объема. Это требует от агентов некоторого интеллекта, но зато разгружает остальные компоненты системы.

Агенты передают информацию в центр распределения, который приводит ее к единому формату, возможно осуществляет дальнейшую фильтрацию, сохраняет в базе данных и направляет для анализа статистическому и экспертному компонентам. Один центр распределения может обслуживать несколько сенсоров.

Рис. 2.4. Основные элементы локальной архитектуры систем обнаружения вторжений

Содержательный активный аудит начинается со статистического и экспертного компонентов. Если в процессе статистического или экспертного анализа выявляется подозрительная активность, соответствующее сообщение направляется решателю, который определяет, является ли тревога оправданной, и выбирает способ реагирования.

Хорошая система обнаружения вторжений должна уметь внятно объяснить, почему она подняла тревогу, насколько серьезна ситуация и каковы рекомендуемые способы действия. Если выбор должен оставаться за человеком, то пусть он сводится к нескольким элементам меню, а не к решению концептуальных проблем.

Глобальная архитектура подразумевает организацию одноранговых и разноранговых связей между локальными системами обнаружения вторжений (рис. 2.5).

На одном уровне иерархии располагаются компоненты, анализирующие подозрительную активность с разных точек зрения. Например, на хосте могут располагаться подсистемы анализа поведения пользователей и приложений. Их может дополнять подсистема анализа сетевой активности. Когда один компонент обнаруживает что-то подозрительное, то во многих случаях целесообразно сообщить об этом соседям либо для принятия мер, либо для усиления внимания к определенным аспектам поведения системы.

Разноранговые связи используются для обобщения результатов анализа и получения целостной картины происходящего. Иногда у локального компонента недостаточно оснований для возбуждения

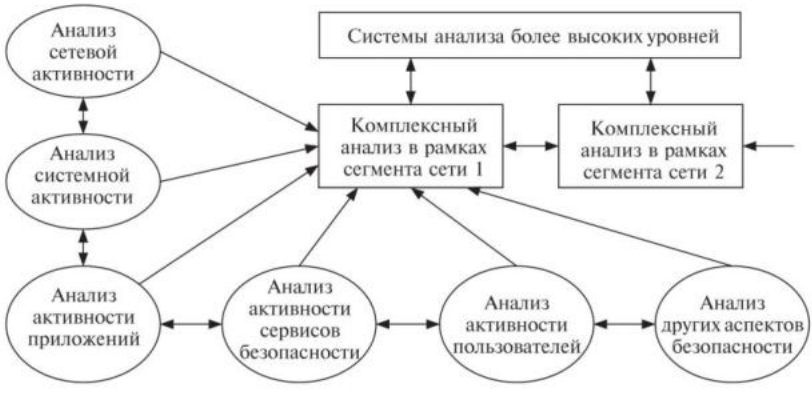

Рис. 2.5. Глобальная архитектура систем обнаружения вторжений

тревоги, но «по совокупности» подозрительные ситуации могут быть объединены и совместно проанализированы, после чего порог подозрительности окажется превышенным. Целостная картина, возможно, позволит выявить скоординированные атаки на разные участки информационной системы и оценить ущерб в масштабе организации.

2.3. Структура системы обнаружения вторжения

В современных системах обнаружения логически выделяют следующие основные элементы (рис. 2.6):

- подсистема сбора информации используется для сбора первичной информации о работе защищаемой системы;
- подсистема анализа (обнаружения) осуществляет поиск атак и вторжений в защищаемую систему;
- подсистема представления данных (пользовательский интерфейс) позволяет пользователю(ям) СОВ следить за состоянием защищаемой системы.

Подсистема сбора информации аккумулирует данные о работе защищаемой системы. Для сбора информации используются автономные модули — датчики. Количество используемых датчиков различно и зависит от специфики защищаемой системы. Датчики в СОВ принято классифицировать по характеру собираемой информации. В соответствии с общей структурой информационных систем выделяют следующие типы:

- датчики приложений — данные о работе программного обеспечения защищаемой системы;
- датчики хоста — функционирование рабочей станции защищаемой системы;
- датчики сети — сбор данных для оценки сетевого трафика;

Рис. 2.6. Структура системы обнаружения вторжения

- межсетевые датчики — содержат характеристики данных, циркулирующих между сетями.

Сетевые датчики бывают двух видов.

Сетевой датчик 1-го типа (рис. 2.7) имеет следующие функции и возможности:

- мониторинг трафика в заданном сегменте: фиксация SPAN, TAP, VACL и т. д.;
- сопоставление трафика с сигнатурами атак;
- поиск эвристических шаблонов атаки, аномалий протоколов;
- определение характера атак на основе встроенных логических алгоритмов фрагментации и повторной сборки потока;

Сетевой датчик 1-го типа является иинструментом выдачи сигналов тревоги и наглядного представления, кроме того, предоставляется возможность для определенных активных действий: разрыв TCP, блокирование, регистрация сеанса IP.

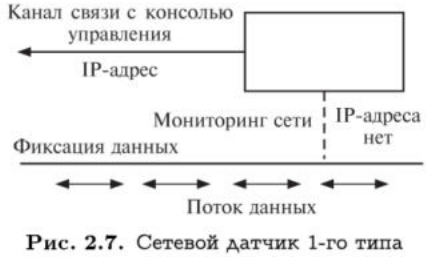

Рис. 2.7. Сетевой датчик 1-го типа

Рис. 2.8. Сетевой датчик 2-го типа

Сетевой датчик 2-го типа (рис. 2.8) обычно имеет следующие функции и возможности:

- мониторинг всего трафика, «прозрачно» проходящего по двум интерфейсам;
- сопоставление трафика с хорошо известными сигнатурами атак, а также поиск эвристических шаблонов атак и аномалий протоколов;
- реализация логического алгоритма фрагментации для четкого определения характера атак, а также нормализации потока пакетов TCP/IP.

Сетевой датчик 2-го типа служит одновременно инструментом выдачи сигналов тревоги и наглядного представления, выполняет задачи профилактики путем фильтрации пакетов. Кроме того, обеспечивает возможность активных ответных действий: разрыв TCP, блокировка, регистрация сеанса IP и отклонение пакета/потока/пользователя. Такие датчики называют *IPS датчиками.*

Подсистема анализа структурно состоит из одного или более модулей анализа — анализаторов. Наличие нескольких анализаторов требуется для повышения эффективности обнаружения. Каждый анализатор выполняет поиск атак или вторжений определенного типа. Входными данными для анализатора является информация из подсистемы сбора информации или от другого анализатора. Результат работы подсистемы — индикация о состоянии защищаемой системы. В случае, когда анализатор сообщает об обнаружении несанкционированных действий, на его выходе может появляться некоторая дополнительная информация. Обычно эта информация содержит выводы, подтверждающие факт наличия вторжения или атаки.

Подсистема представления данных необходима для информирования заинтересованных лиц о состоянии защищаемой системы. В некоторых системах предполагается наличие групп пользователей, каждая из которых контролирует определенные подсистемы защищаемой системы. Поэтому в таких СОВ применяется разграничение доступа, групповые политики, полномочия и т. д.

Взаимодействие сетевых (Network-based) и хостовых (Host based) СОВ показано на рис. 2.9.

Первый шаг в установке выбранной СОВ — логично вписать систему в корпоративную политику информационной безопасности. Прежде всего, нужно определить, каким образом приложение будет работать со всей архитектурой системы безопасности, выделить

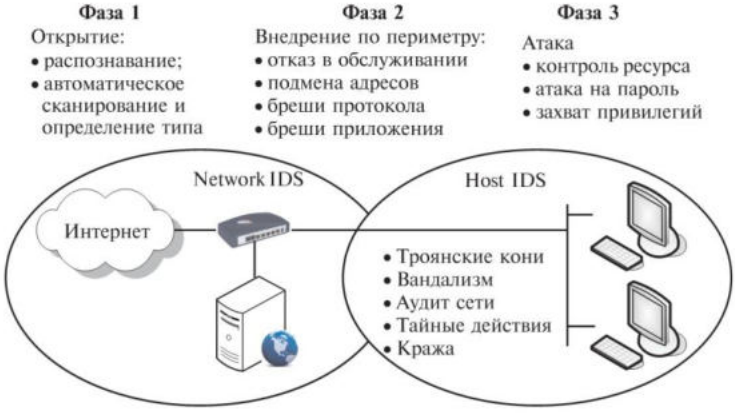

Рис. 2.9. Взаимодействие Network и Host IDS

процессы, которые будут под надзором, и назначить ресурсы (ПО, «железо» и люди, ответственные за технологию).

Непосредственно перед составлением плана внедрения выбранной СОВ необходимо как можно точнее и полнее расписать всю сетевую структуру с указанием характеристик каждого сегмента сети. Проанализировать пограничные сегменты сети (маршрутизаторы, переключатели и межсетевые экраны). Составить список разрешенных пользователей сети (внутренних и внешних). Перечислить все станции с указанием процессов, обрабатываемых станциями, и списком допущенных пользователей.

После установки выбранной системы администратор должен грамотно настроить приложение, особенно учитывая объем информации, накапливаемый системой для последующего анализа. Также файлы записей истории должны быть защищены от возможности умышленного уничтожения следов проникновения в систему. Как и антивирусная база данных, база шаблонов атак должна периодически актуализироваться производителем.

По способу анализа данных СОВ делятся на две группы — signature-based (RBID) и anomaly-based (SBID), т. е. сигнатурные и поведенческие.

Достоинства RBID систем: довольно четкое определение типа атаки, высокая точность работы практически без ложных срабатываний.

Недостатки RBID систем:

- неустойчивость к новейшим типам атак, поскольку на момент атаки базы знаний (сигнатур) еще не содержат соответствующих сценариев;

- зависимость эффективности работы от скорости разработки новых сигнатур атак;
- потери времени от разработки сигнатуры разработчиками СОВ до обновления базы данных сигнатур организацией потребителя СОВ;
- для сложных распределенных атак проверка на соответствие сигнатуре является нетривиальной задачей;
- большинство баз знаний сигнатур и правил общедоступны, поэтому нарушитель может использовать методы «маскировки» атаки.

Достоинства SBID систем:

- могут обнаруживать совершенно новые виды атак;
- способны обнаружить атаки, характеризующиеся большой продолжительностью во времени;
- такие системы в некотором смысле проще обслуживать, так как нет нужды в обновлении сигнатур.

Недостатки SBID систем:

- сложно построить модель «нормальной» работы сети, поэтому SBID склонны к ложному срабатыванию сигналов об атаках;
- такие системы необходимо «обучать» некоторый период времени, и они не могут работать сразу же после инсталляции в ИТС;
- введением такой системы в эксплуатацию должны заниматься высококвалифицированные в данном направлении специалисты;
- в отличие от RBID систем, SBID не генерируют сообщения, точно описывающие атаку; при атаке будет сгенерировано только сообщение об «аномальности», возможно с некоторой дополнительной информацией и статистическими характеристиками;
- необходимость установки эффективного порогового значения для сигнализации атаки; это окажет влияние либо на увеличение частоты ложных срабатываний, либо система не будет выдавать сигналы там, где необходимо.

3 ТЕХНОЛОГИИ ПОСТРОЕНИЯ СИСТЕМ ОБНАРУЖЕНИЯ АТАК

Системы обнаружения атак (СОА), как и большинство современных программных продуктов, должны удовлетворять ряду требований. Это и современные технологии разработки, и ориентировка на особенности современных информационных сетей, и совместимость с другими программами. Чтобы понять, как правильно использовать СОА, нужно четко представлять, как они работают и каковы их уязвимые места.

Рассмотрим принципы, на которых основана идея обнаружения компьютерных атак. Если не учитывать различные несущественные инновации в области обнаружения компьютерных атак, то можно смело утверждать, что существуют две основные технологии построения СОА. Суть их заключается в том, что СОА обладают некоторым набором знаний либо о методах вторжений, либо о нормальном поведении наблюдаемого объекта.

Системы обнаружения аномального поведения основаны на том, что СОА известны некоторые признаки, характеризующие правильное или допустимое поведение объекта наблюдения. Под нормальным или правильным поведением понимаются действия, выполняемые объектом и не противоречащие политике безопасности.

Системы обнаружения злоумышленного поведения (misuse detection) основаны на том, что СОА известны некоторые признаки, характеризующие поведение злоумышленника. Наиболее распро-

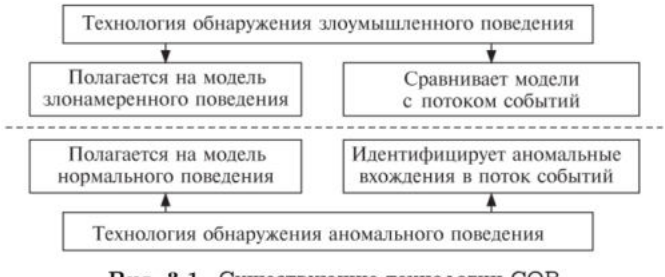

Рис. 3.1. Существующие технологии СОВ

страненной реализацией технологии обнаружения злоумышленного поведения являются экспертные системы (например, системы Snort, RealSecure IDS, Enterasys Advanced Dragon IDS).

Краткая схема на рис. 3.1 обобщает эти сведения. Все остальные подходы являются подмножествами этих технологий.

3.1. Существующие технологии СОВ

Все системы обнаружения вторжений можно разделить на системы, ориентированные на поиск:

- аномалий взаимодействия контролируемых объектов;
- сигнатур всех узнаваемых атак;
- искажения эталонной профильной информации.

3.1.1. Технологии обнаружения аномальной активности

Датчики-сенсоры аномалий идентифицируют необычное поведение, аномалии в функционировании отдельного объекта — трудности их применения на практике связаны с нестабильностью самих защищаемых объектов и взаимодействующих с ними внешних объектов. В качестве объекта наблюдения может выступать сеть в целом, отдельный компьютер, сетевая служба (например, FTP-сервер), пользователь и т. д. Датчики срабатывают при условии, что нападения отличаются от «обычной» (законной) деятельности. Здесь появляется еще одно слабое место, характерное в большей степени для конкретных реализаций, заключающееся в некорректности определения «дистанции» отклонения наблюдаемого поведения от штатного, принятого в системе, и определении «порога срабатывания» сенсора наблюдения.

Меры и методы, обычно используемые в обнаружении аномалии, включают в себя следующие атрибуты:

пороговые значения. Наблюдения за объектом выражаются в виде числовых интервалов. Выход за пределы этих интервалов считается аномальным поведением. В качестве наблюдаемых параметров могут быть, например, количество файлов, к которым обращается пользователь в данный период времени, число неудачных попыток входа в систему, загрузка центрального процессора и т. п. Пороги могут быть статическими и динамическими (т. е. изменяться, подстраиваясь под конкретную систему);

статистические меры. Решение о наличии атаки делается по большому количеству собранных данных путем их статистической предобработки;

параметрические меры. Для выявления атак строится специальный «профиль нормальной системы» на основе шаблонов (т. е.

некоторой политики, которой обычно должен придерживаться данный объект);

непараметрические меры. Здесь уже профиль строится на основе наблюдения за объектом в период обучения;

меры на основе правил (сигнатур). Они очень похожи на непараметрические статистические меры. В период обучения составляется представление о нормальном поведении объекта, которое записывается в виде специальных «правил». Получаются сигнатуры «хорошего» поведения объекта;

другие меры. Нейронные сети, генетические алгоритмы, позволяющие классифицировать некоторый набор видимых сенсорудатчику признаков.

Следует заметить, что существуют две крайности при использовании данной технологии:

- обнаружение аномального поведения, которое не является атакой, и отнесение его к классу атак (ошибка второго рода);
- пропуск атаки, которая не подпадает под определение аномального поведения (ошибка первого рода). Этот случай гораздо более опасен, чем ложное причисление аномального поведения к классу атак.

Поэтому при инсталляции и эксплуатации систем такой категории обычные пользователи и специалисты сталкиваются с двумя довольно нетривиальными задачами:

- *построение профиля объекта* — это трудно формализуемая и затратная по времени задача, требующая от специалиста по безопасности большой предварительной работы, высокой квалификации и опыта;
- *определение граничных значений* характеристик поведения субъекта для снижения вероятности появления одного из двух вышеназванных крайних случаев.

Обычно системы обнаружения аномальной активности используют журналы регистрации и текущую деятельность пользователя в качестве источника данных для анализа. Достоинства систем обнаружения атак на основе технологии выявления аномального поведения можно оценить следующим образом:

- системы обнаружения аномалий способны обнаруживать новые типы атак, сигнатуры для которых еще не разработаны;
- обнаружения аномалий генерируют информацию, которая может быть использована в системах обнаружения злоумышленного поведения;

- они не нуждаются в обновлении сигнатур и правил обнаружения атак.

Недостатками систем на основе технологии обнаружения аномального поведения являются следующие:

- системы требуют длительного и качественного обучения;
- системы генерируют много ошибок второго рода;
- системы обычно слишком медленны в работе и требуют большого количества вычислительных ресурсов.

3.1.2. Анализ систем, использующих сигнатурные методы

Сигнатурные методы позволяют описать атаку набором правил или с помощью формальной модели, в качестве которой может применяться символьная строка, семантическое выражение на специальном языке и т. п. Суть данного метода заключается в использовании специализированной базы данных шаблонов (сигнатур) атак для поиска действий, подпадающих под определение «атака».

Сигнатурный метод может защитить от вирусной или хакерской атаки, когда уже известна сигнатура атаки (например, неизменный фрагмент тела вируса) и она внесена в базу данных СОА. То есть, когда сеть переживает первое нападение извне, первое заражение происходит еще неизвестным вирусом, и в базе попросту отсутствует сигнатура для его поиска, сигнатурная СОА не сможет сигнализировать об опасности, поскольку сочтет атакующую деятельность легитимной.

Кроме того, несмотря на кажущуюся простоту сигнатурного метода, и в его реализации есть свои тонкости. Классический пример — с помощью поиска сигнатуры

/../../../../../../../local.ida

и простого сравнения битовой информации невозможно выявить хакерскую атаку на HTTP-сервер. Нападающий может легко изменить строку в соответствии с соглашением об URI и использовать битовую строку

%2F%2E%2F%2E%2F%2E%2F%2E%2F%2E%2F%2Elocal.ida

которую данная сигнатура уже не охватывает.

Большинство существующих программных продуктов, заявляющих об использовании сигнатурного метода, на самом деле реализуют как раз наиболее примитивный способ сигнатурного распознавания. К ним относятся и западные, и практически все российские

разработки. Многие системы позиционируются как предназначенные для выявления атак в информационных системах на основе интеллектуального анализа сетевых пакетов. На самом же деле сигнатурный метод реализован как алгоритм, исследующий лишь динамику развития атаки, основанный на автомате состояний для оценки сценария развития атаки. По замыслу такой подход должен позволить отследить динамику развития атаки в соответствии с действиями злоумышленника, при этом в качестве модуля сбора данных могут использоваться даже сами системы обнаружения атак.

Однако на практике, например, использование в качестве сбора данных системы Snort сильно замедляет процесс обнаружения, что не позволяет осуществлять анализ в режиме реального времени (хотя оригинальная СОА Snort работает в режиме, близком к реальному времени). С другой стороны, такая система становится очень сложной из-за использования большого количества конфигурационных параметров и переусложнения схемы обработки данных.

Таким образом, эффективность работы сигнатурной СОА определяется тремя основными факторами: оперативностью пополнения сигнатурной базы, ее полнотой с точки зрения определения сигнатур атак, а также наличием интеллектуальных алгоритмов сведения действий атакующих к некоторым базовым шагам, в рамках которых происходит сравнение с сигнатурами.

Для успешной реализации первых двух факторов необходима поддержка международных стандартов и рекомендаций (например, Intrusion Detection Message Exchange Requirements) обмена сигнатурами и информацией об атаках. Поскольку на данный момент не существует достаточно большого количества распределенных и объективных источников сигнатур, то СОА данного типа имеют весьма лимитированную эффективность в реальных сетях.

Рассмотрим различные методы, использующие сигнатуры вторжений:

Продукционные (экспертные) системы обнаружения вторжения. Продукционные СОВ кодируют данные об атаках и правила импликации «если — то», а также подтверждают их, обращаясь к контрольным записям событий. Правила кодируются для определения условий, необходимых для атаки в части «если». Когда эти условия соблюдены, выполняется часть «то» правил.

Продукционные СОВ предназначены, во-первых, для логического вывода факта вторжения на основании данных аудита; проблемами, усложняющими решение данной задачи, являются:

- трудность учета последовательности событий, что заставляет ввести дополнительные проверки данных, определяющие их последовательность;
- необходимость хорошего администратора системы, который мог бы настроить ее в соответствии со своими знаниями о безопасности системы, что может сделать базу знаний плохо переносимой;
- могут быть обнаружены только известные уязвимости,

во-вторых, для объединения возможных атак для получения общей картины вторжения.

Преимуществом продукционных СОВ является отделение управляющего решения от его формулировки. Кроме описанных выше недостатками продукционных СОВ являются управление огромным количеством данных, соответствующих вторжению, а также высокая квалификация необходимая для поддержки базы знаний СОВ.

Обнаружение вторжений, основанное на модели. Этот метод построения СОВ является одним из вариантов, объединяющим модели вторжения и доказательств, поддерживающих вывод о вторжении. В СОВ поддерживается база данных сценариев атак. Каждый из сценариев составляет последовательность поведений, описывающих атаку. СОВ пытается выявлять эти сценарии по записям в контрольном журнале и таким образом доказывать или опровергать их присутствие. Такой процесс называют планировщиком. Планировщик пытается определить последовательность поведений, основанную на текущих активных моделях вторжения, которые могут быть проверены в журнале по шаблонам. Метод основан на накоплении контрольной информации, и поэтому возврат к предыдущему результату проверки сценария не вызывает осложнений. Планировщик переводит описание поведения в активность в вычислительной системе и сравнивает его с данными системнозависимого контрольного журнала.

Преимуществами данного метода являются:

- математическая база поддержки вывода в условиях неопределенности;
- планировщик обеспечивает независимое представление данных аудита.

Недостатками данного метода являются:

- дополнительная нагрузка на эксперта выражается в правильном присвоении имеющих смысл числовых значений элементам графа, представляющего модель;

- поведение описывается последовательным множеством событий, что затрудняет описание сложных атак.

Необходимо отметить, что обнаружение вторжений, основанное на сравнении с моделью, не заменяет, а дополняет метод определения по статистическим критериям.

Анализ перехода системы из состояния в состояние. Данный подход представлен в системах STAT и USTAT под UNIX. В данных СОН атаки представляются как последовательность переходов контролируемой системы из состояния в состояние. Состояние шаблона атаки соответствует состоянию системы и связано с утверждениями, которые должны быть удовлетворены для последующего перехода из состояния в состояние. Возможные состояния связаны дугами, представляющими события необходимые для перехода из состояния в состояние. Типы допустимых событий встроены в модель. Данная модель может определить только атаку, состоящую из последовательных событий, не позволяя выразить атаки с более сложной их структурой. Более того, не существует общего целевого механизма в случае частичного распознавания атак.

Контроль нажатия клавиш. Данный метод использует факт нажатия клавиши пользователем для определения атаки. Метод состоит в сравнении набора на клавиатуре с эталонной последовательностью, определяющей атаку. Их совпадение служит признаком атаки. Недостатки этого подхода — отсутствие надежного получения ввода пользователя без соответствующей поддержки операционной системы и наличие множества способов, включая и непроизвольные, имитации той же атаки. Более того, без семантического анализа введенная в командной строке последовательность может элементарно обойти СОН, использующие данный метод. Так как метод отмечает только нажатие клавиш, то автоматизированные атаки, которые, являясь результатом выполнения некоторого РПС, не могут быть обнаружены.

3.1.3. Концепция обнаружения компьютерных угроз

При построении современной системы обнаружения вторжений необходимо, прежде всего, сформировать правильные взгляды на информационные процессы, проходящие не только в компьютерной сети, но и во всей информационной системе (ИС). Система обнаружения компьютерных вторжений и атак, по сути, является специализированной системой обработки информации, предназначенной для чрезвычайно быстрого анализа огромного объема данных совершенно разного вида. Для того чтобы определить наиболее точно критерии эффективности такой системы и оценить параметры,

которые наиболее сильно влияют на скорость и точность работы, необходимо проанализировать, какого рода данные будут обрабатываться в системе и каким образом это должно происходить.

При этом следует учитывать тот факт, что система обнаружения атак должна функционировать адекватно угрозам безопасности, характерным для рассматриваемых объектов информационной системы, поэтому исходной позицией является выявление перечня угроз, характерных для данной ИС. К сожалению, практически все существующие системы обнаружения компьютерных атак лишены функциональности, позволяющей связывать риски и угрозы безопасности с происходящими в сетевой и локальной вычислительной среде событиями. В результате такого одностороннего анализа, когда в расчет принимаются только технические параметры сети, причем их весьма ограниченное количество, страдает в первую очередь качество обнаружения атак. Более того, пользователь такой системы никогда не получит той информации, ради которой эти системы эксплуатируются — информации о реализации угроз безопасности, которым подвержена защищаемая сетевая и локальная инфраструктура.

Обнаружение угроз безопасности. Для описания нового подхода введем понятия, которые будут применяться в дальнейшем. Под информационной системой в данной работе будет пониматься совокупность технических средств (компьютеров, коммуникационного оборудования, линий передачи данных), при помощи которых обеспечивается обработка информации в организации.

Под угрозой информационной системе будем понимать потенциально возможное действие, предпринимаемое злоумышленником, которое может привести к прямому или косвенному ущербу. В этом предложении рассматриваются действия, направленные на нарушение установленных владельцем правил функционирования системы, выполняемые при помощи различных средств вычислительной техники.

Целью приведенной ниже концепции обнаружения угроз информационной безопасности является определение новых требований и принципов конструирования систем обнаружения компьютерных атак, ориентированных на комплексную обработку информации о защищаемой инфраструктуре для своевременного выявления и предупреждения о возможности реализации угроз, присущих информационной системе.

На сегодня пирамида информационной обработки данных в современной СОА выглядит следующим образом (рис. 3.2).

Рис. 3.2. Информационная пирамида

Верхняя часть информационной пирамиды — это риски и угрозы, присущие рассматриваемой системе. Ниже располагаются различные варианты реализаций угроз (атаки), и самый нижний уровень — это признаки атак. Конечный пользователь, равно как и система обнаружения атак, имеет возможность регистрировать только процесс развития конкретной атаки или свершившийся факт атаки по наблюдаемым характерным признакам. Признаки атаки — то, что мы реально можем зафиксировать и обработать различными техническими средствами, а следовательно, необходимы средства фиксации признаков атак.

Если данный процесс рассматривать во времени, то можно говорить, что определенные последовательности наблюдаемых признаков порождают события безопасности. События безопасности могут переводить защищаемые объекты информационной системы в небезопасное состояние. Следовательно, для системы обнаружения атак необходим информационный срез достаточной полноты, содержащий все события безопасности, произошедшие в информационной системе за рассматриваемый период. Кроме того, поднимаясь вверх по пирамиде, для события безопасности можно указать, к реализации какого вида угроз оно может привести, для того чтобы в процессе развития атаки производить прогнозирование ее развития и принимать меры по противодействию угрозам, которые может вызывать данная атака.

Методология обработки данных в современных информационных системах подразумевает повсеместное использование многоуровневости. Для СОА нового типа можно выделить следующие крупные уровни, на которых возможно осуществление доступа к обрабатываемой информации.

Рис. 3.3. Уровни обработки информации в информационной системе

Уровень прикладного ПО, с которым работает конечный пользователь информационной системы. Прикладное программное обеспечение зачастую имеет уязвимости, которые могут использовать злоумышленники для доступа к обрабатываемым данным ПО.

Уровень СУБД. Является частным случаем средств прикладного уровня, но должен выделяться в отдельный класс в силу своей специфики. СУБД, как правило, имеет свою собственную систему политик безопасности и организации доступа пользователей, которую нельзя не учитывать при организации защиты.

Уровень ОС. Операционная система компьютеров защищаемой ИС является важным звеном защиты, поскольку любое прикладное ПО использует средства, предоставляемые именно ОС. Бесполезно совершенствовать качество и надежность прикладного ПО, если оно эксплуатируется на незащищенной ОС.

Уровень среды передачи. Современные ИС подразумевают использование различных сред передачи данных для взаимосвязи аппаратных компонентов, входящих в состав ИС. Среды передачи данных являются на сегодня одними из самых незащищенных компонентов ИС. Контроль среды передачи и передаваемых данных является одной из обязательных составляющих механизмов защиты данных.

Иллюстративно уровни обработки потоков данных в информационной системе представлены на рис. 3.3.

Исходя из вышесказанного, можно сделать вывод, что любые средства защиты информации, в том числе и системы обнаружения и предупреждения атак, обязаны иметь возможность анализировать обрабатываемые и передаваемые данные на каждом из выделенных уровней. Требование присутствия системы обнаружения атак на каждом функциональном уровне информационной систе-

мы приводит к необходимости выделения подсистемы регистрации событий безопасности в отдельный комплекс информационных зондов СОА, обеспечивающих сбор информации в рамках всей сети информационной системы. В то же время разнородность программно-аппаратных платформ и задач, решаемых различными объектами ИС, требует применения модульной архитектуры информационных зондов для обеспечения возможности максимальной адаптации к конкретным условиям применения.

3.2. Повышение эффективности систем обнаружения атак — интегральный подход

Вообще говоря, современные системы обнаружения вторжений и атак еще далеки от эргономичных и эффективных с точки зрения безопасности решений. Повышение же эффективности следует ввести не только в области обнаружения злонамеренных воздействий на инфраструктуру защищаемых объектов информатизации, но и с точки зрения повседневной «боевой» эксплуатации данных средств, а также экономии вычислительных и информационных ресурсов владельца данной системы защиты [4, 5, 7, 8].

Если же говорить непосредственно о модулях обработки данных, то, следуя логике предыдущего раздела, каждая сигнатура атаки в представленной схеме обработки информации об атаке является базовым элементом для распознавания более общих действий — распознавания фазы атаки (этапа ее реализации). Само понятие сигнатуры обобщается до некоторого решающего правила (например, с помощью поиска аномалий в сетевом трафике или клавиатурном почерке пользователя). А каждая атака наоборот разбивается на набор этапов ее проведения. Чем проще атака, тем проще ее обнаружить и больше возможностей появляется по ее анализу. Каждая сигнатура отображает определенное событие в вычислительной сетевой и локальной среде в фазовое пространство компьютерных атак. Фазы можно определить свободно, но лучше сохранять при этом достаточную степень детализации, чтобы иметь возможность описывать атаки с помощью подробных сценариев атак (списка фаз атак и переходов между ними).

Сценарий атаки в этом случае представляет собой граф переходов, аналогичный графу конечного детерминированного автомата. А фазы атак можно описать, например, следующим образом:

- опробование портов;
- идентификация программных и аппаратных средств;
- сбор баннеров;

- применение эксплойтов;
- дезорганизация функционала сети с помощью атак на отказ в обслуживании;
- управление через бэкдоры;
- поиск установленных троянов;
- поиск прокси-серверов;
- удаление следов присутствия и т. д. (по необходимости с различной степенью детализации).

Преимущества такого подхода очевидны — в случае раздельной обработки различных этапов атаки появляется возможность распознавать угрозу еще в процессе ее подготовки и формирования, а не на стадии ее реализации, как это происходит в существующих системах. При этом элементной базой для распознавания может быть как сигнатурный поиск, так и выявление аномалий, использование экспертных методов и систем, доверительных отношений и прочих информационных, уже известных и реализованных, сетевых и локальных примитивов оценки происходящего в вычислительной среде потока событий. Обобщающий подход к анализу позволяет соответственно определять и распределенные (во всех смыслах) угрозы, как во временном, так и логическом и физическом пространстве. Общая схема обработки поступающих событий также позволяет осуществлять поиск распределенных атак — путем последующей агрегации данных из различных источников и конструирования метаданных об известных инцидентах по защищаемому «периметру» (рис. 3.4).

Разбиение атаки на более мелкие (фазы) позволяет:

- осуществлять более точное распознавание атак (чем проще атака, тем проще ее обнаружить);
- понять, каким должно быть реагирование на атаку;
- использовать классификатор угроз и прогнозировать поведение атакующего;
- комбинировать поиск атак на уровне хоста и сетевом уровне.

Управление трояном. Идея достаточно проста — для управления трояном используется скрипт, написанный на PHP, ASP, Perl или чём-нибудь подобном. Скрипт может размещаться на любом хостинге с поддержкой соответствующих сценариев.

Обмен информацией с трояном происходит следующим образом: когда хакер хочет передать трояну команду, он по HTTP-протоколу посылает её скрипту. Скрипт принимает команду и где-то её сохраняет (например, в текстовом файле у себя на сервере). Как только троян обратится к скрипту и «спросит», не было ли чего-нибудь от хакера, скрипт отдаст ему сохранённую команду. Естественно,

Рис. 3.4. Схема интегрального обнаружения компьютерных атак

для того что бы вовремя узнавать о поступивших командах, трояну нужно регулярно обращаться к скрипту.

Распределенные атаки выявляются путем агрегации данных о поступающих атаках и подозрительных действиях и сопоставления шаблонов и статистической фильтрации. Таким образом, оповещение о подозрительных действиях в компьютерных системах происходит на нескольких уровнях:

- нижний уровень сообщает о примитивных событиях (совпадении сигнатур, выявлении аномалий);
- средний уровень извлекает информацию из нижнего уровня и агрегирует ее с помощью конечных автоматов (сценариев атак), статистического анализа и механизмов пороговой фильтрации;
- высший уровень агрегирует информацию с двух предыдущих и позволяет выявлять обычные и распределенные атаки, их реальный источник и прогнозировать его дальнейшее поведение на основе интеллектуального анализа.

Ядро системы обнаружения компьютерных атак должно быть четко разделено с системой визуализации и сигнализации.

3.3. Характеристика направлений и групп методов обнаружения вторжений

Существует два основных метода в сфере обнаружения атак: обнаружение злоупотреблений (misuse detection) и обнаружение аномалий (anomaly detection). Обнаружение злоупотреблений предпо-

Рис. 3.5. Типовая архитектура систем обнаружения атак

лагает наличие сигнатур атак. Основным недостатком таких систем является их неспособность обнаруживать новые или неизвестные атаки, т. е. записи о которых в системе отсутствуют. Обнаружение аномалий связано с построением профиля нормального поведения пользователя. Причем атакой считается любое отклонение от этого профиля. Главным преимуществом таких систем является принципиальная возможность определения ранее не встречавшихся атак.

Любая атака на систему может быть выявлена в ходе анализа сетевого трафика и/или системных ресурсов (журналы регистрации событий, файлы и т. п.). Поэтому СОВ различаются по уровню обнаружения: сетевого уровня и системного уровня. Системы обнаружения атак на уровне сети используют в качестве источника данных для анализа необработанные (raw) сетевые пакеты. В последнее время, в целях повышения качества распознавания, все чаще применяют комбинированные решения, проводящие мониторинг данных как на уровне сети, так и на уровне системы.

Рассмотрим принципиальную схему системы обнаружения атак, предназначенную для выявления и противодействия атакам злоумышленников на сетевом уровне. Система представляет собой специализированное программно-аппаратное обеспечение с типовой архитектурой, включающей в себя следующие компоненты (рис. 3.5):

- модули-датчики для сбора необходимой информации о сетевом трафике;
- модуль выявления (распознавания) атак, выполняющий обработку данных, собранных датчиками, с целью обнаружения информационных атак;
- модуль реагирования на обнаруженные атаки;

- модуль хранения конфигурационной информации, а также информации об обнаруженных атаках. Таким модулем, как правило, выступает стандартная СУБД;
- модуль управления компонентами СОВ.

Модуль выявления атак наиболее сложный и важный элемент СОВ, от которого зависит эффективность работы всей системы.

Необходимость стандартизации форматов данных и протоколов обмена данными, используемых в СОВ, обусловлена следующими причинами:

1. Для защиты ЛВС, подключенных к Интернету, от распределенных скоординированных атак необходимо обеспечить определенную степень взаимодействия между СОВ, используемыми для защиты различных точек входа в различные ЛВС. Например, в случае осуществления атаки против одной ЛВС, правила реагирования, на которую предусматривают изменение конфигурации межсетевого экрана путем блокирования IP-адреса источника атаки, соответствующие изменения должны быть произведены на всех межсетевых экранов, используемых для защиты всех остальных ЛВС. Для этого между различными СОВ должен осуществляться обмен информацией об источнике атаки и способе реагирования.

2. Центральным компонентом СОВ является специализированное программное ядро (analysis engine — анализатор), предназначенное для анализа данных, поступающих от сенсоров, и принятия решений о способах реагирования на подозрительные события. Стандартизация протоколов и форматов обмена данными между анализатором с одной стороны и сенсорами и средствами реагирования с другой, позволяет применять общее программное ядро анализатора с различными типами сенсоров и средств реагирования.

Среди методов, используемых в подсистеме анализа современных СОВ, можно выделить два направления: одно направлено на обнаружение аномалий в защищаемой системе, а другое — на поиск злоупотреблений [6, 10]. Каждое из этих направлений имеет свои достоинства и недостатки, поэтому в большинстве существующих СОВ применяются комбинированные решения, основанные на синтезе соответствующих методов. Идея методов, используемых для обнаружения аномалий, заключается в том, чтобы распознать, является ли процесс, вызвавший изменения в работе системы, действиями злоумышленника. Методы поиска аномалий приведены в табл. 3.1 и 3.2.

Выделяются две группы методов: с контролируемым обучением («обучение с учителем») и с неконтролируемым обучением («обуче-

Таблица 3.1

Обнаружение аномалии — контролируемое обучение («обучение с учителем»)

Методы обнаружения	Используется в системах	Описание метода
Моделирование правил	W&S	Система обнаружения в течение процесса обучения формирует набор правил, описывающих нормальное поведение системы. На стадии поиска несанкционированных действий система применяет полученные правила и в случае недостаточного соответствия сигнализирует об обнаружении аномалии
Описательная статистика	IDES, NIDES, EMERLAND, JiNao, HayStack	Обучение заключается в сборе простой описательной статистики множества показателей защищаемой системы в специальную структуру. Для обнаружения аномалий вычисляется «расстояние» между двумя векторами показателей — текущими и сохраненными значениями. Состояние в системе считается аномальным, если полученное расстояние достаточно велико
Нейронные сети	Hyperview	Структура применяемых нейронных сетей различна. Но во всех случаях обучение выполняется данными, представляющими нормальное поведение системы. Полученная обученная нейронная сеть затем используется для оценки аномальности системы. Выход нейронной сети говорит о наличии аномалии

ние без учителя»). Основное различие между ними заключается в том, что методы контролируемого обучения используют фиксированный набор параметров оценки и некие априорные сведения о значениях параметров оценки. Время обучения фиксировано. В неконтролируемом же обучении множество параметров оценки может изменяться с течением времени, а процесс обучения происходит постоянно.

Таблица 3.2

Обнаружение аномалии — неконтролируемое обучение («обучение без учителя»)

Методы обнаружения	Используется в системах	Описание метода
Моделирование множества состояний	DPEM, JANUS, Bro	Нормальное поведение системы описывается в виде набора фиксированных состояний и переходов между ними, где состояние есть не что иное, как вектор определенных значений параметров измерений системы
Описательная статистика	MIDAS, NADIR, HayStack, NSM	Этот метод аналогичен соответствующему методу в контролируемом обучении

Таблица 3.3

Обнаружение злоупотреблений — контролируемое обучение («обучение с учителем»)

Методы обнаружения	Используется в системах	Описание метода
Моделирование множества состояний	USTAT, IDIOT	Вторжение представляется как последовательность состояний, где состояние — вектор значения параметров оценки защищаемой системы. Необходимое и достаточное условие наличия вторжения — присутствие этой последовательности. Выделяют два основных способа представления сценария вторжений: 1) в виде простой цепочки событий; 2) с использованием сетей Петри, где узлы — события
Экспертные системы	NIDES, EMERLAND, MIDAS, DIDS	Экспертные системы представляют процесс вторжения в виде различного набора правил. Очень часто используются продукционные системы
Моделирование правил	NADIR, HayStack, JiNao, ASAX, Bro	Простой вариант экспертных систем
Синтаксический анализ	NSM	Системой обнаружения выполняется синтаксический разбор с целью обнаружения определенной комбинации символов, передаваемых между подсистемами и системами защищаемого комплекса

Цель второго направления (обнаружение злоупотреблений) — поиск последовательностей событий, определенных (администратором безопасности или экспертом во время обучения СОВ) как этапы реализации вторжения. Методы поиска злоупотреблений приведены в табл. 3.3. В настоящие время выделяются лишь методы с контролируемым обучением.

В табл. 3.4 представлены общие рекомендации по выбору СОВ в зависимости от типа атак, против которых должна быть сконцентрирована основная защита системы.

Реализованные в настоящее время в СОВ методы основаны на общих представлениях теории распознавания образов. В соответствии с ними для обнаружения аномалии на основе экспертной оценки формируется образ нормального функционирования информационной системы. Этот образ выступает как совокупность значений параметров оценки. Его изменение считается проявлением аномального функционирования системы. После обнаружения аномалии и оценки ее степени формируется суждение о природе изменений: является ли они следствием вторжения или допустимым отклонени-

Таблица 3.4

Рекомендации к выбору host-based или network-based СОВ

Тип атаки	Пример атаки	Network IDS	Host IDS
Отказ в обслуживании	SynFlood Attack	Отлично	Плохо
Сканирование и определение типа и параметров системы	Satan	Отлично	Хорошо
Атака на пароль	L0phtCrack	Плохо	Хорошо
Захват привилегий	Buffer Overflow	—	Отлично
Повреждение программного кода	Malformed URL	Плохо	Отлично
Вандализм	Melissa Virus	Плохо	Отлично
Кража информации	Targeting Key Sources	—	Хорошо
Мошенничество	B02K	Плохо	Хорошо
Аудит сети	Covering a Trail	—	Отлично
Атака на привилегии администратора безопасности	Backdoor insert	—	Отлично

ем. Для обнаружения злоупотреблений также используется образ (сигнатура), однако здесь он отражает заранее известные действия атакующего.

3.4. Сравнительный анализ существующих СОВ

Приведем краткое описание наиболее распространных в настоящее время СОВ. Для каждой системы описывается ее архитектура, используемая платформа и некоторые индивидуальные особенности [10]. В табл. 3.5 приведена краткая информация некоммерческих систем обнаружения компьютерных атак.

Таблица 3.5

Некоммерческие системы обнаружения компьютерных атак

Система	Производитель	Ссылки
AAFID	Purdue University, West Lafayette,IN, USA	www.cs.purdue.edu/coast/projects/autonomous-agents.html
ASAX	University of Namur, Belgium	www.ja.net/CERT/Software/asax/
NetSTAT	University of California at Santa Barbara	www.es.ucsb.edu/~kemm/netstat.html
Prelude	Yoann Vandoorselaere yoann@mandrakesoft.com Laurent Oudot oudot.laurent@wanadoo.fr	www.prelude-ids.org/
SHADOW	Naval Surface Warfare Center, Dahlgren Division	www.nswe.navy.mil/ISSEC/CID
Snort	Martin Roesch	www.snort.org/
SnortNet	Fyodor Yaroshkin	snortnet.scorpions.net/

3.4.1. Bro

Система Bro [13, 14] является разработкой Национальной лаборатории Лоуренса Беркли Калифорнийского университета, Беркли,

США. Система является открытой и распространяется по собственной открытой лицензии в исходных текстах и бинарных пакетах для ряда UNIX-платформ. Система предназначена для пассивного мониторинга сетевого трафика и поиска подозрительной активности. Обнаружение атак выполняется на нескольких уровнях: входящий сетевой трафик разбирается для выявления семантики уровня приложений, после чего полученная трасса событий прикладного уровня анализируется набором событийно-ориентированных анализаторов и сравнивается с шаблонами атак.

Bro использует специализированный язык управления политиками, позволяющий подстраивать поведение системы под защищаемую систему в соответствии с изменяющимися внешними условиями. При обнаружении атаки система может выполнить различные действия — записать сообщение в журнал, оповестить оператора, выполнить команды операционной системы.

Назначением системы является высокоскоростное обнаружение атак на сетевых каналах с высокой пропускной способностью (1 Гбит/с). Архитектурно Bro можно разделить на три основных компонента: библиотека libpcap для захвата пакетов, модуль генерации событий и интерпретатор сценариев.

Интерпретатор сценариев выполняет анализаторы событий, написанные на языке Bro. Данный набор сценариев является политикой безопасности сети, который определяет реакцию системы на различные события. Сценарий может генерировать сообщения, а также выполнять произвольные команды операционной системы, т. е. реагировать на атаки. В состав системы входит утилита snort2bro, которая транслирует сигнатуры Snort в сценарии Bro. Кроме трансляции, данная утилита выполняет оптимизацию сигнатур под анализатор Bro.

3.4.2. OSSEC

OSSEC HIDS является открытой узловой системой обнаружения атак [12]. В её задачи входит: анализ журналов, контроль целостности, обнаружение закладок, оповещение об атаках и активная реакция на атаки.

Система может быть установлена как в одиночной конфигурации на одном узле, так и в распределенной конфигурации на нескольких узлах — в таком случае одна из инсталляций становится сервером, а остальные — агентами системы. При этом управление агентами выполняется централизованно с сервера.

В состав системы входят несколько различных анализаторов. Анализатор журналов использует файлы журналов типовых прило-

жений для UNIX-систем, системные журналы Windows и некоторых приложений (Internet Information Server, IIS). Модуль обнаружения закладок сканирует файловую систему узла и ищет известные закладки по сигнатурам, а также неизвестные закладки и закладки на уровне ядра на основе обнаружения аномалий.

Модуль контроля целостности выполняет проверку наиболее критичных системных файлов (исполняемые, конфигурационные, файлы библиотек и т. п.). При первом запуске данный модуль создаёт базу данных критичных файлов и сохраняет в нее, помимо самих файлов, информацию целостности: параметры доступа, размер, информацию о владельцах, контрольные суммы MD5 и SHA1. Затем модуль периодически производит полное сканирование системы и сравнивает системные файлы с копиями в базе данных. В том случае, если какой-либо файл изменился, генерируется сообщение администратору.

Система OSSEC также включает в себя модуль корреляции сообщений об атаках, который расширяет возможности по анализу сообщений системы Snort, удаляет ложные сообщения и позволяет инициировать реакцию на сложные события.

3.4.3. STAT

Система STAT (State Transition Analysis Tool — средство анализа систем переходов) является результатом проекта Калифорнийского университета, Санта-Барбара, США [11]. Первые публикации о системе датированы 1992 г., последние — 2003 г. Основой используемого системой метода является описание исходной защищаемой системы в виде набора состояний компонентов и последующий анализ переходов из состояния в состояние в результате активных внешних воздействий. Состояния защищаемой системы определяются при настройке и конфигурации системы обнаружения атак. Для каждого состояния определяется характеристика защищенности.

Определяются *переходы* — изменения состояния защищаемой системы.

Атаки описываются в виде последовательностей переходов. Данный подход также является эвристическим и родственен подходу, используемому в экспертных системах, базирующихся на сигнатурах атак. Описание атак в виде последовательности переходов терминов состояний призвано избежать традиционных ограничений методов, основанных на сигнатурах и дать возможность описать шаблоны для целых классов типовых атак.

Основой системы является язык STATL — расширяемый язык, предназначенный для описания шаблонов атак в терминах STAT.

Базовый язык оперирует наиболее абстрактными понятиями, не зависящими от конкретной системы и ее конфигурации.

Язык позволяет достраивать себя, добавляя специфичные для конкретной системы *события*. Для каждого нового события описывается его *предикат*. К примеру, для расширения языка с целью определения событий, характерных для веб-сервера Apache, необходимо определить события, описывающие появляющиеся в журналах данного приложения записи. То есть событие будет иметь поля *host, ident, authuser, request, status* и прочие, определенные в Apache's Common Log Format. После этого требуется описать предикаты интересующих событий. Например, предикат isCGIrequest() будет возвращать *true*, если имел место вызов CGI-сценария. Описания событий и предикатов группируются в Language Extension Module (модуль расширения языка) и в последствии их можно использовать в описании сценариев атак для STAT. Поток реальных событий сравнивается со сценариями ядром STAT. Все конструкции языка STATL и его расширения транслируются в язык C++. Основной функциональной частью системы является *ядро* STAT. Этот компонент оперирует абстрактными объектами и событиями, не зависящими от конкретной системы. Он сравнивает входящий поток событий с имеющимися сценариями атак и выполняет непосредственно функцию обнаружения. Для генерации потока событий используется *источник событий* — программный компонент системы обнаружения, осуществляющий преобразование информации из системных источников, таких как журналы регистрации, в формат, пригодный для функционирования ядра STAT.

В системе также могут использоваться *модули реакции* — связанные с ядром компоненты, осуществляющие реагирование на обнаруженную атаку.

Данная система является открытой и позволяет строить масштабируемые системы обнаружения. На основе STAT строятся агенты или сенсоры системы обнаружения, все остальное — вопросы архитектуры и взаимодействия агентов в распределенной системе. Этому и посвящены все работы xSTAT, которые применяют данную технологию для конструирования систем обнаружения атак различных типов — узловые, сетевые.

На рис. 3.6 представлена архитектура системы обнаружения атак, построенной на основе STAT.

Основные компоненты:

STAT sensor — модуль обнаружения атак на узле на основе ядра STAT.

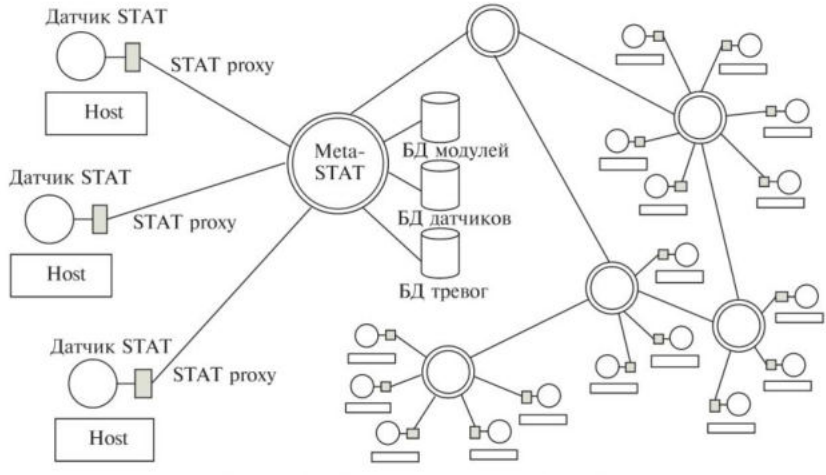

Рис. 3.6. Структура системы STAT

STAT proxy — модуль, связывающий сенсоры с центральным модулем MetaSTAT.

MetaSTAT — модуль сбора информации об атаках, уведомления администратора, хранения информации об атаках.

Языком реализации системы NetSTAT является язык C++. Система работает под управлением ОС Linux и Solaris.

3.4.4. Prelude

Система Prelude является системой с открытыми исходными текстами. Начало разработки — 1998 г. Она изначально задумывалась как гибридная СОВ, которая могла бы помочь администратору сети отслеживать активность как на уровне сети, так и на уровне отдельных узлов. Система распределенная и состоит из следующих компонентов [16]:

- сетевые сенсоры — различные сенсоры, анализирующие данные на уровне сети на основе сигнатурного анализа. Сенсоры генерируют сообщения об обнаружении атак и отправляют их модулям управления. Система Prelude использует в качестве сетевого сенсора систему Snort;
- узловые сенсоры — различные сенсоры уровня системы, анализирующие журналы регистрации ОС, приложений;
- модули управления — процессы, которые получают и обрабатывают сообщения сенсоров;
- агенты реагирования — реализуют сгенерированную менеджером реакцию на атаку.

Сенсоры генерируют сообщения об обнаружении аномалий и отправляют их модулям управления. Существующий набор сенсоров позволяет анализировать данные журналов регистрации таких систем и приложений, как межсетевой экран IPFW, входящий в состав ОС FreeBSD, NetFilter ОС Linux 2.4.x, маршрутизаторы Cisco и Zyxel, GRSecurity и типовые сервисы ОС UNIX.

Различаются следующие виды модулей управления:

- модули журнализации — отвечают за регистрацию сообщений в журналах регистрации или базах данных. В настоящее время реализованы модули для MySQL, PostgreSQL;
- модули реагирования — анализируют сообщение и генерируют возможную ответную реакцию СОВ на атаку. Возможны такие виды реакции, как блокирование нарушителя на межсетевом экране (NetFilter, IPFilter). В дальнейшем возможны такие типы реакции, как изоляция нарушителя и сужение пропускной способности канала нарушителя.

Интерфейс, основанный на протоколе http, предоставляет возможность получать статистику и управлять системой при помощи web-браузера.

На рис. 3.7 представлена логическая схема соединения компонентов СОВ Prelude. У системы Prelude есть несколько особенностей, которые отличают ее от других современных открытых СОВ. Система везде, где возможно, построена на использовании открытых стандартов. Так, для обмена сообщениями используется формат IDMEF (Intrusion Detection Message Exchange Format), оптимизированный для высокоскоростной обработки. Это позволяет в дальнейшем интегрировать компоненты в системы сторонних производителей и наоборот.

При разработке системы особое внимание было уделено вопросам безопасности. Каналы передачи данных шифруются по протоколу SSL, кроме того, используется специализированная библиотека, которая предотвращает классические ошибки выхода за границы массивов и переполнения буферов.

Дополнительные модули анализа сетевых данных делают систему устойчивой к некорректным сетевым пакетам на разных уровнях стека и выходу ее компонентов из строя. Такие атаки, как отправка пакетов с неправильными контрольными суммами, обнуленными флагами TCP, ресинхронизация сессий, случайная отправка и «обрезание» сегментов системой, игнорируются и не приводят к отказу компонентов СОВ.

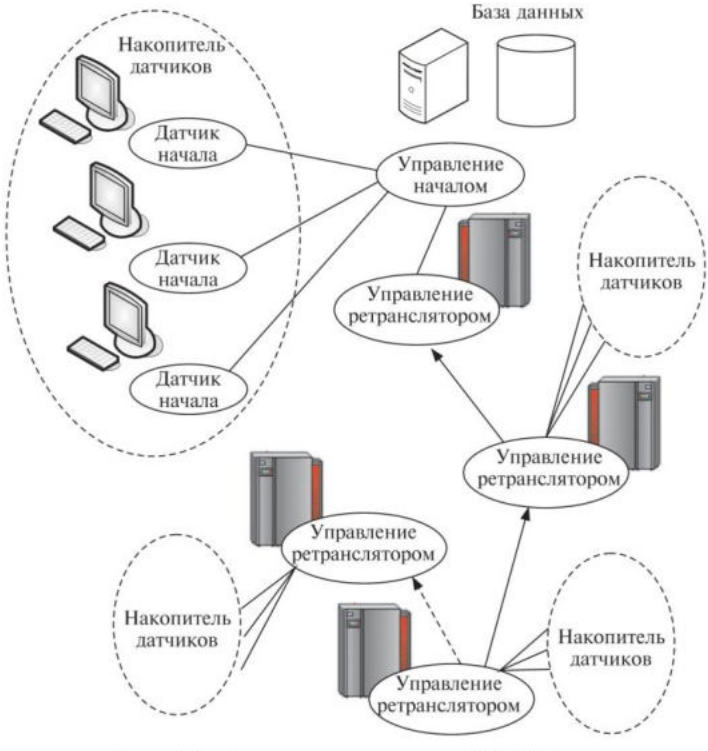

Рис. 3.7. Логическая структура СОВ Prelude

3.4.5. Snort

Snort является открытой Network Intrusion Detection/Prevention System (NIDS/NIPS) системой, позволяющей проводить анализ трафика в реальном времени, а также логинг пакетов в IP сетях [5, 15]. Она позволяет анализировать протоколы верхних уровней на предмет поиска и соответствия нужного содержимого и может использоваться для обнаружения различных атак, таких как buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts и других. Snort использует гибкий язык написания правил, который позволяет охарактеризовать интересный трафик для сбора или анализа, а также имеет детектор атак, имеющий модульную архитектуру. Обнаруживает атаки исключительно на основе анализа сетевого трафика. Основным методом обнаружения атак, используемым в системе, является обнаружение злоупотреблений на основе описания сигнатур атак. В системе используется простой язык описания сигнатур атак, который полностью описан в документации и позволяет администраторам системы дополнять базу сигнатур сво-

ими сигнатурами. Каждое правило на этом языке состоит из двух частей: условие применения и действие.

Пример правила системы Snort:

$$\text{alerttcpanyany} - > 10.1.1.0/2480$$
$$(\text{content} : \text{"/cgi} - \text{bin/phf"}; \text{msg} : \text{"PHFprobe!"};)$$

Это правило определяет, что любой сегмент TCP, направленный на порт 80 на любой адрес в сети 10.1.1.0/24, и при этом имеющий в поле данных строку «/cgibin/phf», является подозрительным и необходимо послать уведомление администратору.

Кроме того, в последних версиях системы появилась специальная конструкция языка сигнатур, позволяющая классифицировать сетевой трафик по степени потенциальной опасности. Степень опасности определяется экспертом, который формирует сигнатуру атаки.

В настоящее время система находится в стадии активной разработки: каждые несколько месяцев появляются новые версии системы и новые функции.

Архитектура системы Snort целиком разрабатывалась из соображений эффективности и скорости работы. Поэтому она предельно проста и состоит из следующих подсистем: декодер пакетов, ядро обнаружения и подсистемы оповещения и реагирования. Декодер пакетов реализует набор процедур для последовательной декомпозиции пакетов в соответствии с уровнями сетевого стека, т. е. принятый кадр последовательно преобразуется в пакет, сегмент и блок данных с применением специфичных для данного уровня сигнатур атак. В настоящее время поддерживаются протоколы канального уровня Ethernet, SLIP, PPP. Ядро выстраивает имеющиеся правила в так называемой *цепи правил* — двумерной последовательности правил, где правила с общей частью условий применения объединяются в одно звено цепи, а несовпадающие компоненты правил строятся цепью во втором измерении от полученного звена. Это сделано для ускорения анализа сетевого трафика. Каждый пакет проходит по цепочке от корня, первое подходящее правило выполняет свой блок действий и проход завершается.

На рис. 3.8 представлен небольшой пример такой цепи правил.

Кроме модуля анализа трафика на основе правил, к ядру обнаружения могут подключаться модули сторонних разработчиков (препроцессоры) и производить анализ на одном из уровней декомпозиции пакета. С помощью таких модулей можно добавлять функциональность ядру обнаружения атак и реализовывать различные

Рис. 3.8. Цепи правил системы Snort

методы обнаружения. Препроцессоры выполняют декомпозицию сетевого трафика и проверку соответствия спецификациям протоколов, дефрагментацию и т. п. В поставку системы входит несколько модулей, например модуль обнаружения сканирования портов, модуль обнаружения Unicode-атаки на веб-сервер компании Microsoft и другие.

Кроме того, в одной из версий в составе Snort был модуль статистического анализа, который предназначен для обнаружения аномалий в сетевом трафике. Подсистема оповещения и реагирования отвечает за сохранение результатов анализа трафика в журналы регистрации самой системы Snort либо вывод этой информации через системные службы регистрации событий ОС. Например, в UNIX-подобной ОС это может быть сервис регистрации событий *syslog*. Система Snort реализована под множество UNIX платформ.

Snort можно использовать в трёх различных вариантах в качестве:

- пакетного сниффера (packet sniffer) по типу утилиты tcpdump;
- логгера пакетов (packet logger) для изучения сетевого трафика;
- NIDS или NIPS.

Для работы IDS/IPS Snort необходим как можно более мощный сервер с большим объёмом дискового пространства для хранения базы событий. Для развертывания Snort необходимо установить и настроить ряд программ:

- операционная система FreeBSD или MS Windows;
- Snort — сам сенсор с детекторами для обнаружения атак;
- Libpcap — сниффер для захвата пакетов;
- СУБД MySQL — для хранения базы данных событий;

- PHP — язык разработки для Web;
- Apache — web-сервер;
- Basic Analysis and Security Engine (BASE) — консоль управления и просмотра событий (alerts);
- Oinkmaster — утилита для обновления сигнатур и некоторые другие.

3.4.6. SnortNet

SnortNet это распределенное расширение системы Snort, целью которого является придание ей дополнительных возможностей по масшабируемости и расширяемости [54]. Система состоит из нескольких программных модулей: сенсоров, модулей пересылки сообщений и станции мониторинга. Система позволяет проводить мониторинг сетевого трафика и осуществлять информирование станции мониторинга обо всех обнаруженных аномалиях в поведении сетевых объектов. Система использует Snort в качестве сетевого сенсора. В качестве протокола обмена данными система использует протокол Internet Alert Protocol (протокол передачи сигналов тревоги — IAP). Для шифрования каналов передачи данных, аутентификации и контроля доступа система использует библиотеки SSL и TCP wrappers.

Обнаружение атак производится сенсорами Snort, после чего сообщения отправляются по протоколу IAP через модули пересылки сообщений (proxy) на станцию мониторинга. Модули пересылки сообщений и станцию мониторинга рекомендуется располагать в демилитаризованной зоне сети (рис. 3.9).

Система SnortNet протестирована под операционными системами Linux, FreeBSD, OpenBSD.

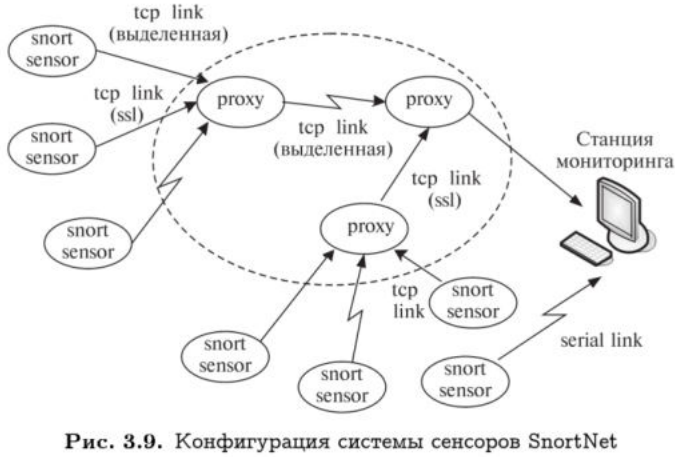

Рис. 3.9. Конфигурация системы сенсоров SnortNet

3.4.7. AAFID

Система AAFID (Autonomous Agents for Intrusion Detection) разработана в университете Purdue, West Lafayette, IN, USA (www.cs.purdue.edu/coast/projects/autonomous-agents.html). Первые публикации по системе датированы 1998 г., последние — 1999 г. AAFID — это одновременно название распределенной архитектуры систем обнаружения атак и собственно системы обнаружения атак [17]. Основой системы являются *автономные агенты* обнаружения. Наиболее интересна в системе именно ее архитектура. Данная система базируется на работах Crosbie и Spafford, которые предложили использование автономных агентов, работающих на основе генетических алгоритмов и адаптирующихся к поведению пользователей. Идея использования генетических алгоритмов не была реализована, но архитектурные идеи были воплощены в системе AAFID.

Основными компонентами системы являются *агенты, трансиверы, мониторы, фильтры*. Система AAFID является полностью распределенной. На любом узле в ЛВС может быть размещено любое число агентов, наблюдающих за интересными с их точки зрения событиями на данном узле.

Агент — автономная программная компонента системы обнаружения атак, функционирование которой зависит лишь от ОС, под управлением которой она работает. То есть функционирование агента не зависит от других компонентов системы обнаружения атак. Агенты могут использовать *фильтры* для сбора информации о поведении объектов в архитектурно-независимом представлении. Все агенты на одном узле передают собранную информацию одному трансиверу.

Трансивер — компонент системы обнаружения, который управляет запуском и остановкой агентов на данном узле. Каждый узел, на котором есть агенты, имеет один трансивер. Кроме того, трансиверы могут осуществлять некоторую редукцию получаемой от агентов информации в более обобщенное представление. Трансиверы передают полученную информацию одному или нескольким мониторам. Каждый монитор наблюдает и взаимодействует с несколькими трансиверами.

Мониторы функционируют на уровне защищаемой системы в целом, поэтому они могут анализировать получаемую информацию с учетом корреляции событий в разных областях защищаемой системы. Мониторы могут располагаться в иерархическом порядке. Монитор также является связующим компонентом между системой

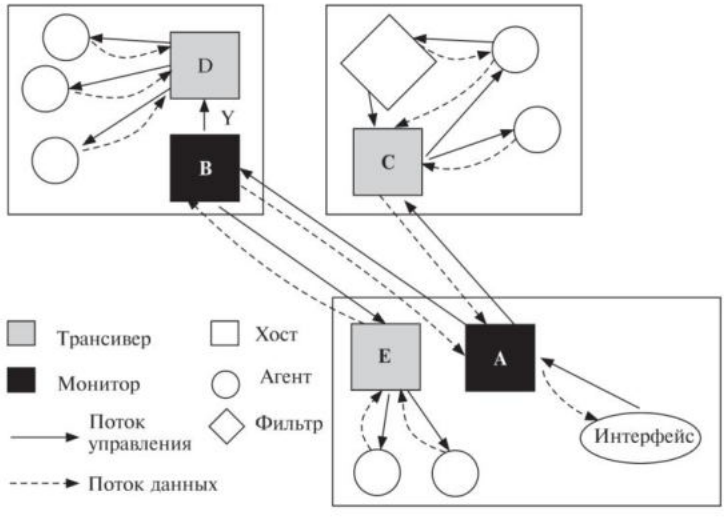

Рис. 3.10. Архитектура системы AAFID, основные компоненты системы

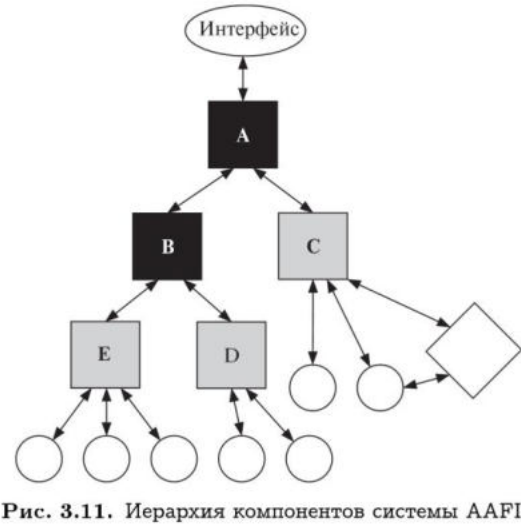

Рис. 3.11. Иерархия компонентов системы AAFID

обнаружения и пользовательским интерфейсом — он получает управляющие команды и отдает полученную и проанализированную информацию.

На рис. 3.10 представлена архитектура, а на рис. 3.11 — иерархия компонентов системы обнаружения.

Остановимся более подробно на описании компонентов системы AAFID. Агент наблюдает за конкретными аспектами функциониро-

вания узла и уведомляет соответствующий трансивер об аномальных событиях на узле. Агенты не могут непосредственно генерировать сообщения для пользователя. Таким образом, трансиверы получают полную информацию о функционировании узла, а мониторы — о сети в целом. Функциональность агента ничем не ограничена, он может реализовывать любой метод обнаружения атак. Агенты могут быть предназначены для обнаружения аномалий и злоупотреблений любого типа.

Фильтры предназначены для сбора однотипной информации из различных источников (различных системных журналов или от наблюдателей). Фильтры также выполняют функции уровня представления модели OSI (Open System Interconnect): преобразуют форматы данных, собираемых из аналогичных источников, но на разных архитектурах, к единому виду.

Один фильтр может предоставлять данные нескольким агентам. Типичный пример применения фильтра — использование его для преобразования журналов регистрации различных версий ОС UNIX в формат, понимаемый агентами. Это снимает необходимость реализовывать различные агенты для разных платформ.

На каждый источник данных существует только один фильтр. Агенты могут *подписываться* на получение данных от фильтра.

Трансивер — интерфейс взаимодействия между узловыми компонентами системы обнаружения и сетевыми компонентами системы обнаружения. Основными функциями трансивера являются управление агентами на данном узле (запуск, останов), сбор и анализ данных от агентов, передача данных и результатов анализа мониторам или другим агентам.

Мониторы — наивысшие в иерархии компоненты системы. Функционально они похожи на трансиверы с тем отличием, что сбор и анализ информации происходит не на одном узле, а на части сети или на всей сети. Мониторы могут управлять трансиверами и другими мониторами. Мониторы имеют механизмы взаимодействия с пользовательским интерфейсом и являются точками доступа к системе обнаружения атак.

В системе AAFID практически не реализованы агенты, необходимые для эффективного обнаружения атак различных классов, и основной ценностью данной системы является ее архитектура. В документации хорошо описаны и стандартизированы интерфейсы между компонентами системы обнаружения атак. Особое внимание уделяется вопросам безопасного взаимодействия компонентов распределенной системы обнаружения. Насколько можно судить по

описанию системы, в настоящее время все рабочие агенты реализованы по принципу экспертной системы, они обнаруживают заранее описанные ненормальные ситуации и события на узле и в сети.

Система все еще находится на стадии прототипа. Последнее обновление имело место в сентябре 1999 года, и данная версия была реализована и протестирована под операционными системами Solaris и Linux. Языком реализации является алгоритмический язык Perl. На нем реализованы все компоненты системы.

4 АНАЛИЗ СЕТЕВОГО ТРАФИКА И КОНТЕНТА

Существуют два не исключающих друг друга подхода к выявлению сетевых атак: анализ сетевого трафика и анализ контента. В первом случае анализируются только заголовки сетевых пакетов, во втором — их содержимое.

Конечно, наиболее полный контроль информационных взаимодействий может быть обеспечен только анализом всего содержимого сетевых пакетов, включая их заголовки и области данных. Однако с практической точки зрения эта задача является трудновыполнимой из-за огромного объема данных, которые приходилось бы анализировать. Современные СОВ начинают испытывать серьезные проблемы уже в сетях с производительностью 100 Мбит/с. Поэтому в большинстве случаев целесообразно использовать для выявления атак методы анализа сетевого трафика, в некоторых случаях сочетая их с анализом контента.

Сигнатура сетевой атаки концептуально практически не отличается от сигнатуры вируса. Она представляет собой набор признаков, позволяющих отличить сетевую атаку от других видов сетевого трафика.

4.1. Программы анализа и мониторинга сетевого трафика

Мониторинг трафика жизненно важен для эффективного управления сетью. Он является источником информации о функционировании корпоративных приложений, которая учитывается при распределении средств, планировании вычислительных мощностей, определении и локализации отказов, решении вопросов безопасности [39].

В недалеком прошлом мониторинг трафика был относительно простой задачей. Как правило, компьютеры объединялись в сеть на основе шинной топологии, т. е. имели разделяемую среду передачи. Это позволяло подсоединить к сети единственное устройство,

с помощью которого можно было следить за всем трафиком. Однако требования к повышению пропускной способности сети и развитие технологий коммутации пакетов, вызвавшее падение цен на коммутаторы и маршрутизаторы, обусловили быстрый переход от разделяемой среды передачи к высоко сегментированным топологиям. Общий трафик уже нельзя увидеть из одной точки. Для получения полной картины требуется выполнять мониторинг каждого порта. Использование соединений типа «точка-точка» делает неудобным подключение приборов, да и понадобилось бы слишком большое их число для прослушивания всех портов, что превращается в чересчур дорогостоящую задачу. Вдобавок сами коммутаторы и маршрутизаторы имеют сложную архитектуру, и скорость обработки и передачи пакетов становится важным фактором, определяющим производительность сети.

Одной из актуальных научных задач в настоящее время является анализ (и дальнейшее прогнозирование) самоподобной структуры трафика в современных мультисервисных сетях. Для решения этой задачи необходим сбор и последующий анализ разнообразной статистики (скорость, объемы переданных данных и т. д.) в действующих сетях. Сбор такой статистики в том или ином виде возможен различными программными средствами. Однако существует набор дополнительных параметров и настроек, которые оказываются весьма важными при практическом использовании различных средств.

Приведем обзор основных возможностей некоторых распространенных программ-анализаторов сетевого трафика.

4.1.1. Программы-анализаторы сетевого трафика

BMExtreme. Это новое название хорошо известной многим программы Bandwidth Monitor. Ранее программа распространялась бесплатно, теперь же она имеет три версии, и бесплатной является только базовая. В этой версии не предусмотрено никаких возможностей, кроме, собственно, мониторинга трафика, поэтому вряд ли можно считать ее конкурентом других программ. По умолчанию BMExtreme следит как за интернет-трафиком, так и за трафиком в локальной сети, однако мониторинг в LAN при желании можно отключить.

BWMeter. Эта программа имеет не одно, а два окна слежения за трафиком: в одном отображается активность в Интернете, а в другом — в локальной сети. Программа имеет гибкие настройки для мониторинга трафика. С ее помощью можно определить, нужно ли следить за приемом и передачей данных в Интернет только

с этого компьютера или со всех компьютеров, подключенных к локальной сети, установить диапазон IP-адресов, порты и протоколы, для которых будет или не будет производиться мониторинг. Кроме этого, можно отключить слежение за трафиком в определенные часы или дни. Для каждого ПК можно задать максимальную скорость приема и передачи данных, а также одним щелчком мыши запретить сетевую активность.

При весьма миниатюрном размере программа обладает множеством возможностей, часть из которых можно представить так:

- мониторинг любых сетевых интерфейсов и любого сетевого трафика;
- мощная система фильтров, позволяющая оценить объем любой части трафика — вплоть до конкретного сайта в указанном направлении или трафика с каждой машины в локальной сети в указанное время суток;
- неограниченное количество настраиваемых графиков активности сетевых соединений на основе выбранных фильтров;
- управление (ограничение, приостановка) потоком трафика на любом из фильтров;
- удобная система статистики (от часа до года) с функцией экспорта;
- возможность просмотра статистики удаленных компьютеров с BWMeter;
- гибкая система оповещений и уведомлений по достижении определенного события;
- максимальные возможности по настройке, в том числе внешнего вида;
- возможность запуска как сервиса.

Bandwidth Monitor Pro. Её разработчики очень много внимания уделили настройке окна мониторинга трафика. Во-первых, можно определить, какую именно информацию программа будет постоянно показывать на экране. Это может быть количество полученных и переданных данных (как отдельно, так и в сумме) за сегодня и за любой указанный промежуток времени, среднюю, текущую и максимальную скорость соединения.

Отдельно стоит сказать о системе оповещений, которая удачно реализована. Можно задавать поведение программы при выполнении заданных условий, которыми могут быть передача определенного количества данных за указанный период времени, достижение максимальной скорости загрузки, изменение скорости соединения и

пр. Если на компьютере работает несколько пользователей и необходимо следить за общим трафиком, то программу можно запускать как службу. В этом случае Bandwidth Monitor Pro будет собирать статистику всех пользователей, которые заходят в систему под своими логинами.

DUTraffic. От всех программ обзора DUTraffic отличает бесплатный статус. Как и коммерческие аналоги, DUTraffic может выполнять разнообразные действия при выполнении тех или иных условий. Так, например, он может проигрывать аудиофайл, показывать сообщение или же разрывать соединение с Интернетом, когда средняя или текущая скорость загрузки меньше заданного значения, когда продолжительность интернет-сессии превышает указанное число часов, когда передано определенное количество данных. Кроме этого, различные действия могут выполняться циклически, например каждый раз, когда программа фиксирует передачу заданного объема информации. Статистика в DUTraffic ведется отдельно для каждого пользователя и для каждого соединения с Интернетом. Программа показывает как общую статистику за выбранный промежуток времени, так и информацию о скорости, количестве переданных и принятых данных и финансовых затратах за каждую сессию.

Система мониторинга Cacti. Cacti это open-source веб-приложение (соответственно отсутствует установочный файл). Cacti собирает статистические данные за определённые временные интервалы и позволяет отобразить их в графическом виде. Система позволяет строить графики при помощи RRDtool. Преимущественно используются стандартные шаблоны для отображения статистики по загрузке процессора, выделению оперативной памяти, количеству запущенных процессов, использованию входящего/исходящего трафика.

Интерфейс отображения статистики, собранной с сетевых устройств, представлен в виде дерева, структура которого задается самим пользователем. Как правило, графики группируют по определенным критериям, причем один и тот же график может присутствовать в разных ветвях дерева (например, трафик через сетевой интерфейс сервера — в той, которая посвящена общей картине интернет-трафика компании, и в ветви с параметрами данного устройства). Есть вариант просмотра заранее составленного набора графиков, и есть режим предпросмотра. Каждый из графиков можно рассмотреть отдельно, при этом он будет представлен за последние день, неделю, месяц и год. Есть возможность самостоятельного

Таблица 4.1

Основные характеристики программ мониторинга сетевого трафика

Параметр	BM-Extreme	BW-Meter	Bandwidth Monitor Pro	DU-Traffic	Cacti
Размер установочного файла, Мбайт	0,473	1,91	1,05	1,4	–
Язык интерфейса	русский	русский	английский	русский	английский
График скорости	+	–	+	+	–
График трафика	–	+	+	–	+
Экспорт/импорт (формат файла экспорта)	–/–	+/+ (*.csv)	–/–	–/–	+/+ (*.xls)
Запуск мониторинга по требованию	–	+	–	–	+
Минимальный временной шаг между отсчётами данных, с	300	1	60	1	1
Возможность изменения минимального шага между отсчётами данных	+	+	+	–	+

выбора временного промежутка, за который будет сгенерирован график, причем сделать это можно как указав календарные параметры, так и просто выделив мышкой определенный участок на нем.

В табл. 4.1 представлены сравнительные основные характеристи представленных программ мониторинга сетевого трафика.

4.1.2. Обзор программ-анализаторов (снифферов) сетевого трафика

Анализатор трафика, или сниффер, — сетевой анализатор трафика, программа или программно-аппаратное устройство, предназначенное для перехвата и последующего анализа либо только анализа сетевого трафика, предназначенного для других узлов.

Анализ прошедшего через сниффер трафика позволяет:

- обнаружить паразитный, вирусный и закольцованный трафик, наличие которого увеличивает загрузку сетевого оборудования и каналов связи (снифферы здесь малоэффективны; как правило, для этих целей используют сбор разнообразной статистики серверами и активным сетевым оборудованием и её последующий анализ);
- перехватить любой незашифрованный (а порой и зашифрованный) пользовательский трафик с целью получения паролей и другой информации;
- локализовать неисправность сети или ошибку конфигурации сетевых агентов (для этой цели снифферы часто применяются системными администраторами).

Поскольку в «классическом» снифферe анализ трафика происходит вручную, с применением лишь простейших средств автоматизации (анализ протоколов, восстановление TCP-потока), то он подходит для анализа лишь небольших его объёмов.

Wireshark (ранее — Ethereal). Программа-анализатор трафика для компьютерных сетей Ethernet и некоторых других. Имеет графический пользовательский интерфейс. Wireshark — это приложение, которое «знает» структуру самых различных сетевых протоколов и поэтому позволяет разобрать сетевой пакет, отображая значение каждого поля протокола любого уровня. Поскольку для захвата пакетов используется библиотека Pcap (Packet Capture), существует возможность захвата данных только из тех сетей, которые поддерживаются этой библиотекой. Тем не менее, Wireshark умеет работать с множеством форматов входных данных, соответственно можно открывать файлы данных, захваченных другими программами, что расширяет возможности захвата.

Iris Network Traffic Analyzer (NTA). Помимо стандартных функций сбора, фильтрации и поиска пакетов, а также построения отчетов, программа предлагает уникальные возможности для реконструирования данных. Iris NTA помогает детально воспроизвести сеансы работы пользователей с различными web-ресурсами и даже позволяет имитировать отправку паролей для доступа к защищенным web-серверам с помощью cookies. Уникальная технология реконструирования данных, реализованная в модуле дешифрования, преобразует сотни собранных двоичных сетевых пакетов в привычные глазу электронные письма, web-страницы, сообщения ICQ и др. eEye Iris позволяет просматривать незашифрованные сообщения web-почты и программ мгновенного обмена сообщениями, расширяя возможности имеющихся средств мониторинга и аудита.

Анализатор пакетов eEye Iris позволяет зафиксировать различные детали атаки, такие, как дата и время, IP-адреса и DNS-имена компьютеров хакера и жертвы, а также использованные порты.

Ethernet Internet Traffic Statistic (EITS). Эта система показывает количество полученных и принятых данных (в байтах, всего и за последнюю сессию), а также скорость подключения. Для наглядности собираемые данные отображаются в режиме реального времени на графике. Работает без инсталляции, интерфейс — русский и английский.

Утилита для контроля за степенью сетевой активности показывает количество полученных и принятых данных, ведя статистику за сессию, день, неделю и месяц.

Таблица 4.2

Сравнительный анализ характеристик программ-анализаторов сетевого трафика

Параметр	Wireshark	Iris NTA	EITS	CT
Размер установочного файла, Мбайт	17,4	5,04	0,651	7,2
Язык интерфейса	английский	русский	англ./ русский	русский
График скорости	+	+	—	—
График трафика	—	—	+	+
Экспорт/импорт (формат файла экспорта)	+/– (*.txt, *.px, *.csv, *.psml, *.pdml, *.c)	–/–	–/–	–/–
Запуск мониторинга по требованию	–	–	–	–
Минимальный шаг между отсчётами данных, с	0,001	1	1	1
Возможность изменения минимального шага между отсчётами данных	+	+	–	–

CommTraffic (CT) Это сетевая утилита для сбора, обработки и отображения статистики интернет-трафика через модемное (dial-up) или выделенное соединение. При мониторинге сегмента локальной сети CommTraffic показывает интернет-трафик для каждого компьютера в сегменте.

CommTraffic включает в себя легко настраиваемый, понятный пользователю интерфейс, показывающий статистику работы сети в виде графиков и цифр.

В табл. 4.2 представлен сравнительный анализ характеристик рассмотренных программ-анализаторов сетевого трафика.

4.2. Получение и подготовка исходных данных для анализа свойств аномалий трафика

В общем виде алгоритм анализа сетевого трафика выглядит следующим образом (рис. 4.1).

При включении режима захвата сетевых пакетов сетевой адаптер переводится в режим Promiscuous и фиксирует любой пакет, прошедший через интерфейс. Это обусловлено технологией передачи информации в сетях Ethernet. Определённые ограничения в режиме анализа дают возможность уменьшить объём анализируемой информации, отсекая излишки неактуальных данных и увеличив быстродействие системы.

В качестве хранилищ используются как SQL-базы данных, так и сформированные двоичные текстовые файлы, хранящие числовые характеристики.

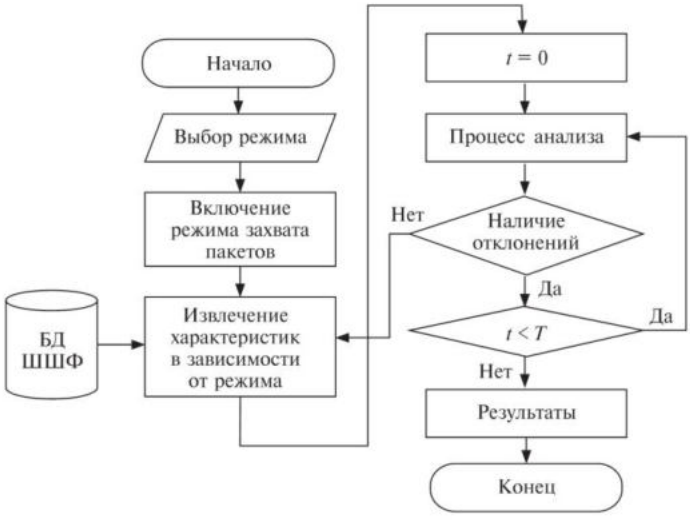

Рис. 4.1. Алгоритм процедуры контроля сети. БД ШШФ — банк данных шаблонов штатного функционирования сети

Исходными данными трафика, анализируемого впоследствии на наличие аномалий, предлагается принимать данные, полученные с помощью программы-сниффера Wireshark (или любого другого программного комплекса, осуществляющего задачи снятия дампа сетевой активности).

Программный комплекс Wireshark может использоваться как для записи «живого» трафика с интерфейсов, непосредственно входящих в анализирующую систему, так и для анализа снятого и сохраненного до этого трафика.

TCPdump данные. TCPdump (или Windump для Windows) — популярное и широко применяемое программное средство, позволяющее детально исследовать процесс передачи информации в сети. Вывод *tcpdump* содержит данные пакетов сетевых соединений в порядке появления пакета в сети. Перед сбором информации данные пакета должны предварительно обрабатываться. Конверторы *tcpdump* данных преобразовывают записи соединения в множество особенностей (т. е. атрибутов), например *time* (стартовое время соединения, т. е. *timestamp* первого пакета), *dur* (продолжительность соединения), *src* и *dst* (хост источник и адресат), *bytes* (число байтов данных, отправленных из источника до адресата), *srv* (сервис, т. е. порт адресата), *flag* (как соединение соответствует сетевому протоколу) и т. д. Эти существенные особенности суммируют информацию пакетного уровня в пределах соединения.

4.3. Анализ образцов трафика

Приведем анализ наборов данных, полученных DARPA Intrusion Detection Evaluation Group в MIT Lincoln Laboratory.

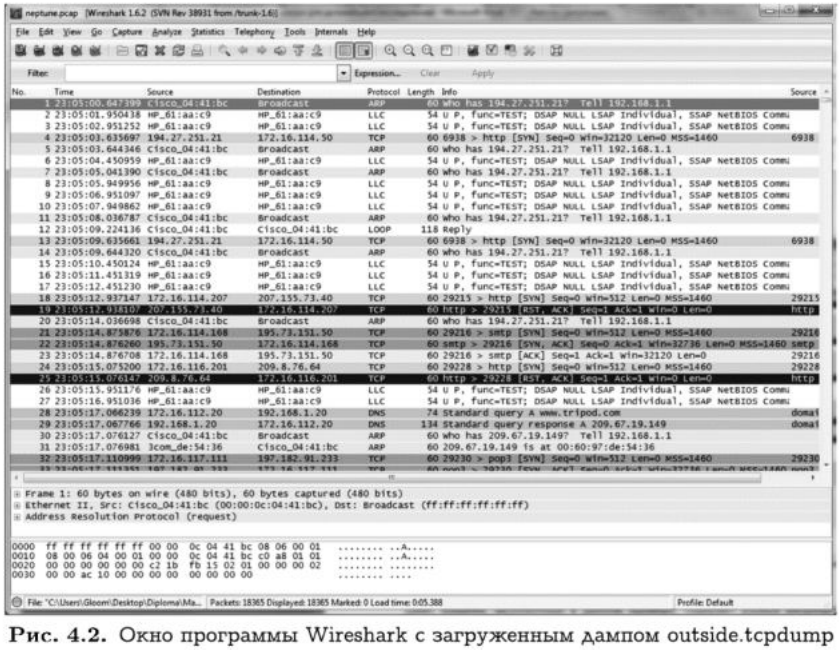

Рис. 4.2. Окно программы Wireshark с загруженным дампом outside.tcpdump

Рис. 4.3. Окно инструмента IO Grapghs с выведенной статистикой с временным делением 10 с

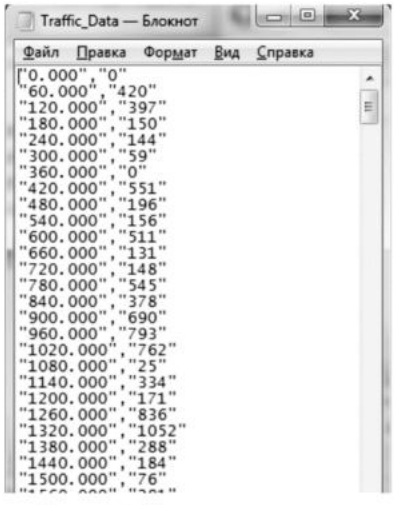

Рис. 4.4. Статистические данные, перенесенные в текстовый файл. Слева — временные значений в секундах, справа — статистика числа сообщений

Процесс подготовки исходных данных состоит из трех шагов.

1. Загрузка полученного дампа трафика в Wireshark (рис. 4.2).

Дамп трафика в данном примере содержит 246827 входящих и исходящих сообщений данных различных протоколов.

Комплекс Wireshark включает множество методов анализа и сбора статистических данных, необходимых в работе сетевых инженеров. Для снятия статистики сообщений, распределенных с некоторой частотой существует инструмент IO Graphs (рис. 4.3).

Для сохранения статистики достаточно нажать кнопку Copy и вставить запомненные в буфере обмена данные в любой текстовой файл (рис. 4.4).

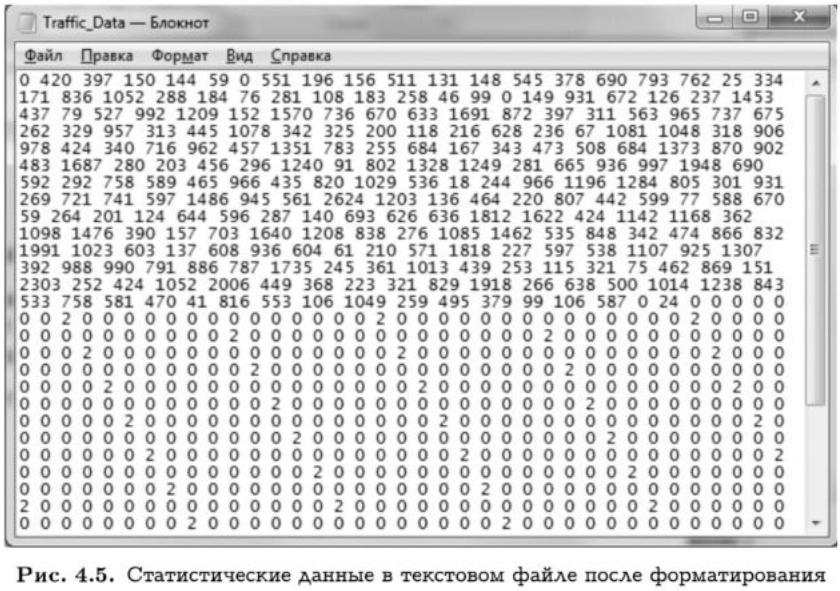

Рис. 4.5. Статистические данные в текстовом файле после форматирования

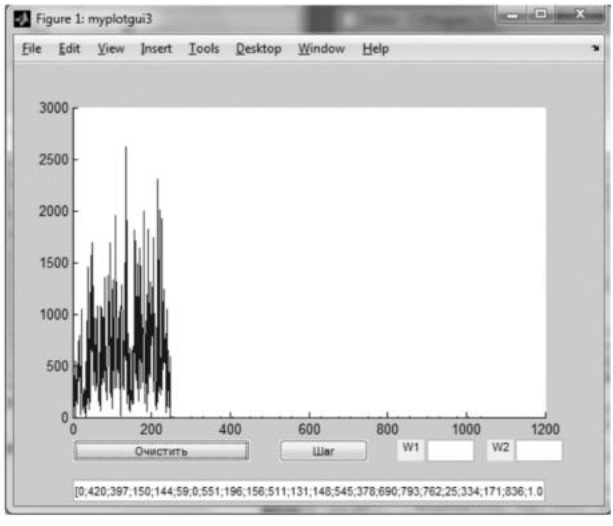

Рис. 4.6. Окно приложения анализа сетевых аномалий с загруженными данными, полученными из файла outside.tcpdump после форматирования данных

2. Загрузка и обработка полученного текстового файла в комплексе Matlab (рис. 4.5).

В целях форматирования полученных данных была разработана функция Convert, которая преобразует записанную в файле информацию, удаляя временные отметки и символы разделителя.

Запускается фунция по команде

» Convert('Traffic_Data.txt');

nrows = 1041

ncols = 1

3. Загрузка полученных и отформатированных данных в разработанное приложение для анализа сетевых аномалий (рис. 4.6).

4.3.1. Трассы и их анализ

Для синтеза и анализа алгоритмов обнаружения атак часто используют базу данных KDD-99 [46]. Эта база содержит около 5 млн записей о соединениях. Каждая запись в этой базе представляет собой образ сетевого соединения. Соединение — это последовательность TCP пакетов за некоторое конечное время, моменты начала и завершения которого четко определены и в течение которого данные передаются от IP-адреса источника на IP-адрес приемника (и в обратном направлении), используя некоторый определенный протокол.

Каждое сетевое соединение характеризуется отдельной записью и состоит из около 100 байтов, включающей 41 параметр сетевого трафика, и промаркирована как «атака» или «не атака». Например, первый параметр определяет длительность соединения, второй указывает используемый протокол, третий — целевую службу и т. д.

В базе KDD-99 представлены 22 типа атаки. При этом атаки делятся на четыре основные категории:

DoS — отказ в обслуживании, характеризуется генерацией большого объема трафика, что приводит к перегрузке и блокированию сервера;

U2R предполагает получение зарегистрированным пользователем привилегий локального суперпользователя (администратора);

R2L характеризуется получением доступа незарегистрированного пользователя к компьютеру со стороны удаленной машины;

Probe заключается в сканировании портов с целью получения конфиденциальной информации.

В качестве выходных данных используется 5-мерный вектор, где 5 — количество классов атак плюс нормальное состояние.

В качестве второго источника анализируемых последовательностей можно взять наборы данных, предоставленные Линкольнской лабораторией Масачусетского технологического института (1999 DARPA Intrusion Detection Evaluation) [30].

4.3.2. Тестирование программного обеспечения

Для тестирования разработанного программного обеспечения рассмотрим образцы трассировок, содержащих сетевые атаки различных типов.

Сетевая атака Portsweep — наблюдатель изучает множество портов для определения, какие сервисы поддерживаются на единичном хосте.

Как видно из временно́й шкалы (горизонтальная) (рис. 4.7 и 4.8), атака длится менее 1 секунды, таким образом, сетевой тра-

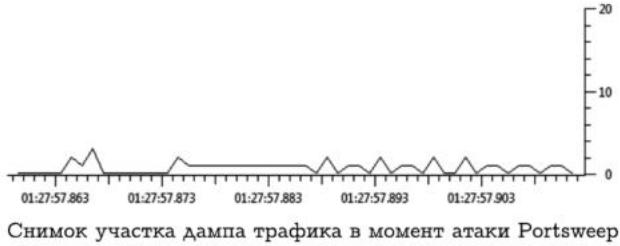

Рис. 4.7. Снимок участка дампа трафика в момент атаки Portsweep утилитой IO Graph

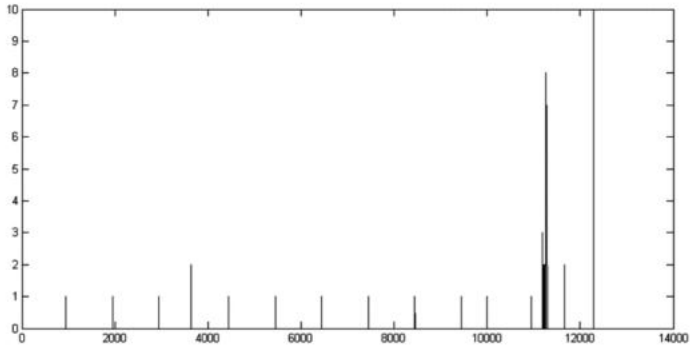

Рис. 4.8. Трассировка сетевого трафика продолжительностью около 10 с, содержащая атаку Portsweep. Информация из дампа трафика была отформатирована и перенесена в Matlab

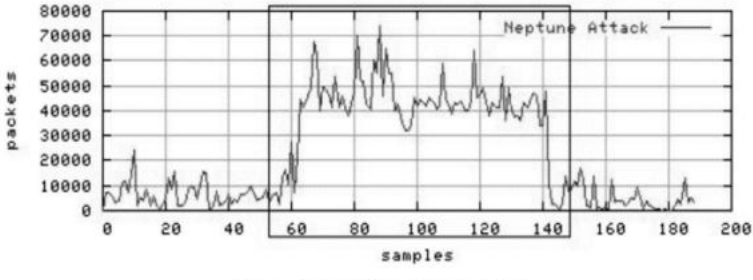

Рис. 4.9. Участок с атакой

фик для рассмотрения данной атаки берется с большим масштабом (шаг — 0,01 секунды).

Сетевая атака Neptune (рис. 4.9) — пример DoS атаки, которая выполняет рассылку флуда (SYN пакетами) на один или множественные порты. Нормальное установление TCP-соединения происходит в соответствии со схемой «трехстороннего рукопожатия»: клиент отправляет SYN, сервер отвечает SYN-ACK, клиент отвечает ACK. Если со стороны клиента отправлять большое количество SYN-пакетов, то сервер будет вынужден каждый раз инициировать соединение, которое не будет установлено. Количество таких соединений ограничено и когда оно достигнет предела, сервер перестанет отвечать на какие-либо запросы.

Как видно из рис. 4.10 и 4.11 в момент атаки средний уровень трафика возрастает, при этом данная атака является непродолжительной — около 1 часа, таким образом масштаб, с которым берется трафик, можно понизить (шаг 0,1 секунды).

Сетевая атака Nmap (рис. 4.12) — составление карты сети с

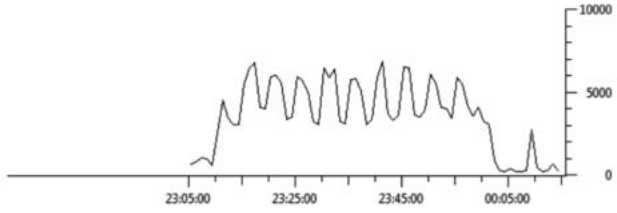

Рис. 4.10. Снимок участка дампа трафика в момент атаки Neptune утилитой IO Graph

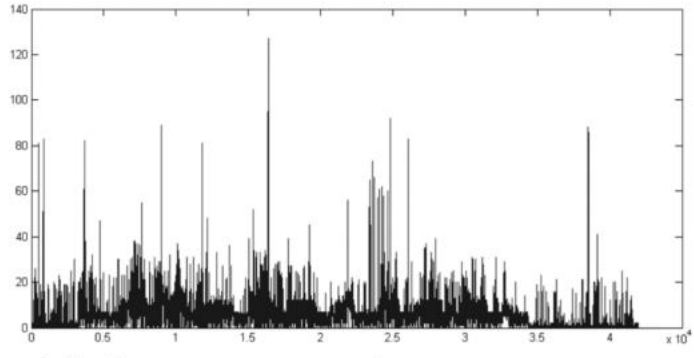

Рис. 4.11. Трассировка сетевого трафика продолжительностью около 1ч, содержащая атаку Neptune. Информация из дампа трафика была отформатирована и перенесена в Matlab

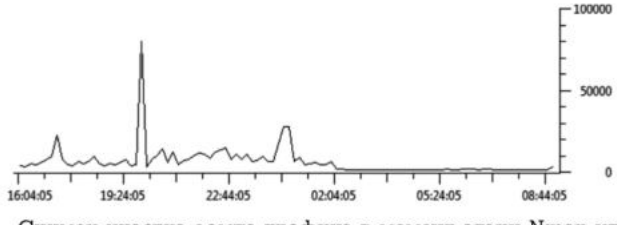

Рис. 4.12. Снимок участка дампа трафика в момент атаки Nmap утилитой IO Graph

использованием утилиты nmap. Способы исследования сети могут меняться — возможно использование SYN пакетов.

Особенностью представленной атаки является продолжительность — более 12 ч (рис. 4.13). В связи с наличием фонового трафика, который скрывает саму атаку, масштаб был взят наиболее большим, однако при данной длительности атаки появляется проблема хранения и обработки большого объема данных (шаг 1 с).

Mailbomb — почтовые бомбы (рис. 4.14) — один из простейших видов сетевых атак. Злоумышленником посылается на компьютер

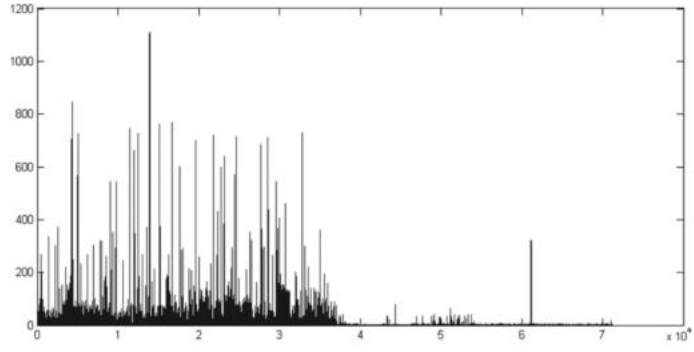

Рис. 4.13. Трассировка сетевого трафика продолжительностью около 12 ч, содержащая атаку Nmap

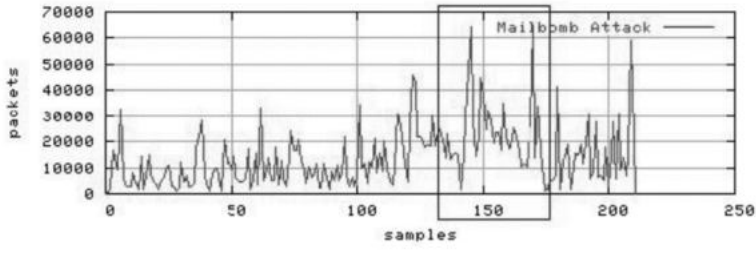

Рис. 4.14. Участок с атакой

пользователя или почтовый сервер компании одно огромное сообщение или множество (десятки тысяч) почтовых сообщений на SMTP-порт mail-сервера жертвы, что приводит к выводу системы из строя, если не установлено специальных блокировок.

Smurf. Атака SMURF (рис. 4.15) относится экспертами к наиболее опасной разновидности атаки DDoS, поскольку имеет эффект усиления, являющийся результатом отправки прямых широковещательных запросов ping к системам, которые обязаны послать ответ. Запрос направляется либо на сетевой адрес, либо по адресу

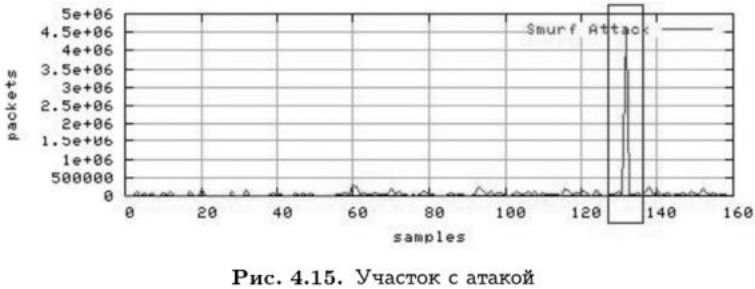

Рис. 4.15. Участок с атакой

широковещательной рассылки, но в любом случае устройство должно выполнить преобразование уровня 3 (IP) к уровню 2 (сетевой), как этого требует RFC 1812 Requirements for IP Version 4 Routers (Требования к маршрутизаторам протокола IP версии 4). В стандартной сети класса C (24-разрядное выделение адресов) сетевым адресом будет .0, а адресом широковещательной рассылки — .255. Прямая широковещательная рассылка обычно служит для диагностики, позволяя выявить работающие системы без запроса по ping каждого адреса из диапазона. Атака smurf пользуется особенностями прямой широковещательной рассылки и требует как минимум трех участников: атакующий, усиливающая сеть и жертва. Атакующий посылает мистифицированный пакет ICMP ECHO по адресу широковещательной рассылки усиливающей сети. Адрес источника этого пакета заменяется адресом жертвы, чтобы представить дело так, будто именно целевая система инициировала запрос. После этого происходит следующее: поскольку пакет ECHO послан по широковещательному адресу, все системы усиливающей сети возвращают жертве свои ответы (если только конфигурация не определяет другого поведения). Использует диагностический протокол ICMP. Послав один пакет ICMP в сеть из 100 систем, атакующий инициирует усиление атаки DDoS в сто раз! Коэффициент усиления зависит от состава сети, поэтому атакующий ищет большую сеть, способную полностью подавить работу системы-жертвы. Получая запросы, все компьютеры подсети посылают эхо-ответы на адрес жертвы, от чего она быстро прекращает работу, не справляясь с большим количеством информации.

Ipsweep (рис. 4.16). Существует два типа сканирования портов: вертикальное, когда один хост сканируется по всем открытым портам, и горизонтальное, когда группа хостов сканируется на открытый определенный порт. В большинстве случаев на порт посылается SYN-пакет, и если порт открыт, он ответит SYN-ACK. Быстрые способы сканирования также описаны в [24].

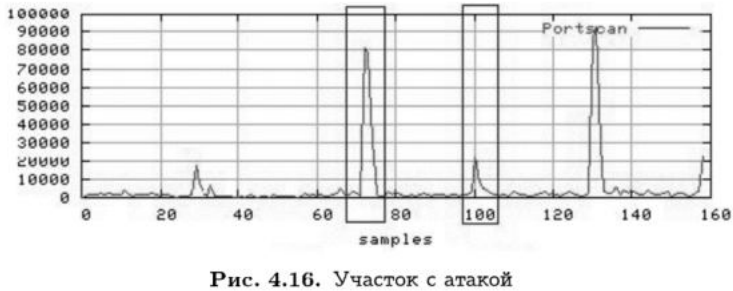

Рис. 4.16. Участок с атакой

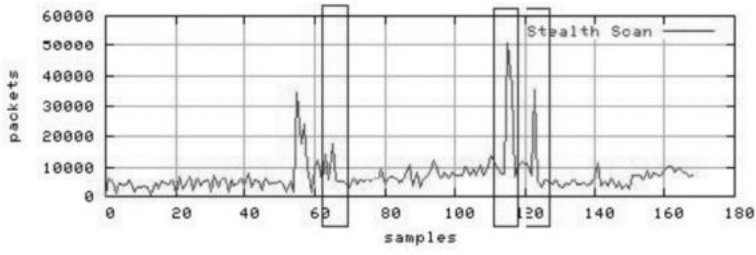

Рис. 4.17. Участок с атакой

Stealth scan (рис. 4.17). Скрытое сканирование реализуется отсылкой пакета FIN на порт. В соответствии со стандартом RFC 793 [28 Postel, Transmission Control Protocol, RFC 793, Sep. 1981] корректный ответ закрытого порта будет ответный пакет RST, тогда как открытый порт просто отбросит пакет и ничего не пошлет в ответ. Такое сканирование заметить сложнее.

5 АНАЛИЗ МЕТОДОВ ОБНАРУЖЕНИЯ АНОМАЛИЙ

Методы обнаружения аномалий направлены на выявление неизвестных атак и вторжений. Для защищаемой системы СОВ на основе совокупности параметров оценки формируется «образ» нормального функционирования. В современных СОВ выделяют несколько способов построения «образа»:

- накопление наиболее характерной статистической информации для каждого параметра оценки;
- обучение нейронных сетей значениями параметров оценки;
- событийное представление.

Легко заметить, что в обнаружении очень значительную роль играет множество параметров оценки. Поэтому в обнаружении аномалий одной из главных задач является выбор оптимального множества параметров оценки.

Другой, не менее важной задачей является определение общего показателя аномальности. Сложность заключается в том, что эта величина должна характеризовать общее состояние «аномальности> в защищаемой системе.

5.1. Статистические методы обнаружения аномального поведения

Основной недостаток сигнатурных методов обнаружения сетевых атак, связанный с неспособностью системы обнаруживать атаки неизвестных типов, может быть устранен применением методов, основанных на выявления аномалий сетевой активности. Такие методы основаны на предположении, что для вычислительной системы существует свой профиль нормального состояния и любые значительные отклонения от него являются вероятным кандидатом на возможную атаку. Основное достоинство такого метода – возможность выявления новых, неизвестных ранее видов атак. Для построения базового профиля системы используется набор данных, свободный от аномалий, или статистические методы.

Как класс статистический анализ относится к поведенческим методам определения нарушений в сети и основан на сопоставлении текущего состояния сетевой инфраструктуры с некими определенными заранее признаками, характеризующими штатное функционирование сетевой инфраструктуры. Методы статистического анализа имеют различные интерпретации, основанные на различных динамических характеристиках сетевого трафика, однако базовые принципы практически у всех идентичны. Неоспоримым преимуществом применения методов статистического анализа является возможность определения впервые реализуемых методов негативного воздействия на объект атаки со стороны злоумышленника. Однако для его успешной реализации необходимо определить объект анализа, иметь определенные структурированные характеристики, образующие корректную конфигурацию, и критерии, по которым можно определить потенциальную угрозу сетевой безопасности.

Применение методов статистического анализа является наиболее распространенным видом реализации технологии обнаружения аномального поведения. Статистические датчики собирают различную информацию о типичном поведении объекта и формируют ее в виде профиля. Профиль в данном случае — это набор параметров, характеризующих типичное поведение объекта. Существует период начального формирования профиля. Профиль формируется на основе статистики объекта, и для этого могут применяться стандартные методы математической статистики, например метод скользящих окон и метод взвешенных сумм. Статистические методы универсальны, поскольку для проведения анализа не требуется знания о возможных атаках и используемых ими уязвимостях, и основаны на изменениях некоторых статистических характеристик потока пакетов. Для применения статистических методов анализа TCP/IP трафика необходимо выделить основные показатели, характеризующие штатное функционирование сетевой инфраструктуры, и осуществлять динамический контроль над их состоянием. В качестве таких показателей должна использоваться информация, по которой можно проанализировать историю сетевого взаимодействия. К данным, которые могут быть проанализированы, например, при захвате трафика TCP/IP, относятся поля заголовков протоколов IP, TCP, UDP, ICMP и содержимое полей данных.

После того как профиль сформирован, действия объекта сравниваются с соответствующими параметрами и при обнаружении существенных отклонений подается сигнал о начале атаки.

В контрольной аппаратуре механизмы наблюдения размещают-

ся для наблюдений свойств или поведения системы и выдачи тревоги, если важный параметр изменяется в диапазоне звуковых операций. Обнаружение таких событий часто ассоциируется с обнаружением существенных изменений по сравнению с нормальным состоянием системы. Нормальное состояние может определяться спецификацией или измерениями в самом начале и правильной калибровкой системы. Возможных причин изменения множество, например неисправности, сбои и износ. Независимо от причин, часто важно быстро устранить изменение, например с помощью ремонта или перекалибровки системы, с тем, чтобы избежать будущих повреждений.

В контексте анализа трафика мы интересуемся методами фиксирования изменений для обнаружения аномалий трафика. Предполагается, что причинами аномалий трафика является существенное изменение некоторых характеристик трафика. Однако качество результатов обнаружения зависит не только от выбранного метода обнаружения изменений. Еще более важным является выбор показателей рассматриваемого трафика, которые наиболее чувствительны к событиям, имеющим отношение к операции и администрированию сети, такие, как сетевые сбои, атаки вредного трафика. С другой стороны, показатели должны быть достаточно чувствительны к изменению трафика и неисправностям, вызванных законным и безвредным трафиком. В противном случае мы рискуем получить большое число ложных и неинтересных тревог.

Особенностью нахождения изменений являются серии наблюдений, а не конкретное значение. В пределах такой серии изменения ищутся в момент времени, в который статистические свойства наблюдаемой величины резко меняются. «Резко» означает, что изменение происходит мгновенно или, по крайней мере, очень быстро за период наблюдения. До и после изменений статистические свойства либо не меняются, либо меняются незначительно. При таких условиях даже небольшие и устойчивые изменения могут быть обнаружены, но с более длительным временем обнаружения задержки, чем при большом изменении. Причина — большее количество наблюдений, собранных после изменения.

Со статистической точки зрения обнаружение изменений опирается на тестовые гипотезы с нулевой гипотезой H_0, утверждающей что нет изменений, и альтернативной гипотезой H_1, утверждающей обратное. Разработка таких гипотез требует априорного знания распределения вероятностей до изменения. Кроме того, распределение может быть оценено по предыдущим наблюдениям, которые должны быть без аномалий. В этом случае ожидаемая оценочная ошиб-

ка может учитываться (она может оказать существенное влияние на критическое значение, использующееся как пороговое в статистике), особенно если оценка основана на небольшом количестве наблюдений.

Исходя из статистики, выбирают, насколько H_0 должна быть отклонена от данного образца на одно или большее число наблюдений и значение уровня. Значение уровня определяет максимально допустимую вероятность отклонения H_0, хотя гипотеза верна при появлении ошибки первого рода, или ложной тревоги. С другой стороны, если H_0 не отклоняется при изменении свойств, мы говорим про ошибку второго рода. При обнаружении изменений тревога становится ложной, если статистические свойства рассматриваемой величины существенно не изменились. Однако статистические различия тревоги и ложной тревоги могут быть связаны с классификацией уместных и неуместных тревог с точки зрения аналитика.

Методы обнаружения изменений могут классифицироваться в соответствии со следующими критериями.

Онлайн — офлайн: регистрация онлайн изменений повторяет тесты с поступающими новыми наблюдениями. Цель состоит в том, чтобы обнаружить изменения с низкой задержкой и длительным временем между ложными тревогами. Офлайн регистрация изменений анализирует серию наблюдений фиксированной длины. Цель состоит в том, чтобы определить, существует ли изменение в любой момент времени. Если да, то время или величина изменения может быть оценена.

Байесовский — небайесовский: байесовская регистрация изменений использует априорную информацию о распределении времени изменения для того, чтобы улучшить качество обнаружения. Небайесовская регистрация изменений не рассматривает такую информацию, т. е. вероятность изменения предполагается независимой от времени.

Параметрический — непараметрический: в случае параметрического метода вероятность распределения наблюдаемых переменных должна следовать параметрической модели. Кроме того, изменение предполагает действие с параметром, а не с самой параметрической моделью. С другой стороны, непараметрическая регистрация изменений не делает никаких предположений о распределении наблюдаемой величины до и после изменения.

Следовательно, вместо рассмотрения параметров модели непараметрический метод рассматривает статистические свойства наб-

людений, такие, как среднее, дисперсия, корреляция и т. д., которые предположительно будут затронуты изменением.

Известное — неизвестное изменение: если модель системы и параметры после изменения априорно известны, процедура испытаний может базироваться на максимально вероятном решении, которое определяет образцы вероятностей для H_0 и H_1 и выбирает гипотезы, которые наиболее вероятны. Если величина изменения априорно не известна, H_0 может быть только отклонена по отношению к данному уровню значимости.

Одномерный — многомерный: одномерная регистрация изменений принимает решение по одной переменной. Многомерная регистрация изменений рассматривает несколько переменных и корреляции между ними, например используя статистику T2.

В общем случае большинство априорных знаний доступно, что облегчает регистрацию изменений с высокой точностью. К примеру, параметрический метод более сильный, чем непараметрический, что позволяет обнаруживать настоящие аномалии среди некоторого уровня ложных тревог (т. е. вероятность тревоги в отсутствие каких-либо значительных изменений). Однако, если предложенная модель некорректна, параметрический метод теряет свою силу и может привести к неправильным решениям. В случае обнаружения аномального трафика мы обычно не знаем конкретного распределения наблюдаемых переменных, таким образом, метод параметрической регистрации изменений должен быть применен осторожно, зная то, что таблицы параметризации обычно действительны для нормально распределенных переменных. В общем случае непараметрический метод регистрации изменений более подходящий. Кроме того, изменения должны быть зарегистрированы очень быстро (т. е. онлайн) без каких-либо априорных знаний об их величине, так как такая информация обычно недоступна. Кроме того, мы ограничимся небайесовскими методами и не будем считать, что изменения происходят чаще в определенные моменты времени.

5.2. Ошибки первого и второго рода. ROC кривые

Для оценки эффективности алгоритмов обнаружения, как правило, вводится понятие ошибок первого и второго рода.

Пусть дана выборка $X = (X_1, ..., X_n)$ из неизвестного совместного распределения P^X и поставлена бинарная задача проверки статистических гипотез H_0 и H_1, где H_0 — нулевая гипотеза, а H_1 — альтернативная гипотеза. Предположим, что задан статистический

Таблица 5.1

Гипотезы и критерии

		Верная гипотеза	
		H_0	H_1
Результат применения критерия	H_0	H_0 верно принята	H_0 неверно принята (ошибка *второго* рода)
	H_1	H_0 неверно отвергнута (ошибка *первого* рода)	H_0 верно отвергнута

критерий $f: R^n \to \{H_0, H_1\}$, сопоставляющий каждой реализации выборки $X = x$ одну из имеющихся гипотез. Тогда возможны следующие четыре ситуации:

1) распределение P^X выборки X соответствует гипотезе H_0, и она точно определена статистическим критерием, т. е. $f(x) = H_0$;

2) распределение P^X выборки X соответствует гипотезе H_0, но она неверно отвергнута статистическим критерием, т. е. $f(x) = H_1$;

3) распределение P^X выборки X соответствует гипотезе H_1, и она точно определена статистическим критерием, т. е. $f(x) = H_1$;

4) распределение P^X выборки X соответствует гипотезе H_1, но она неверно отвергнута статистическим критерием, т. е. $f(x) = H_0$.

Во втором и четвертом случае говорят, что произошла статистическая ошибка, и ее называют *ошибкой первого и второго рода* соответственно (табл. 5.1).

Как видно из вышеприведённого определения, ошибки первого и второго рода являются взаимно симметричными, т. е. если поменять местами гипотезы H_0 и H_1, то *ошибки первого рода* превратятся в *ошибки второго рода* и наоборот. Тем не менее в большинстве практических ситуаций путаницы не происходит, поскольку принято считать, что *нулевая гипотеза* H_0 соответствует состоянию «по умолчанию» (естественному, наиболее ожидаемому положению вещей), например обследуемый человек здоров или проходящий через рамку металлодетектора пассажир не имеет запрещённых металлических предметов. Соответственно, *альтернативная гипотеза* H_1 обозначает противоположную ситуацию, которая обычно трактуется как менее вероятная, неординарная, требующая какой-либо реакции.

С учётом этого *ошибку первого рода* часто называют ложной тревогой, ложным срабатыванием или ложно положительным срабатыванием, например анализ крови показал наличие заболевания, хотя на самом деле человек здоров, или металлодетектор выдал сигнал тревоги, сработав на металлическую пряжку ремня. Слово «по-

ложительный» в данном случае не имеет отношения к желательности или нежелательности самого события.

Соответственно, *ошибку второго рода* иногда называют пропуском события или ложноотрицательным срабатыванием.

Вероятность ошибки первого рода при проверке статистических гипотез называют уровнем значимости и обычно обозначают греческой буквой α (отсюда название α-errors).

Вероятность ошибки второго рода не имеет какого-то особого общепринятого названия, на письме обозначается греческой буквой β (отсюда β-errors). Однако с этой величиной тесно связана другая, имеющая большое статистическое значение — мощность критерия. Она вычисляется по формуле $(1 - \beta)$. Таким образом, чем выше мощность, тем меньше вероятность совершить ошибку второго рода.

Обе эти характеристики обычно вычисляются с помощью так называемой функции мощности критерия. В частности, вероятность ошибки первого рода есть функция мощности, вычисленная при нулевой гипотезе. Для критериев, основанных на выборке фиксированного объема, вероятность ошибки второго рода есть единица минус функция мощности, вычисленная в предположении, что распределение наблюдений соответствует альтернативной гипотезе. Для последовательных критериев это также верно, если критерий останавливается с вероятностью единица (при данном распределении из альтернативы).

В статистических тестах обычно приходится идти на компромисс между приемлемым уровнем *ошибок первого и второго рода*. Зачастую для принятия решения используется пороговое значение, которое может варьироваться, с целью сделать тест более строгим или, наоборот, более мягким. Этим пороговым значением является уровень значимости, которым задаются при проверке статистических гипотез.

Наличие уязвимостей в вычислительных системах приводит к тому, что приходится, с одной стороны, решать задачу сохранения целостности компьютерных данных, а с другой стороны — обеспечивать нормальный доступ легальных пользователей к этим данным. В данном контексте возможны следующие нежелательные ситуации:

- когда *авторизованные пользователи* классифицируются как *нарушители* (ошибки первого рода);
- когда *нарушители* классифицируются как *авторизованные пользователи* (ошибки второго рода).

ROC кривые. В качестве алгоритма оценки эффективности СОВ часто используются ROC кривые. Данный термин возник в те-

ории сигналов и является графической интерпретацией зависимости между чувствительностью (или относительным числом правильных срабатываний, TPR) исследуемой системы и ее тревожностью (или относительным числом ложных срабатываний, FPR) при изменении пороговых параметров анализируемого метода. Формулы для построения ROC кривых представлены ниже:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}},$$

где TP — число выявленных аномалий; FN — число пропущенных аномалий; FP — число ложных срабатываний; TN — число верно отмеченных событий как нормальных.

5.3. Критерии соответствия и однородности

Контрольные графики являются не только средством для обнаружения изменений в ряде наблюдений. Обычные критерии соответствия и критерии двойной выборки однородности могут хорошо использоваться. Критерии соответствия оценивают, соответствует ли образец заданной плотности распределения, в то время как критерии однородности проверяют, соответствуют ли два образца этим же распределениям.

Для того чтобы применить подобные критерии для серии, наблюдения должны быть сгруппированы в выборках. С этой целью определим скользящее окно n_1 последних наблюдений $x_{t-n_1}+1, ..., x_t$, которые описывают поведение текущего трафика. Этот образец сравнивается с заданным распределением (соответствие) или другим образцом наблюдений (однородность), который, как предполагается, представляет нормальный трафик. В случае двух образцов можем увидеть второе скользящее окно n_2 прошлых наблюдений $x_{t-n_1-n_2}+1, ..., x_{t-n_1}$.

Критерий хи-квадрат. Для оценки случайности или существенности расхождений между частотами эмпирического и теоретического распределений одной выборки используется ряд показателей, именуемых критериями согласия. Одним из основных и наиболее распространенных показателей является критерий Q, предложенный К. Пирсоном:

$$Q = \sum_{i=1}^{m} \frac{(N_i - E_i)^2}{E_i},$$

где m — число категорий; N_i — посчитанное число наблюдений в категории i; E_i — ожидаемое число наблюдений в категории i.

Все хи-квадрат критерии полагаются на вычисление статистики хи-квадрат Q, которая имеет хи-квадрат распределение, если наблюдения в выборке (выборках) независимы и достаточно многочисленны.

К. Пирсоном найдено распределение величины Q и составлены таблицы, позволяющие определить предельное верхнее значение при заданном уровне значимости и числе степеней свободы, которое в общем случае равно количеству наблюдений за вычетом числа ограничений, необходимых для расчета статистической характеристики. Если фактическое значение Q меньше табличного, то расхождения между эмпирическими и теоретическими частотами считают случайными, а гипотезу о принятом законе распределения принимают.

В критерии сравнения двух образцов на однородность хи-квадрат статистика

$$Q = \sum_{j=1}^{2} \sum_{i=1}^{m} \frac{(N_{j,i} - E_{j,i})^2}{E_{j,i}}; \quad E_{j,i} = \frac{N_{1,i} + N_{2,i}}{n_1 + n_2} n_j, \quad j \in \{1, 2\}.$$

Здесь $N_{1,i}$ и $N_{2,i}$ — число наблюдений в категории i для двух образцов; $E_{j,i}$ — ожидаемое число наблюдений в выборке j, принадлежащей категории i, предполагая однородность наблюдений в обоих образцах. В случае независимых и нормально распределенных наблюдений Q примерно хи-квадрат распределено с $(m-1)$ степенью свободы, если $E_{j,i}$ превышает 5 для всех образцов и категорий.

Критерий Колмогорова–Смирнова. Одновыборочный критерий нормальности Колмогорова–Смирнова основан на максимуме разности между кумулятивным распределением выборки и предполагаемым кумулятивным распределением. Если статистика Колмогорова–Смирнова D значима, то гипотеза о том, что соответствующее распределение нормально, должна быть отвергнута. Выводимые значения вероятности допустимы, если среднее и стандартное отклонение нормального распределения известны априори и не оцениваются из данных. Однако обычно эти параметры вычисляются непосредственно из данных. В этом случае критерий нормальности включает сложную гипотезу («насколько вероятно получить статистику D данной или большей значимости, зависящей от среднего и стандартного отклонения, вычисленных из данных»). Для критериев Колмогорова и Смирнова выбор меры расхождения связан с эмпирической функцией распределения $F_n^*(t)$. А именно, рассмат-

ривается статистика Колмогорова

$$D_n = \sup_{t \in R} |F_n^*(t) - F_0(t)|$$

и статистика Смирнова

$$w_n^2 = \int_{-\infty}^{+\infty} (F_n^*(t) - F_0(t))^2 \, dF_0(t)$$

соответственно. Примечательно то, что эти функции легко могут быть вычислены по выборке (не требуется брать какие-либо интегралы, все сводится к простым выражениям, содержащим конечное суммирование и взятие максимума).

Теоремы Колмогорова и Смирнова являются основой для построения соответствующих критериев согласия с критическими множествами вида $S = \{\sqrt{n} D_n > C_1\}$ и $S = \{n w_n^2 > C_2\}$ соответственно. Числа C_1, C_2 определяются по заданным уровням значимости из таблиц допредельных (или предельных, если n очень велико) распределений Колмогорова и Смирнова. Таким образом критерий Колмогорова–Смирнова предполагает определение эмпирических накопленных частостей (долей) и сравнение их с теоретическими частостями и используется в случаях, когда исходные данные упорядочены. Точка, в которой два распределения будут иметь максимальное расхождение (по модулю), может быть использована в качестве расчетного критерия, обозначаемого через D_n и определяемого по формуле

$$D_n = \left| \sum f_i - \sum f_i' \right|,$$

где $\sum f_i$ — накопленные частости (доли) эмпирического распределения; $\sum f_i'$ — накопленные частости теоретического распределения.

Величина D_n, рассчитанная по данным выборки, сравнивается с критическим значением $D_{\text{крит}}$:

$$D_{\text{крит}} = \lambda / \sqrt{n},$$

где λ — критерий Колмогорова–Смирнова, соответствующий заданному уровню значимости α; n – размер выборки.

Различным значениям λ соответствуют различные значения вероятностей. Эти показатели табулированы. Так, например, при уровне значимости $\alpha = 0{,}05$ значение λ для большой выборки равно 1,36. Как и для показателя Q, считается вполне допустимым рассматривать расхождения между эмпирическими и теоретическими частотами случайными, если фактическое значение D_n меньше критического значения $D_{\text{крит}}$.

Критерий оценки Вилкоксона–Манна–Уитни работает только как критерий сравнения двух образцов на однородность. Для этого сортируют все наблюдения, чтобы определить их ранг. Если в наблюдениях попадаются одинаковые значения (так называемые связи), то выделяется средний ранг для затронутых наблюдения.

Если оба образца в результате из некоторого распределения, ранг распределения напоминает случайно полученные значения из $\{1, 2, ..., (n_1 + n_2)\}$. При таком условии сумма рангов наблюдений S_j образца $j \in \{1, 2\}$ имеет следующие значения математического ожидания и дисперсии:

$$E[S_j] = \frac{n_i(n_j + n_2 + 1)}{2}; \quad \text{Var}[S_j] = \frac{n_1 n_2(n_1 + n_2 + 1)}{12}.$$

Когда размеры выборки n_1 и n_2 достаточно велики ($\geqslant 8$), распределения S_1 и S_2 примерно нормальны; если $n_1 + n_2 > 60$, приблизительно нормальны. Критерий оценки Вилкоксона–Манна–Уитни отвергает гипотезу однородности, если разность между вычисленной суммой рангов и средним значением значительна. Учитывая оценку, а не абсолютные значения, критерий Вилкоксона–Манна–Уитни надежен относительно небольшого числа выбросов в наблюдениях. Отличительная сила критерия возрастет, если образцы следуют из распределения с различными средними или ассиметричны. С другой стороны, критерий не так хорош, если распределения отличаются только дисперсией.

Увеличение размера выборки последних наблюдений n_1 позволяет обнаружить небольшие изменения, если они сохраняются в течении длительного времени, но также увеличивает задержку определения. Для критерия однородности двух образцов нам также необходимо указать размер второго образца n_2, который будет представителем для состояния входного контроля. Важнейшее значение для всех этих критериев имеют условие независимости и одинаковое распределение наблюдений до изменения.

5.4. Параметрический метод регистрации изменений

Методы обнаружения изменений должны анализировать трафик во времени и искать те моменты времени, когда статистические свойства резко меняются, т. е. мгновенно или по крайней мере очень быстро с момента начала изменений. Предполагается, что до и после изменения мониторинг либо ничего не показывает, либо показывает незначительные изменения. В этих условиях даже небольшие из-

менения могут быть обнаружены с высокой вероятностью, если они сохраняются в течении длительного срока.

В целом, чем больше входных данных доступно анализу, тем проще отследить изменения. К примеру, параметрические методы имеют больше возможностей, чем непараметрические, что означает, что они позволяют обнаружить «истинные аномалии», в то же время сохраняя уровень ложных тревог на должном уровне (т. е. вероятность тревоги в отсутствие каких-либо изменений в трафике). Однако, если модель выбрана неверно, то параметрические методы теряют свои главные возможности, что может привести к неправильным решениям.

В случае обнаружения аномалии в трафике нельзя определить конкретное распределение характеристик трафика, таким образом, метод обнаружения должен быть непараметрическим или, по крайней мере, устойчивым к аномалиям. Более того, изменения должны быть обнаружены как можно быстрее (например в реальном времени), не требуя никакой дополнительной информации, которая, как правило, и не существует.

5.4.1. Контрольные карты

Практическое решение для статистического обнаружения изменений дает метод контрольных карт. В контрольных картах среднее значение и изменение переменной характеризуется центральным (CL), верхним (UCL) и нижним (LCL) контрольными пределами. Изменение обнаруживается, если значение превышает один из контрольных пределов. Решение может быть сделано с помощью сравнения гипотез — гипотезы H_0, когда нет изменений, и альтернативной гипотезы H_1, когда изменения есть.

Рассмотрим возможности применения контрольных карт в проблеме обнаружения аномалий в трафике. С этой целью будем анализировать временные ряды байтов, пакетов и количество потоков, которые можно с легкостью получить с маршрутизаторов с помощью SNMP, IPFIX или CiscoNetFlow. Эти величины подвергаются систематическим изменениям, в частности сезонным вариациям. Кроме того, может существовать такая зависимость между последующими измерениями, как сериальная корреляция. Систематические изменения и сериальная корреляция должны быть учтены, так как большинство контрольных карт предназначены для независимых наблюдений и наблюдений с одинаковым распределением.

Полезным инструментом являются методы прогнозирования, которые предсказывают будущие изменения на основе прошлых наблюдений.

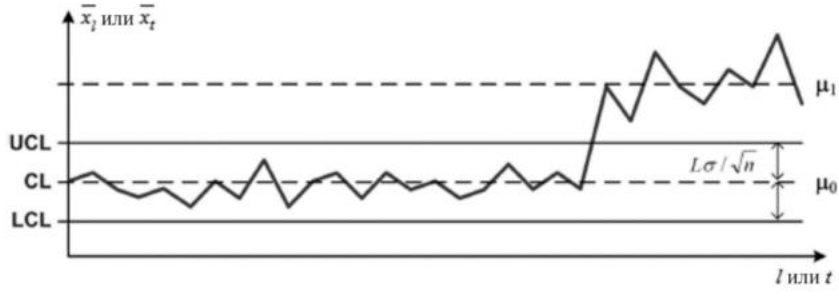

Рис. 5.1. Пример контрольной карты

Типичные контрольные карты содержат центральную линию (CL), представляющую среднее значение наблюдаемой величины Y в нормальных условиях. Выше и ниже центральной линии находятся верхний и нижний контрольные пределы (UCL, LCL), определяющие диапазон нормальных колебаний величины, как это показано на рис. 5.1. Функция, принимающая решения об атаке, наблюдает за изменениями переменной y вне этих границ.

Статистические свойства контрольных карт могут быть выведены из теории последовательного отношения вероятностей тестов (sequential probability ratio tests, SPRT). Если данные до и после изменения известны, то решающая функция может быть преобразована в логарифмическое отношение:

$$s(y) = \log \frac{p_{\Theta_1}(y)}{p_{\Theta_0}(y)}, \tag{5.1}$$

где $p_\Theta(y)$ — функция плотности вероятности Y с параметром Θ; Θ_0 и Θ_1 – параметры до и после изменения.

Если отношение $s(y)$ положительно, наблюдаемая случайная величина чаще соответствует распределению до изменения, нежели после. Таким образом, можно определить порог h для $s(y)$, отклоняющий нулевую гипотезу H: $\Theta = \Theta_0$ и подтверждающий альтернативную гипотезу H: $\Theta = \Theta_1$ при заданном уровне значимости. Уровень значимости соответствует уровню ложной тревоги.

Если Y имеет нормальное распределение с постоянной дисперсией σ^2 и средними значениями μ_0 и μ_1 до и после изменения, то $s(y)$ принимает вид

$$s(y) = \frac{\mu_1 - \mu_0}{\sigma^2}\left(y - \frac{\mu_1 + \mu_0}{2}\right). \tag{5.2}$$

Если $\mu_0 < \mu_1$, то $s(y) > h$ эквивалентно функции

$$y > \mu_0 + L\sigma, \tag{5.3}$$

где

$$L = \frac{h\sigma}{\mu_1 - \mu_0} + \frac{\mu_1 - \mu_0}{2\sigma}. \tag{5.4}$$

В этом уравнении, очевидным является признак контрольных карт: μ_0 — центральная линия и $\mu_1 + L\sigma$ — верхний предел.

Так как дисперсия одного наблюдений достаточно высока, методы обнаружения изменений обычно применяются к последовательности $\{y_t \mid t = a, ..., b\}$ для увеличения мощности гипотезы теста. При условии, что наблюдения являются независимыми, отношение правдоподобия последовательности будет иметь вид:

$$s(y_a, ..., y_b) = \log \frac{\prod_{t=a}^{b} p_{\Theta_1}(y_t)}{\prod_{t=a}^{b} p_{\Theta_0}(y_t)} = \sum_{t=a}^{b} s(y_t). \tag{5.5}$$

Предположения теста основано на том, что $s(y_a, ..., y_b)$ соответствует контрольной карте тестовой статистики, которая вычисляется из $y_a, ..., y_b$.

5.4.2. Контрольные карты Шухарта

Контрольные карты Шухарта определяют уровни UCL, CL и LCL для статистики, вычисленной из N наблюдений $y_{(t-1)N+1}, ..., y_{tN}$. Например, статистика имеет среднее значение $\overline{y}_l$, которое подходит для определения изменений, в виде

$$\overline{y}_l = \frac{1}{N} \sum_{t=(l-1)N+1}^{lN} y_t, \quad l = 1, 2, ... \tag{5.6}$$

Если наблюдения независимы и имеют одинаковое распределение со средним значением μ_0 и дисперсией σ^2, $\overline{y}_l$ является оценкой μ_0 с дисперсией μ_0/N. Следовательно, верхний UCL и нижний LCL пределы могут быть определенны в форме $\mu_0 \pm L_\sigma/\sqrt{N}$ с настраиваемым параметром L.

Тревога возникает, если $\overline{y}_l$ пересекает один из пределов. Для больших значений N в силу центральной предельной теоремы $\overline{y}_l$ имеет нормальное распределение, таким образом, контрольные пределы для данной вероятности ложных тревог α будут $\mu_0 \pm (1-\alpha/2)\sigma/\sqrt{N}$.

Однако это приближение не выполняется при малых значениях N, если наблюдается сериальная корреляция.

Особым случаем будет $N = 1$, так называемые индивидуальные контрольные карты Шухарта. Эти карты сравнивают отдельные наблюдения в отношении контрольных лимитов. Очевидно, что

невозможно применить центральную предельную теорему, поэтому распределение Y должно быть известно, чтобы определить точные границы для заданной вероятности ложной тревоги.

5.4.3. Контрольные карты CUSUM

Контрольные карты CUSUM, также называемые алгоритмом кумулятивных сумм (CUSUM cumulative sum), основаны на том, что $S_1 = s(y_1, ..., y_t)$ имеет отрицательное значение в нормальных условиях и положительное при изменении. Решение в алгоритме кумулятивных сумм принимается сравнением функции g_t с порогом h:

$$g_t = S_t - \min_{1 \leqslant i \leqslant t} S_i = \max(0, s(y_t) + g_{t-1}) = [g_{t-1} + s(y_t)]^+ \geqslant h; \quad g_0 = 0. \tag{5.7}$$

Тревога возникает, если g_t пересекает порог h. Чтобы перезапустить алгоритм, необходимо обнулить g_t.

С точки зрения проверки гипотезы контрольная карта CUSUM неоднократно выполняет отношение правдоподобия, где каждое решение учитывает, столько последовательных наблюдений необходимо для принятия H_0 или H_1. Контрольные карты CUSUM начинает новый заход SPRT, если получилось H_0, и останавливается и объявляет тревогу, если принимается гипотеза H_1. Порог h позволяет достичь компромисса между средним временем обнаружения и частотой ложных тревог. Если распределение Y не известно, логарифмическое отношение правдоподобия $s(y_t)$ нужно заменить на статистическую характеристику $u(y_t)$ с сопоставимыми свойствами: среднее значение $u(y)$ должно быть отрицательным до H_0 и положительным после H_1. Такой метод также называют непараметрическим алгоритмом CUSUM.

Соответствующие статистические характеристики для выявления положительных сдвигов среднего значения имеют вид $u^+(y) =$

Рис. 5.2. Контрольный график CUSUM

$y - (\mu_0 + K)$. Параметр K называется опорным значением. Для выявления отрицательных сдвигов нужна другая статистическая характеристика $u^-(y) = (\mu_0 - K) - y$.

В результате получим две функции:

$$g_t^+ = \left[g_{t-1}^+ + y_t - (\mu_0 + K)\right]^+ \geqslant h;$$
$$g_t^- = \left[g_{t-1}^- + (\mu_0 - K) - y_t\right]^+ \geqslant h.$$

Типичные настройки $K = \sigma/2$ и $h = 4\sigma$ или $h = 5\sigma$, где σ — стандартное отклонение Y_t.

По сравнению с контрольными картами Шухарта CUSUM обнаруживает небольшие, но постоянные изменения с большей вероятностью, ибо они накапливаются со временем.

Обнаружение DDOS-атак с применением алгоритма CUSUM. Поясним используемые далее термины.

Распределенное вторжение — это вторжение, удовлетворяющее соотношению «много нарушителей — одна жертва». Вторжение считается распределенным, если различные его этапы выполняются от имени различных источников в сети.

События, составляющие распределенное вторжение, являются *скоординированными*. *Источники*, от имени которых выполняется распределенное вторжение (атака), *взаимосвязаны*.

Существуют различные подходы к обнаружению распределенных атак отказа в обслуживании. Некоторые из них предполагают функционирование датчиков по периметру сети и централизованную обработку данных, поступающих от них, в других используется установка дополнительных программных средств на промежуточных узлах на пути от нарушителя к жертве.

Все широко распространенные подходы к обнаружению DDoS-атак выполняют мониторинг трафика, направленного из внешней сети к защищаемой станции или сервису, и определяют атаку как заметное отклонение от некоторых наблюдаемых характеристик трафика. Можно выделить три основных группы методов обнаружения DDoS-атак.

Методы первой группы основаны на построении профиля активности удаленных станций по отношению к защищаемой станции (сервису) во время обучения и сравнения характеристик трафика с характеристиками профиля в режиме обнаружения. При обнаружении отклонений от профиля генерируется сигнал тревоги. Во многих разработках и исследованиях основным показателем служит среднее количество пакетов, получаемых защищаемой станцией или отдельными сетевыми сервисами. Для обнаружения аномалий

может применяться статистические критерии (среднеквадратичное отклонение, хи-квадрат, отклонение от стандартного нормального распределения, значительное увеличение энтропии и т. д.), кластеризация и др. Эти методы позволяют обнаружить DoS и DDoS-атаки. Такой механизм обнаружения DoS-атак (но не DDoS-атак) как аномалий уже реализован в модуле обработки статистики СОВ «Авгур».

Во вторую группу был выделен широко распространенный статистический метод CUSUM, основанный на обнаружении «точки перехода» (change-point). В данном методе анализируемый трафик обычно сначала разделяется на основе IP-адреса назначения, порта или протокола. Далее значения определенного наблюдаемого параметра трафика (количество новых станций, обратившихся к сервису, разница в количестве пакетов с установленным флагом SYN и пакетов с установленными флагами SYN-ACK, разница в количестве устанавливаемых и закрываемых соединений и т. п.) преобразуются в элементы некоторой последовательности. При наличии атаки значение текущего элемента последовательности будет существенно отличаться от предыдущих членов последовательности. Данный метод имеет несколько существенных преимуществ по сравнению с другими методами. Во-первых, плюсом метода является высокая скорость работы, что позволяет применять его в режиме реального времени. Во-вторых, метод адаптируется к различным нагрузкам в сети при условии их постоянства. Кроме того, данный алгоритм считается наилучшим среди алгоритмов с определяемым уровнем ложных срабатываний, поскольку параметры алгоритма вычисляются на основе формулы, связывающей пороговое значение алгоритма для обнаружения атаки и время обнаружения.

К третьей группе методов можно отнести спектральный анализ и вейвлет-анализ, которые основаны на анализе спектральных характеристик трафика для обнаружения DDoS-атак.

Для обнаружения DDoS-атак был выбран метод, основанный на обнаружении «точки перехода», в силу его быстродействия и незначительных затрат памяти. Для обнаружения DDoS предлагается отслеживать следующие характеристики трафика:

- количество новых станций, обратившихся к сервису;
- количество различных станций, обратившихся к защищаемому сервису;
- разница в количестве устанавливаемых и закрываемых соединений.

Обнаружение DDoS-атак на основе соответствия между устанавливаемыми и закрываемыми соединениями. Проверка на соответствие между пакетами с установленным флагом SYN и пакетами с установленным FIN (RST) для обнаружения была предложена в [28]. При этом необходимо учитывать возможность закрытия соединения с помощью отправки RST. В общем случае существует два варианта нормального поведения протокола TCP: отправка пакета с установленным флагом FIN и разрыв соединения с помощью RST.

RST может быть получен в двух случаях: при получении сервером пакета на закрытый порт (пассивный) и при явном закрытии соединения в случае ошибки (активный). Определить при анализе собранного трафика, в каком из указанных двух случаев получен RST, нет возможности. Поэтому существуют два взаимоисключающих варианта: все пакеты RST считаются пассивными или все пакеты RST считаются активными. В первом случае увеличивается количество ложных срабатываний. Во втором случае уменьшается чувствительность при обнаружении атак.

В качестве метода обнаружения отклонений пар SYN-FIN (RST) от порогового значения в [49] предлагается использовать статистический метод CUSUM (cumulative sum). CUSUM позволяет итеративно отслеживать изменение заданного параметра, выявляя «точки перехода». При этом сбор параметров трафика выполняется в течение равных промежутков времени. Значение Δ_n определяется как разность между количеством пакетов с флагом SYN и количеством пакетов с флагами FIN или RST за n-й промежуток времени. Δ_n нормализуется значением $R(n)$, которое определяется следующим выражением:

$$R(n) = \alpha R(n-1) + (1-\alpha)\mathrm{REPLY}(n), \tag{5.8}$$

где REPLY — количество FIN и RST пакетов; α — константа, равная 0,01.

Отслеживаемые параметры трафика представляются как последовательности величин вида $\{X_n = \Delta_n / R(n)\}$. Изменение величины X_n во времени может считаться случайным стационарным процессом. Из свойств случайного стационарного процесса следует, что среднее значение случайной величины не изменяется с течением времени и равно ее энтропии.

Энтропия $E(X_n) \approx A$ близка к нулю, так как в отсутствие атаки разность между количеством устанавливаемых и закрываемых соединений небольшая. Необходимым условием применения алгорит-

ма CUSUM является то, что при нормальном функционировании элементы последовательности принимают небольшие отрицательные значения, а при отклонении наблюдаемого параметра от среднего значения элементы последовательности становятся большими положительными величинами. Поэтому изначально выбирается параметр $\beta > c$, а затем формируется последовательность $Z_n = X_n - \beta$. Параметр β должен удовлетворять условию $\beta > c$, тогда его можно представить в виде $\beta = c + a$, где a — очень малое положительное число $(a < 1)$.

При нормальном функционировании системы значения элементов последовательности Z_n будут очень малы $(Z_n = X_n - \beta \approx \approx A - A - a \approx -a)$.

Пусть параметр h представляет собой нижнюю границу отклонения элементов последовательности Z_n от среднего значения при возникновении атаки. В алгоритме обнаружения изменений случайной последовательности предполагается, что h намного больше β и, следовательно, h намного больше с и намного больше a. При возникновении SYN-flood атаки значение Z_n станет равным $Z_n = X_n - \beta \geqslant A + h - A - a$ и превысит 0.

Для проверки наличия атаки на каждом шаге алгоритма также формируется другая последовательность $\{y_n\}$, от которой алгоритм и получил свое название:

$$y_n = (y_{n-1} + Z_n)^+; \quad y_0 = 0, \tag{5.9}$$

где $(X)^+ = \begin{cases} X, & X > 0; \\ 0, & X \leqslant 0. \end{cases}$

Элементы последовательности также могут быть представлены как

$$y_n = S_n - \min_{1 \leqslant k \leqslant n} S_k, \tag{5.10}$$

где $S_k = \sum_{i=1}^{k} Z_n$; $S_0 = 0$. Тогда функция обнаружения атаки определяется как

$$d_N(y_n) = \begin{cases} 0, & y_n \leqslant N; \\ 1, & y_n > N \end{cases} \tag{5.11}$$

где N представляет пороговое значение алгоритма. Таким образом, $d_N(y_n)$ принимает значение 1 при обнаружении атаки и 0 в противном случае.

При обнаружении атаки значения элемента последовательности, на котором обнаружено отклонение, не учитывается при вычислении последующих элементов. Это позволяет предотвратить

значительное увеличение новых членов последовательности за счет атак, обнаруженных в прошлом.

Выбор параметров алгоритма CUSUM. Порог N в алгоритме не имеет интуитивно понятного пользователю смысла, поэтому не может быть задан априорно. В [49, 50] предложен следующий подход для определения его значения. Время между ложными срабатываниями растет экспоненциально с ростом N. Обозначим время обнаружения как $\tau_n = \inf\{n: d_N(\cdot) = 1\}$, а нормализованное время обнаружения атаки как

$$\rho_n = (\tau_n - m)^+ / N, \tag{5.12}$$

где m — время начала атаки. При этом для любого $m > 0$ $\rho_n \to \gamma = 1/(h - |c + a|)$, где $h - |c + a|$ — среднее значение последовательности $\{Z_n\}$ во время начала атаки $n > m$.

Пороговое значение может быть определено по формуле (5.12) при заданных параметрах a и h, выбор которых жестко не регламентирован. Во время атаки значение элемента последовательности резко возрастает, что приводит к значительному отклонению его от среднего значения. Параметр a представляет собой верхнюю границу значений элементов последовательности при нормальном функционировании, таким образом, оно должно быть очень малым по модулю числом ($a < 1$). Параметр h, как было указано выше, есть нижняя граница отклонения элементов последовательности $\{Z_n\}$ от среднего значения при возникновении атаки, тогда он значительно больше c. В этом случае обнаружение нечувствительно к выбору a. В [28, 29] параметр h предлагается выбирать равным $2|a|$, такое условие на выбор h теоретически обосновано. В [50] при использовании CUSUM при мониторинге новых IP-адресов для обнаружения DDoS-атак были выбраны значения $a = 0{,}05$ и $h = 0{,}1$, при тестировании реализации с такими параметрами не было отмечено значительного числа ложных срабатываний.

В случае атаки h значительно превышает c (что верно и для данной реализации), то для отслеживания всех трех параметров a выбрано малым положительным числом, значение h равно $2|a|$. После задания a, h вычисляется значение γ при $c = 0$. Далее определяется значение τ_n. Обычно τ_n немного больше m, например в работе [50] $\tau_n = m + 1$. Откуда вычисляется значение порога N. При выборе параметров β, связанного с a, и N преследуются две основные цели: уменьшение количества ложных срабатываний и минимизация времени обнаружения атаки. β используется для осуществления перехода от $\{X_n\}$ к $\{Z_n\}$. Чем больше β, тем реже положительные

значения будут встречаться в $\{Z_n\}$. Поэтому последовательность y_n будет с меньшей вероятностью принимать большие значения для обнаружения атаки. N — это пороговое значение для последовательности y_n. Чем больше N, тем меньше количество ложных срабатываний и тем дольше время обнаружения атаки. В результате N может быть получено из значений a и h.

Из сказанного видно, что выбор параметров алгоритма CUSUM для всех трех случаев основан на том, что при наличии DDoS-атаки элемент последовательности, сформированный на основе отслеживаемого параметра трафика, значительно отличается от элементов последовательности при нормальном функционировании. При расчете параметров алгоритма используются формулы, основанные на взаимосвязи между временем обнаружения и количеством ложных срабатываний.

Мониторинг различных IP-адресов во входящем трафике. При мониторинге новых IP-адресов во входящем трафике (Source IP Monitoring, SIM) метод CUSUM применяется к количеству новых IP-адресов за сессию. Предлагается, что обращение большого числа ранее не обращавшихся к сервису станций может свидетельствовать о DDoS-атаке. Данный метод требует предварительного обучения — формирования базы известных IP-адресов.

Мониторинг новых IP-адресов во входящем трафике состоит из двух частей: обучения и обнаружения.

Обучение состоит в добавлении легитимных IP-адресов источников в базу данных IP-адресов (IPAddress Database, IAD) [50]. На этапе обнаружения выполняется подсчет количества IP-адресов станций, впервые обратившихся к сервису, и вычисляется отклонение этой величины от среднего значения с помощью теста CUSUM. Кроме того, на этапе обнаружения в базу известных IP-адресов добавляются адреса станций, превысивших указанное администратором количество сессий взаимодействия с сервисом.

Непараметрические многомерные CUSUM тесты для быстрого обнаружения DOS-атак в компьютерных сетях. DOS-атаки и черви представляют собой наиболее распространенные и опасные категории внешних вторжений, которые охватывают широкий спектр сетевых и компьютерных ресурсов. Типичные DOS-атаки и черви могут привести к резким изменениям в статистической модели трафика по сравнению с нормальным трафиком. Более того, эти изменения происходят в неизвестные моменты времени и должны быть обнаружены «как можно скорее>. Таким образом, обнаружение компьютерных атак естественным образом вписывается

в так называемую теорию обнаружения точки изменения: выявить изменения в распределении с наименьшей задержкой, сохранив при этом уровень ложных тревог (FAR) на низком уровне.

Непараметрический многомерный CUMSUM алгоритм имеет несколько явных преимуществ.

Во-первых, он использует минимум имеющейся информации, что является важным, поскольку распределения неизмененного (легитимный трафик) и измененного (атака) трафика, как правило, заранее не известны.

Во-вторых, алгоритм обладает управляемой вычислительной сложностью и может быть использован в реальном времени.

В-третьих, алгоритм адаптивен и самообучаемый, что позволяет ему адаптироваться к различным сетевым нагрузкам и моделям использования.

Для обработки ICMP-flood и UDP-flood атак, будем наблюдать количество принятых пакетов, разделяя их по размеру и типу. Так, для TCP SYN атак, будем контролировать размер буферов связанных с полученными и переданными SYN пакетами.

Пусть $\Delta_n = t_n - t_{n-1}$ — выборка из n-го интервала. Для каждого типа пакетов pt (ICMP, UDP, TCP и т. д.) будем классифицировать пакеты по размеру и помещать в группу M_{pt}. В случае UDP и ICMP атак, будем наблюдать общее количество $N_{n,i}^{pt}$ пакетов типа pt с размерами в i-й группе, полученные в n-м интервале времени.

Для обнаружения TCPSYN атак будем отслеживать SYN пакеты, требующие размера буфера B_n в конце n-го интервала времени. В тестовом режиме статистика $N_{n,i}^{pt}$ и B_n контролируются одновременно.

Предположим, что имеется набор случайных величин $X_1, X_2, ...$, которые выбраны для наблюдения и имеют общую плотность вероятностей $p_0(X_1, ..., X_n)$ для $n < \lambda$ и другую плотность вероятностей $p_1(X_1, ..., X_n)$ для $n \geqslant \lambda$, где $\lambda \in \{1, 2, ...\}$ — неизвестная точка измерения. Другими словами, в зависимости от точки изменения $\lambda = k$ и вектора из $(n-1)$ наблюдений $(x_1, ..., x_{n-1})$ плотность условного распределения n-го наблюдения x_n будет

$$p(X_n \mid X_1, ..., X_n, \lambda = k) = \begin{cases} p_0(X_n | X_1, ..., X_{n-1}), & k > n; \\ p_1(X_n | X_1, ..., X_{n-1}), & k \leqslant n, \end{cases}$$

где p_0 и p_1 — неизмененная и измененная плотность вероятностей соответственно.

Процедура последовательного обнаружения точки изменения определяется с помощью времени остановки τ для наблюдаемой пос-

ледовательности $\{X_n\}_{n\geqslant 1}$, т. е. τ — случайная величина, зависящая от наблюдения.

Последовательный алгоритм обнаружения точки изменения использует логарифмическое отношение правдоподобия (LLR) для гипотезы H^k: $\lambda = k$, что изменение произошло в точке $\lambda = k$, и H^∞: $\lambda = \infty$, что изменения нет. Она определяется так:

$$z_n^k = \sum_{j=k}^{n} \log \frac{p_1\left(X_j \mid X_1, ..., X_{j-1}\right)}{p_0\left(X_j \mid X_1, ..., X_{j-1}\right)}, \quad n \geqslant k.$$

Процедура CUSUM определяется, когда максимум LLR статистики $U_n = \max\{0, \max_{1\leqslant k\leqslant n} z_n^k\}$ в первый раз превышает положительный порог h:

$$\tau_{CU} = \tau_{CU}(h) = \min\{n \geqslant 1 : U_n \geqslant h\}.$$

Так как сетевые атаки происходят в неизвестный момент времени и обычно приводят к резкому изменению нормального сетевого трафика, мы считаем, что обнаружение аномалий методом точки изменения идеально вписывается в типичные сценарии DOS-атак. Тем не менее хорошо развитые параметрические методы не могут быть достаточно эффективными, так как они требуют слишком много предварительной информации, которой нет. Поэтому надежные непараметрические методы оказываются крайне необходимыми. Предлагаемый непараметрический многомерный CUSUM метод обладает несколькими привлекательными особенностями. Во-первых, он легко реализуем в распределенных компьютерных сетях. Во-вторых, он почти не требует предварительной информации и потому довольно надежен. В-третьих, когда он оптимизирован и настроен, метод становится крайне эффективным. Исследования показали, что данный алгоритм позволяет быстро обнаруживать различные DOS атаки с низким уровнем ложных тревог.

5.4.4. Контрольные карты EWMA

Контрольные карты EWMA (Exponentially Weighted Moving Average) опираются на экспоненциальное сглаживание наблюдений. С учетом постоянной сглаживания λ $(0 < \lambda < 1)$

$$z_t = \lambda y_t + (1-\lambda) z_{t-1} = \lambda \sum_{i=0}^{t-1} (1-\lambda)^i y_{t-i} + (1-\lambda)^t z_0. \qquad (5.13)$$

Представляет собой взвешенное среднее всех наблюдений до времени t. Начальное значение наблюдаемого среднего H_0 $\mu_0 = z_0$. Если наблюдения независимы и имеют одинаковое распределение с

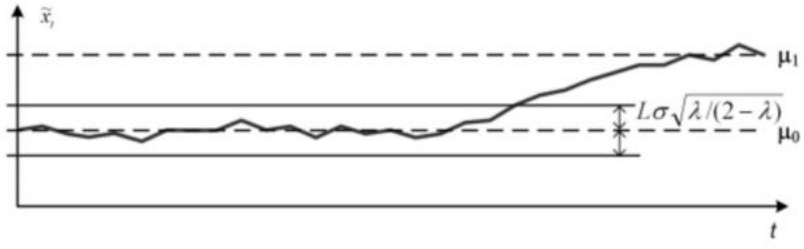

Рис. 5.3. EWMA контрольная карта

дисперсией σ^2, то дисперсия z_1 подходит для $\lambda/(2-\lambda)\sigma^2$ при $t \to \infty$, что позволяет определить контрольные пределы для z_t:

$$\mu_0 \pm L\sigma\sqrt{\frac{\lambda}{2-\lambda}}, \tag{5.14}$$

где λ и L — основные параметры метода EWMA контрольных карт. Популярный выбор $2{,}6 \leqslant L \leqslant 3$ и $0{,}05 < \lambda < 0{,}25$ обусловлен тем, что чем меньше λ, тем более маленькие сдвиги можно обнаружить.

Метод контрольных карт EWMA имеет некоторые интересные свойства. Во-первых, он может быть настроен для достижения результатов, схожих с методом CUSUM. Во-вторых, он довольно устойчив к нестандартным распределениям Y при малых значениях λ (т. е. $\lambda = 0{,}05$). Наконец, после корректировки контрольных пределов метод контрольных карт EWMA все еще хорошо работает при низких уровнях сериальной корреляции в Y_t.

5.5. Критерии аномального поведения и их практическое применение

Признаком появления аномалии в потоке будем считать значительное отклонение локальных статистических характеристик от соответствующих глобальных характеристик. Для выявления аномалий потока в сети будем использовать в качестве статистических характеристик выборочное среднее числовой характеристики X:

$$\xi = \sum_{b=1}^{B} \bar{x}_b Y_b, \tag{5.15}$$

где $\bar{x}_b = (x_{b-1} + x_b)/2$ — середина полуинтервала $[x_{b-1}, x_b)$, выборочную дисперсию

$$d^2 = \sum_{b=1}^{B} (\bar{x}_b - \xi)^2 Y_b \tag{5.16}$$

и характеристику

$$\chi^2 = \sum_{b=1}^{B} \frac{(Y_b - y_b)^2}{y_b}. \tag{5.17}$$

Величина χ^2 подчиняется хорошо известному χ^2-распределению с $(B-1)$ степенями свободы.

Для каждой из используемых статистических характеристик (5.15)–(5.17) можно сформулировать свой критерий присутствия аномалии в потоке событий.

Для выборочного среднего (5.15) признаком аномалии будем считать превышение заданного порога при отклонении величины ξ от её среднего значения:

$$|\xi - \bar{\xi}| \geqslant k\sigma_x, \tag{5.18}$$

где $\xi = \sum_{b=1}^{B} \bar{x}_b y_b$ — математическое ожидание величины ξ; $y_b \approx p_b$ — вероятность попадания события в контейнер с номером b; k задаёт границы интервала $[\bar{\xi} - k\sigma_x, \bar{\xi} + k\sigma_x]$, выход за пределы этого интервала мы воспринимаем как аномалию.

Используя то обстоятельство, что распределение суммы независимых случайных величин близко к нормальному распределению, можно записать

$$k \approx u_\alpha / \sqrt{n},$$

где u_α — α-значение нормального отклонения; α — вероятность случайного отклонения величины ξ за пределы (5.18) и n — количество событий, участвующих в формировании локальных статистических характеристик. Таким образом, при выполнении условия (5.18) можно считать, что с вероятностью $1 - \alpha$ это отклонение вызвано появлением аномалии. Так, например, при выбранном числе событий $n = 30$, вероятности $\alpha = 0{,}001$ и соответствующем табличном значении $u_\alpha = 3{,}30$ [1] параметр $k \approx 0{,}6$.

5.5.1. Процентное отклонение

Процентное отклонение (PD) определяется как

$$\mathrm{PD}_x = (x - m_x) \cdot 100.$$

где m_x — медиана последовательности. Затем для всей анализируемой трассы вычисляется среднее процентное отклонение

$$\mathrm{PD}_{\mathrm{avg}} = \frac{1}{n} \sum_{i=1}^{n} \mathrm{PD}_i,$$

где n — размер окна. Если вся анализируемая последовательность будет длиной N, то обработке подлежат $(N-n)$ значений PD.

Если в качестве характеристики потока событий использовать характеристику χ^2 (5.17), то признаком появления аномалии будет считаться превышение χ^2 установленного порогового значения

$$\chi^2 \geqslant X_0^2. \tag{5.19}$$

Критерии (5.18) и (5.19) аномального поведения потока событий не эквивалентны, факт появления аномалии по одному критерию может соответствовать нормальному поведению потока согласно другому критерию [18]. Это связано с тем обстоятельством, что используемые критерии введены для разных статистических характеристик. В случае критерия (5.18) оценивают отклонение выборочного среднего от математического ожидания, в случае (5.19) отклонение плотности локальной функции распределения от плотности глобальной функции распределения величины X.

Для согласования используемых критериев рассмотрим ситуацию, в которой возникновение аномалии приводит к одновременному выполнению условий (5.18), (5.19) и установим связь между параметрами k и X_0^2. Будем считать, что в обычном состоянии при отсутствии аномалий частоты попадания событий в контейнеры имеют вид

$$Y_b = y_b + \beta_b \text{ при } 1 \leqslant b \leqslant B,$$

а аномалия заключается в том, что частота попадания в первый контейнер возрастает на регулярную величину δ:

$$\begin{aligned} Y_1 &= y_1 + \beta_1 + \delta; \\ Y_b &= y_b + \beta_b - \frac{\delta}{(B-1)} \quad \text{при } 1 \leqslant b \leqslant B. \end{aligned} \tag{5.20}$$

В реальности подобная ситуация может возникнуть при сканировании сети или flood-атаке, когда на фоне обычного сетевого трафика появляется множество пакетов с близкими характеристиками. В этом случае из условий (5.18) и (5.19), представления (5.20) для Y_b и свойств вариации β_b

$$M[\beta_b] = 0; \quad M\left[\sum_{k=1}^{B} \beta_b^2\right] = \frac{B-1}{n},$$

где $M[z]$ — математическое ожидание величины z, находим связь между параметрами двух разных критериев наличия аномалий в

сети:

$$X_0^2 = (B-1)\left(1 + n\frac{k^2\sigma_x^2}{(\bar{x}-\tilde{x}_1)^2}\right). \tag{5.21}$$

При получении соотношения (5.21) было использовано допущение о равенстве вероятностей $p_b \approx 1/B$, $1 \leqslant b \leqslant B$, это легко реализовать посредством подбора границ полуинтервалов $[x_{b-1}, x_b)$ таким образом, чтобы вероятности попадания событий в разные контейнеры были равны.

Рассуждая аналогичным образом, для случая использования d^2 (5.16) в качестве статистических характеристик потока событий находим значение для граничной величины S_0^2:

$$S_0^2 = \sigma_x^2\left(1 + k\frac{\bar{x}-\tilde{x}_1}{\sigma_x}\right)\left(1 - k\frac{\sigma_x}{\bar{x}-\tilde{x}_1}\right). \tag{5.22}$$

Величина S_0^2 задаёт нижнюю границу, выход за пределы которой свидетельствует о наличии аномалии:

$$d^2 \leqslant S_0^2. \tag{5.23}$$

Для того чтобы значение S_0^2 (5.22) соответствовало критерию приведения аномалии (5.23), необходимо выполнение условия

$$k > \frac{\bar{x}-\tilde{x}_1}{\sigma_x} - \frac{\sigma_x}{\bar{x}-\tilde{x}_1}. \tag{5.24}$$

В случае использования в качестве числовой характеристики X временного интервала между двумя соседними пакетами $X_i = t_i - t_{i-1}$ и при условии, что плотность функции распределения реального потока пакетов в сети близка к плотности функции распределения для стационарного пуассоновского потока, последнее условие обычно выполняется, так как для стационарного пуассоновского потока значение математического ожидания и среднеквадратического отклонения равны.

Графики, приведенные на рис. 5.4, иллюстрируют возможность практического использования различных критериев присутствия аномалии в потоке пакетов сети.

При построении графиков использованы реальные данные, полученные на одном из узлов сети. В качестве числовой характеристики X используется временной интервал между соседними пакетами: $X_i = t_i - t_{i-1}$. В приведенном примере средний интервал времени между пакетами и среднеквадратическое отклонение составляют $\bar{x} \approx \sigma_x \approx 85$ мс. При обработке потока пакетов и нахождении статистических характеристик (5.15), (5.16) и (5.17) были выбраны значения $B = 5$ и $n = 30$.

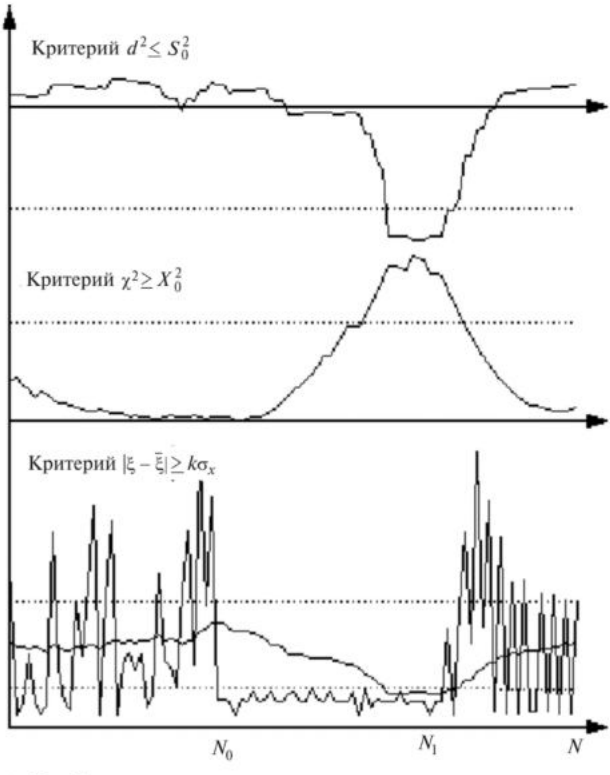

Рис. 5.4. Графики поведения локальных статистических характеристик

Вдоль горизонтальных осей графиков отложены номера событий в сети (событие — поступление нового пакета), вертикальная ось соответствует промежутку времени между приходом двух пакетов для нижнего графика и значениям статистики χ^2 и $(d^2 - \sigma_x^2)$ для двух других графиков. На верхнем графике показано поведение выборочной дисперсии d^2 (5.16) минус σ_x^2, на среднем — поведение статистики χ^2 (5.17), на нижнем графике приведены значения числовых характеристик X_i поступающих пакетов и поведение выборочного среднего ξ (5.15). Пунктирные линии на графиках обозначают границы (5.18), (5.19) и (5.23), выход за указанные границы свидетельствует о наличии аномалии в сети. Начиная с пакета номер N_0, трафик резко возрастает, средняя частота поступления пакетов увеличивается в 5...6 раз. Из приведенных графиков видно, что в этом случае значения статистических характеристик d^2, χ^2 и ξ выходят за границы «коридора» допустимых значения и используемые критерии указывают на появление аномалии.

5.5.2. Энтропия

Спонтанные изменения в сигнале должны увеличить энтропию. Измерение энтропиии ограничено внутри функции плотности распределения вероятности, ее можно также измерить с помощью информации Реньи как

$$H_r(x) = \frac{1}{1-r} \log \left(\int_0^{T\max} f_x^r(t)\, dt \right), \quad 0 < r < \infty,\ r \neq 1.$$

Заметим, что при $r = 1$ выражение является определением классической энтропии. Мы будем вычислять при $r = 3$, поэтому для дискретной серии длиной N энтропия H определяется как

$$H_3(x) = -\frac{1}{2} \log \left(\frac{1}{N} \sum_{i=1}^{N} f_x^3(i) \right).$$

5.6. Методы описательной статистики

Один из способов формирования «образа» нормального поведения системы состоит в накоплении в специальной структуре измерений значений параметров оценки. Эта структура называется профайлом. Основные требования, которые предъявляются к структуре профайла: минимальный конечный размер, операция обновления должна выполняться как можно быстрее.

В профайле используется несколько типов измерений, например в IDES используются следующие типы.

Показатель активности — величина, при превышении которой активность подсистемы оценивается как быстро прогрессирующая. В общем случае используется для обнаружения аномалий, связанных с резким ускорением в работе. Пример: среднее число записей аудита, обрабатываемых для элемента защищаемой системы в единицу времени.

Распределение активности в записях аудита — распределение во всех типах активности в свежих записях аудита. Здесь под активностью понимается любое действие в системе, например доступ к файлам, операции ввода-вывода.

Измерение категорий — распределение определенной активности в категории (категория — группа подсистем, объединенных по некоему общему принципу). Например, относительная частота регистрации в системе (логинов) из каждого физического места нахождения. Предпочтения в использовании программного обеспечения системы (почтовые службы, компиляторы, командные интерпретаторы, редакторы и т. д.).

Порядковые измерения — величина используется для оценки активности, которая поступает в виде цифровых значений. Например, количество операций ввода-вывода, инициируемых каждым пользователем. Порядковые изменения вычисляют общую числовую статистику значений определенной активности, в то время как измерение категорий подсчитывают количество активностей.

При обнаружении аномалий с использованием профайла в основном применяют статистические методы оценки. Процесс обнаружения происходит следующим образом: текущие значения измерений профайла сравнивают с сохраненными значениями. Результат сравнения — показатель аномальности в измерении. Общий показатель аномальности в простейшем случае может вычисляться при помощи некоторой общей функции от значений показателя аномалии в каждом из измерении профайла. Например, пусть $M_1, M_2, \ldots, M_n$ — измерения профайла, а $S_1, S_2, \ldots, S_n$ — значения аномалии каждого из измерений, причем чем больше число S_i, тем больше аномалии в i-м показателе. Объединяющая функция может быть весом сумм их квадратов:

$$a_1 s_1^2 + a_2 s_2^2 + \ldots + a_n s_n^2 > 0, \tag{5.25}$$

где a_i — показывает относительный вес метрики M_i.

Параметры $M_1, M_2, \ldots, M_n$ на самом деле могут зависеть друг от друга, и поэтому для их объединения может потребоваться более сложная функция.

Основное преимущество заключается в том, что применяются хорошо известные статистические методы.

Недостатки:

1) нечувствительность к последовательности возникновения событий. То есть статистическое обнаружение может упустить вторжение, которое проявляется в виде последовательности сходных событий;

2) система может быть последовательно обучена таким образом, что аномальное поведение будет считаться нормальным. Злоумышленники, которые знают, что за ними наблюдают при помощи таких систем, могут обучить их для использования в своих целях. Именно поэтому в большинстве существующих схем обнаружения вторжения используется комбинация подсистем обнаружения аномалий и злоупотреблений;

3) трудно определить порог, выше которого аномалии можно рассматривать как вторжение. Занижение порога приводит к лож-

ному срабатыванию (false positive), а завышение — к пропуску вторжений (false negative);

4) существуют ограничения к типам поведения, которые могут быть смоделированы, используя чистые статистические методы. Применение статистических технологий для обнаружения аномалий требует предположения, что данные поступают от квазистатического процесса.

5.7. Поиск и оценка аномалий сетевого трафика на основе циклического анализа

Общая схема управления трафиком ВС на основе выявления аномалий предусматривает, что в ней могут быть выделены следующие функциональные блоки:

- извлечение информации о сетевых пакетах;
- построение прогноза;
- поиск и оценка аномалии;
- реагирование на аномалию;
- заполнение и редактирование базы правил (БП).

На первом этапе из трафика извлекается вся необходимая для прогнозирования информация. Поскольку цель прогнозирования на основе имеющихся данных о загрузки сети — получить объем трафика на определенный период времени в будущем, поэтому из заголовка IP-пакета необходимо выделять информацию об общей длине пакета, а также сохранять дату и время получения пакета. При фильтрации может быть использована информация об адресе источника и адресе назначения IP-пакета.

Таким образом, для прогнозирования трафика из IP-пакета извлекается следующая информация:

- объем IP-пакета;
- IP-адрес источника;
- IP-адрес назначения;
- дата получения IP-пакета;
- время получения IP-пакета.

На основе собранной статистики производится прогнозирование сетевого трафика, например на базе циклического анализа временных рядов. Рассмотрим основные шаги, необходимые для проведения циклического анализа.

Отбор данных. На первом этапе необходимо определить форму и количество данных, на которых будет производиться прогнозирование. Циклический анализ сильно зависит от однородности данных. Используемые данные должны иметь однородную структуру,

иначе неоднородность данных при анализе, скорее всего, изменит структуру циклов. Таким образом резкое изменение в работе ВС (например, подключении большого количество хостов или изменение в расписании), которое может изменить форму циклов, должно учитываться при поиске и оценке аномалии.

Поскольку циклический анализ предполагает работу с рядом данных, необходимо сформировать имеющиеся данные по сетевому трафику в виде ряда значений, описывающих изменение объема трафика во времени. Для этого необходимо провести дискретизацию потока трафика. Рассмотрим этот процесс на примере (рис. 5.5).

На рисунке представлен график, изображающий поток сетевого трафика: на оси абсцисс представлено время t, на оси ординат отложен объем трафика V. Пусть имеется статистика по трафику, собранная за период времени T. Чтобы получить ряд данных, разделим период времени T на Q равных интервалов Δt:

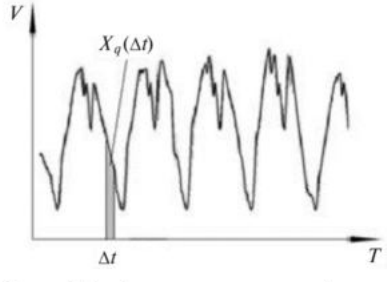

Рис. 5.5. Дискретизация трафика

$$Q = \frac{T}{\Delta t}. \tag{5.26}$$

Далее для каждого интервала Δt складываем объемы сетевых пакетов, попавших в данный интервал времени:

$$X_q(\Delta t) = \sum_{j=1}^{R} V_{\text{пакет}}, \tag{5.27}$$

где R — количество сетевых пакетов, попавших в интервал Δt; q — номер интервала, $q = 1, ..., Q$; X_q — ряд упорядоченных данных, описывающий изменения объема трафика во времени, с частотой дискретизации Δt.

Сглаживание данных. Определившись с данными, необходимо исключить из трафика случайные колебания. Для этого предусмотрено сглаживание данных.

Для устранения случайных колебаний используется метод краткосрочной центрированной скользящей средней ряда данных. Количество точек для сглаживания данных возьмем равным L. При вычислении скользящей средней по L точкам, из первоначального ряда данных будет выброшено $L - 1$ точек, по $(L - 1)/2$ точек в начале и в конце ряда. Таким образом, длина нового ряда данных $\overline{X}_k$

равна $N = Q - (L-1)$, $k = 1, ..., N$:

$$\overline{X}_k = \frac{1}{L} \sum_{j=k}^{(k+L-1)} X_j. \tag{5.28}$$

Поиск возможных циклов. Устранив случайные колебания, можно приступить к непосредственному поиску циклов. Чтобы определить частотные составляющие рассматриваемого ряда, используем метод спектрального анализа. Математической основой спектрального анализа является преобразование Фурье [2]. Поскольку обрабатываемая статистика сетевого трафика имеет вид цифрового ряда, для определения частотных составляющих подойдет метод дискретного преобразования Фурье. С помощью прямого дискретного преобразования Фурье найдем комплексные амплитуды ряда данных $\overline{X}_k$:

$$Y_n = \sum_{k=1}^{N} \overline{X}_k \mathrm{e}^{-2\pi i n k/N}, \tag{5.29}$$

где N — количество элементов ряда данных $\overline{X}_k$ и количество компонентов разложения; i — мнимая единица.

Модуль комплексного числа может быть найден как

$$|Y_n| = \sqrt{\mathrm{Re}^2(Y_n) + \mathrm{Im}^2(Y_n)}.$$

На основе комплексных амплитуд Y_k вычисляется спектр мощности $R_n = |Y_n|^2 = \mathrm{Re}^2(Y_n) + \mathrm{Im}^2(Y_n)$.

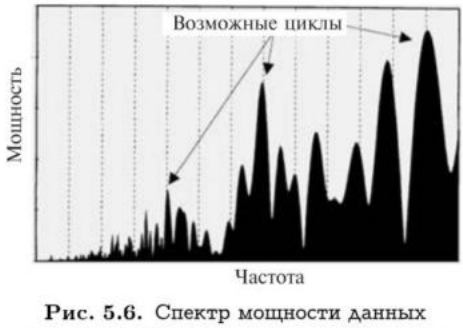

Рис. 5.6. Спектр мощности данных

Изобразим спектр мощности графически (рис. 5.6) [3]. На рисунке видно, что высокие значения скапливаются около некоторых частот. Пики в областях скопления высоких значений показывают возможные циклы. Значением частоты цикла будет являться индекс n, при котором наблюдается высокое значение спектра мощности R_n.

Определив возможные циклы и их частоты, рассчитаем обычную (вещественную) амплитуду A и фазу φ. Пусть найдено b возможных циклов, частоты которых составляют множество S, т. е. каждое значение частоты, при которой наблюдается пик в области скопления высоких значений спектра, является элементом мно-

жества S. Тогда амплитуды и фазы найденных циклов могут быть вычислены по формулам

$$A_h = \frac{|S_h|}{N} = \frac{1}{N}\sqrt{\mathrm{Re}^2(S_h) + \mathrm{Im}^2(S_h)};$$
$$\varphi_h = \mathrm{Arg}(S_h) = \mathrm{arctg}\left(\frac{\mathrm{Im}(S_h)}{\mathrm{Re}(S_h)}\right),$$

где $h = 1, ..., b$; $\mathrm{Arg}(S_h)$ — функция мнимого числа: угол мнимого числа (в радианах), соответствующий (S_h).

Функция, описывающая цикл, $f_h(t) = A_h \cos(S_h t + \varphi_h)$. Однако, как уже было сказано, высокое значение спектра мощности лишь предполагает наличие цикла. Поэтому следующим шагом является подтверждение найденных циклов. Для этого необходимо проверить определенное количество критериев.

Удаление трендовых компонентов в трафике. Качество проверки циклов на статистическую надежность сильно зависит от существования направленности в данных. Поэтому перед проверкой необходимо провести удаление тренда из данных. Для этого можно применить метод отклонения от скользящего среднего. В данном случае скользящая средняя будет отражать силы роста в данных, следовательно, ее вычитание из данных удалит и трендовую составляющую. Таким образом, чтобы удалить тренд в данных, необходимо для каждой найденной частоты рассчитать *скользящую* среднюю для ряда данных $\overline{X}_k$ с количеством точек сглаживания $L = S_h$:

$$\overline{X}_k = \frac{1}{L}\sum_{j=k}^{(k+L-1)} \overline{X}_j, \tag{5.32}$$

где полученный ряд данных будет короче исходного на $L - 1$ точек: $N' = N - (L - 1)$, $k = 1, ..., N'$.

Далее вычитаем из исходного ряда данных $\overline{X}_k$ полученную скользящую среднею $\overline{X'}_k$:

$$\overline{X}''_k = \overline{X}_k - \overline{X}'_k. \tag{5.33}$$

Удалив силы роста в данных, можно приступать к проверке найденных циклов на статистическую значимость.

Проверка циклов с точки зрения статистической значимости. Для оценки циклов обычно используют тесты F-коэффициент и хи-квадрат, поэтому они же будут использованы для проверки циклов в сетевом трафике. Отметим, что результаты теста зависят от количества повторений цикла в данных. Чем таких повторений больше, тем более статистически значим данный цикл.

Комбинирование и проецирование циклов в будущее. Прогнозирование трафика происходит на этапе комбинирования и проецирования циклов. Для этого циклы объединяются, и на основе полученного результата можно спрогнозировать их поведение в будущее. Для проецирования циклы математически комбинируются в одну общую кривую. Допустим, что тесты прошло D циклов. Подтвердившиеся циклы проецируются в общую кривую, описывающую периодичность в ряде данных:

$$\overline{V}(t) = \sum_{j=1}^{D} f_h(t). \tag{5.34}$$

Данная функция описывает периодичность в трафике, найденную на основе данных за период времени T. Полученная функция может быть экстраполирована в будущее и позволяет получить прогнозируемое значение трафика на период времени $\overline{T}$ в будущее:

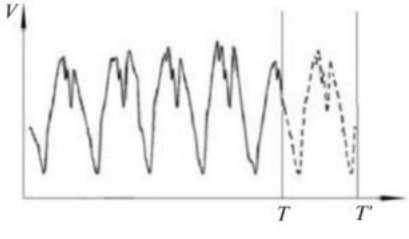

Рис. 5.7. Прогнозирование трафика

$$V_{\text{прогноз}}(t') = \sum_{j=1}^{D} f_h(t'), \tag{5.35}$$

где $t' \in (T, \overline{T})$ (рис. 5.7).

Определив математическую модель прогнозирования сетевого трафика, рассмотрим систему поддержки принятия решений (СППР) поиска и оценки аномалии (рис. 5.8).

Для определения объема трафика, поступающего в реальном времени, используется извлекаемая информация о сетевых пакетах. На основе получаемых данных о трафике и текущем прогнозе дает описание аномалии. В случае, если аномалия была найдена, результат поиска аномалии передается на блок поиска источников аномалии. Определение источников аномалии осуществляется на основе результата поиска аномалии и информации о сетевых пакетах, поступающих в реальном времени. Далее производится оценка величины аномалии. В оценке используется полученная информация о нарушителях, информация о пакетах, а также лицо, принимающее решение (ЛПР), и эксперт. Информация о величине аномалии обобщается и передается для дальнейшего использования.

Поиск аномалии происходит на основе сравнения трафика, поступающего в реальном времени, с прогнозируемым значением. Для этого в единицу времени t сравниваются два значения $V_{\text{реал}}$ — объ-

Рис. 5.8. Поиск и оценка величины аномалии

ем текущего трафика и $V_{\text{прогноз}}$ — прогнозируемый объем. Аномальным будет считать отклонение объема, превышающее или равное заданной величине α:

$$|V_{\text{реал}} - V_{\text{прогноз}}| \geqslant \alpha. \tag{5.36}$$

Отметим, что прогноз представлен в виде относительно гладкой кривой. В свою очередь, трафик состоит из кратковременных случайных флуктуаций. Если сравнивать эти два ряда данных, возможны ложные определения аномалий (рис. 5.9,*а*). Чтобы этого избежать, реальный трафик сглаживается методом скользящей средней, при этом окно сглаживания смещается по мере поступления нового трафика (рис. 5.9,*б*).

Если аномалия была обнаружена, происходит поиск источников аномалии. Поиск источников определяется на основе информации, извлекаемой из текущего трафика.

Оценка величины аномалии происходит на основе продукционной базы правил. Ее начальным заполнением занимается эксперт.

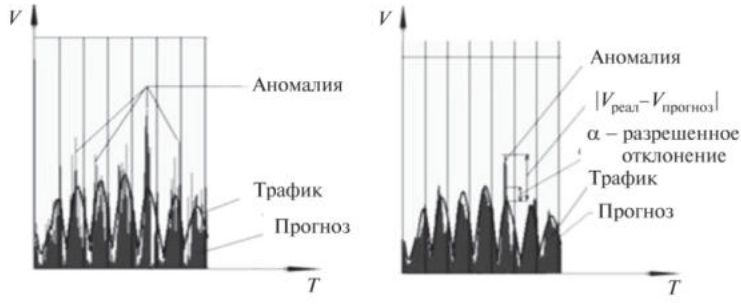

Рис. 5.9. Аномалия трафика: *а* — трафик до сглаживания; *б* — трафик после сглаживания

В дальнейшем ЛПР может корректировать БП исходя из результатов фильтрации трафика. Рассмотрим структуру БП более подробно, введя понятие «терма».

Термом назовем некоторое понятие предметной области, имеющее определенный синтаксис и семантическое содержание. С каждым модулем связан набор входных и выходных термов. Каждый терм приписан к одному и только одному модулю. Входные термы — это понятия, необходимые для понимания содержания модуля. Они должны быть определены на ранних этапах обучения. Это соответствие также является задачей анализа плана. Выходные термы — это понятия, которые вводятся при чтении соответствующего модуля и которые могут использоваться в последующих модулях. В результате модуль можно рассматривать как оператор преобразования входных термов в выходные.

Для оценки величины аномалии можно использовать следующие лингвистические переменные с соответствующими терм-множествами:

величина отклонения $V = \{$низкая, ниже среднего, средняя, выше среднего, высокая$\}$;

частота появления аномалии $M = \{$низкая, ниже среднего, средняя, выше среднего, постоянная$\}$;

количество источников аномалий $I = \{$незначительное, ниже среднего, среднее, выше среднего, большое$\}$;

средний объем трафика от одного источника $W = \{$незначительный, ниже среднего, средний, выше среднего, высокий$\}$.

Выходным параметром является величина аномалии $E = \{$незначительная, ниже среднего, средняя, выше среднего, высокая$\}$.

На основе введенных переменных формируется набор правил.

Примеры правил:

ЕСЛИ $V = \{$ОТ низкая ДО ниже среднего$\}$ и $M = \{$ОТ низкая ДО средняя$\}$ и $I = \{$ОТ незначительное ДО среднее$\}$ и $W = \{$ОТ незначительный ДО средний$\}$, ТО $E = \{$незначительная$\}$;

ЕСЛИ $V = \{$ОТ выше среднего ДО высокая$\}$ и $M = \{$ОТ постоянная ДО постоянная $\}$ и $I = \{$ОТ выше среднего ДО большое$\}$ и $W = \{$ОТ выше среднего ДО выше среднего$\}$, ТО $E = \{$высокая$\}$.

При формировании набора правил в качестве основы была использована схема с N экспертами, каждый из которых независимо друг от друга, продуцирует набор правил [3].

Суть данного подхода заключается в следующем. Каждый эксперт создает свой набор правил. Сгенерированный набор правил каждым последующим экспертом дополняет базу правил новыми правилами, тем самым увеличивая полноту модели. Поскольку заполнение базы правил экспертами происходит независимо, в каждом последующем наборе правил могут содержаться правила, которые могут повторять уже существующие правила. Также в новом наборе правил могут быть правила, противоречащие правилам из других наборов. Таким образом, появляется проблема проверки базы правил на противоречивость, избыточность и полноту.

Понятие однозначности означает, что каждому сочетанию координат V, M, I, W соответствует только одно значение выходной координаты E. В идеале правила должны полностью соответствовать понятию однозначности, однако из-за возможной размытости знаний экспертов и значениях лингвистических переменных допускается частичная однозначность.

Избыточность подразумевает ситуацию, когда одно правило включает в себя другое правило из общего набора, например:

Правило 1:

ЕСЛИ $V = \{$ОТ низкая ДО ниже среднего$\}$ и $M = \{$ОТ низкая ДО средняя$\}$ и $I = \{$ОТ незначительное ДО среднее$\}$ и $W = \{$ОТ незначительный ДО средний$\}$, ТО $E = \{$незначительная$\}$.

Правило 2:

ЕСЛИ $V = \{$ОТ низкая ДО ниже среднего$\}$ и $M = \{$ОТ низкая ДО средняя$\}$ и $I = \{$ОТ незначительное ДО среднее$\}$ и $W = \{$ОТ незначительный ДО незначительный$\}$, ТО $E = \{$незначительная$\}$.

Как видно из примера, в первых трех координатах правила полностью идентичны, однако параметр «средний объем трафика от одного источника» для первого правила измеряется «ОТ незначительный ДО средний», во втором случае этот параметр имеет диапазон «ОТ незначительный ДО незначительный» Очевидно, что второе правило входит в первое правило.

Понятие противоречивости. Если два правила имеют на входе одинаковые значения координат V, M, I, W, а на выходе значение E различно (нарушение гипотезы однозначности), то данные правила считаются противоречивыми:

Правило 1:

ЕСЛИ $V = \{$ОТ низкая ДО ниже среднего$\}$ и $M = \{$ОТ низкая ДО средняя$\}$ и $I = \{$ОТ незначительное ДО среднее$\}$ и $W = \{$ОТ незначительный ДО незначительный$\}$, ТО $E = \{$незначительная$\}$.

Правило 2:

ЕСЛИ $V = \{$ОТ низкая ДО ниже среднего$\}$ и $M = \{$ОТ низкая ДО средняя$\}$ и $I = \{$ОТ незначительное ДО среднее$\}$ и $W = \{$ОТ незначительный ДО незначительный$\}$, ТО $E = \{$средняя$\}$.

Под полнотой понимается отношение доли охвата знаний выходных координат к общему диапазону возможных решений. Для проверки полноты базы правил необходимо найти отношение количества выходных значений для существующей базы правил к количеству всех возможных значений.

Чтобы определить число существующих выходных значений, необходимо сложить все решения от каждого созданного правила:

$$S_{\text{сущ}} = \sum \text{Решения\_правила1} + \sum \text{Решения\_правила2} + \ldots + \sum \text{Решения\_правила}N.$$

Полное множество возможных решений может быть рассчитано перебором всех возможных значений входных координат:

$$S_{\text{общ}} = S_v S_m S_i S_w,$$

где S_v, S_m, S_i, S_w — количество значений лингвистических переменных. Тогда полное множество возможных решений

$$S_{\text{общ}} = 5 \cdot 5 \cdot 5 \cdot 5 = 625.$$

Тогда полнота базы правил

$$\text{П} = \frac{S_{\text{сущ}}}{S_{\text{общ}}} \cdot 100\ \%. \tag{5.37}$$

Если отношение меньше 100%, производится поиск правил, которые не были учтены экспертами. На основе полученного результата формируется новый набор правил, который передается для оценки экспертам. Далее снова проводится проверка базы правил на противоречивость, избыточность и полноту до тех пор, пока база не будет полностью сформирована. После рассмотрения всех ин-

дивидуальных наборов правил экспертов формирование правил заканчивается.

Если в процессе эксплуатации произойдет ситуация, решение которой не будет в базе правил (например, ошибки при обучении), предлагается два варианта действий: блокировка аномального трафика и ожидание действий ЛПР. Получив решение, система формирует на его основе правило и пополняет базу правил. Во втором случае система выбирает наиболее подходящее правило из базы правил и выполняет действие, исходя из найденного решения. При этом ЛПР может либо добавить новое правило в базу правил, либо подтвердить выбор СППР, и тогда новое правило будет добавлено автоматически.

Таким образом, величина аномалии объема сетевого трафика может быть представлена в виде системы:

$$e = \begin{cases} \{\{V\},\{M\},\{I\},\{W\},\{E\}\}, & |V_{\text{реал}} - V_{\text{прогноз}}| \geqslant \alpha; \\ 0, & |V_{\text{реал}} - V_{\text{прогноз}}| < \alpha, \end{cases} \quad (5.38)$$

где $\{V\},\{M\},\{I\},\{W\}$ — лингвистические переменные, применяемые для оценки величины аномалии объема трафика; $\{E\}$ — множество значений лингвистической переменной, определяющей выходные значения базы правил, $V_{\text{реал}}$ — объем трафика, поступающего из сети в реальном времени; $V_{\text{прогноз}}$ — прогнозируемой значение трафика; α – величина, определяющая, какое отклонение трафика, поступающего из сети в реальном времени, от прогнозируемого значения можно считать аномальным.

Процесс реагирования на аномалию может быть представлен в виде трех основных блоков:

- определение необходимости фильтрации;
- фильтрация пакетов;
- подготовка отчета об аномалии.

Необходимость фильтрации определяется на основании информации об аномалии, а также за счет настроек ЛПР, на основе которых формируются исключения для фильтрации $\{Z\}$ (список заблокированных источников; источники, которые запрещено блокировать и т. п.). Далее происходит непосредственная фильтрация трафика и подготовка отчета об аномалии.

Рассмотрим блок фильтрации (рис. 5.10). Первым действием является обновление списка для фильтрации. В данном блоке добавляются новые источники в список фильтрации, а также удаляются источники, время блокирования которых истекло. Далее подготавливаются параметры для фильтрации, на основе которых в фильтре пакетов формируются правила фильтрации трафика.

Рис. 5.10. Фильтрация трафика

Чтобы различать трафик из разных подсетей, необходимо учитывать IP-адрес и маску подсети. Это позволит индивидуально настраивать фильтрацию трафика для каждой подсети. Также должно быть предусмотрено раздельное отслеживание входящего и исходящего трафика.

Поскольку трафики из внешних и внутренних сетей имеют разную информативность и, как следствие, различия при построении модели прогноза, то имеет принципиальное значение разделение трафиков из внешней и внутренней сети.

Таким образом, при поиске аномалий объема сетевого трафика, необходимо использовать следующие характеристики:

- величина отклонения реального трафика от прогнозируемого;
- размер окна для сглаживания реального трафика;
- IP-адрес подсети и маски;
- направление трафика (входящий или исходящий);
- внешняя или внутренняя сеть.

Для определения параметров фильтрации также используется база правил, которая на основе величины аномалии определяет время фильтрации $\{F\}$. В качестве входного параметра используется величина аномалии e, описанная ранее.

Пример правил:

ЕСЛИ $E = \{$незначительная$\}$ ТО Время блокирования $= 1$ минута;
ЕСЛИ $E = \{$высокая$\}$ ТО Время блокирования $= 120$ минут.

Полученные новые параметры фильтрации используются для настройки фильтра пакетов. Также формируется отчет о проведенной фильтрации. Общая схема формирования правил фильтрации может быть представлена как

$$G = \{e, \{Z\}, \{F\}, \{U\}\},$$

где $\{Z\}$ — сетевые адреса источников аномалии; $\{F\}$ — время фильтрации, определяется в зависимости от величины e; $\{U\}$ — список исключений фильтрации.

5.8. Обнаружение аномалий методом главных компонент

5.8.1. Метод главных компонент

Метод главных компонент (Principal Components Analysis, PCA) — один из основных способов уменьшить размерность данных, потеряв наименьшее количество информации. Изобретен К. Пирсоном в 1901 г. Применяется во многих областях, таких, как распознавание образов, компьютерное зрение, сжатие данных и т. п. Вычисление главных компонент сводится к вычислению собственных векторов и собственных значений ковариационной матрицы исходных данных или к сингулярному разложению матрицы данных. Иногда метод главных компонент называют преобразованием Кархунена–Лоэва (Karhunen–Loeve) или преобразованием Хотеллинга (Hotelling transform).

Другие способы уменьшения размерности данных — это метод независимых компонент, многомерное шкалирование, а также многочисленные нелинейные обобщения: метод главных кривых и многообразий, поиск наилучшей проекции (Projection Pursuit), нейросетевые методы «узкого горла», самоорганизующиеся карты Кохонена и др.

Формальная постановка задачи. Задача анализа главных компонент имеет, как минимум, четыре базовых версии:

1) аппроксимировать данные линейными многообразиями меньшей размерности;

2) найти подпространства меньшей размерности, в ортогональной проекции на которые разброс данных (т. е. среднеквадратичное уклонение от среднего значения) максимален;

3) найти подпространства меньшей размерности, в ортогональной проекции на которые среднеквадратичное расстояние между точками максимально;

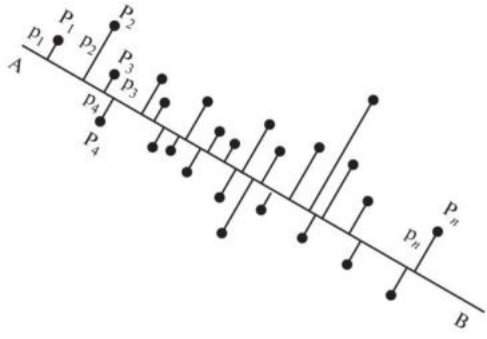

Рис. 5.11. Аппроксимация данных линейными многообразиями

4) для данной многомерной случайной величины построить такое ортогональное преобразование координат, что в результате корреляции между отдельными координатами обратятся в ноль.

Первые три версии оперируют конечными множествами данных. Они эквивалентны и не используют никакой гипотезы о статистическом порождении данных. Четвёртая версия оперирует случайными величинами. Конечные множества появляются здесь как выборки из данного распределения, а решение трёх первых задач — как приближение к «истинному» преобразованию Кархунена-Лоэва. При этом возникает дополнительный и не вполне тривиальный вопрос о точности этого приближения.

Иллюстрация к знаменитой работе К. Пирсона (1901): даны точки P_i на плоскости, P_i — расстояние от P_i до прямой AB. Ищется прямая AB, минимизирующая сумму $\sum_i P_i^2$ (рис. 5.11).

Метод главных компонент начинался с задачи наилучшей аппроксимации конечного множества точек прямыми и плоскостями (К. Пирсон, 1901). Дано конечное множество векторов $x_1, x_2, ..., x_m \in \mathbf{R}^n$. Для каждого $k = 0, 1, ..., n-1$ среди всех k-мерных линейных многообразий в $\mathbf{R}^n$ найти такое $L_k \subset \mathbf{R}^n$, что сумма квадратов уклонений x_i от L_k минимальна:

$$\sum_{i=1}^{m} \text{dist}^2(x_i, L_k) \to \min,$$

где $\text{dist}(x_i, L_k)$ — евклидово расстояние от точки до линейного многообразия.

Всякое k-мерное линейное многообразие в $\mathbf{R}^n$ может быть задано как множество линейных комбинаций $L_k = \{a_0 + \beta_1 a_1 + ... + \beta_k a_k \mid \beta_i \in \mathbf{R}\}$, где параметры β_i пробегают вещественную прямую $\mathbf{R}$, $a_0 \in \mathbf{R}^n$, $\{a_1, ..., a_k\} \subset \mathbf{R}^n$ — ортонормированный набор векторов.

Тогда

$$\mathrm{dist}^2(x_i, L_k) = \left| x_i - a_0 - \sum_{j=1}^{k} a_j (a_j, x_i - a_0) \right|^2,$$

где $|y|^2$ — евклидова норма; (a_j, x_i) — евклидово скалярное произведение.

Решение задачи аппроксимации для $k = 0, 1, ..., n-1$ даётся набором вложенных линейных многообразий $L_0 \subset L_1 \subset ... \subset L_{n-1}$, $L_k = \{a_0 + \beta_1 a_1 + ... + \beta_k a_k \mid \beta_i \in \mathbf{R}\}$. Эти линейные многообразия определяются ортонормированным набором векторов $\{a_1, ..., a_{n-1}\}$ (векторами главных компонент) и вектором a_0. Вектор a_0 ищется как решение задачи минимизации для L_0:

$$a_0 = \arg \min_{a_0 \in \mathbf{R}^n} \left(\sum_{i=1}^{m} \mathrm{dist}^2(x_i, L_0) \right),$$

т. е.

$$a_0 = \arg \min_{a_0 \in \mathbf{R}^n} \left(\sum_{i=1}^{m} |x_i - a_0|^2 \right).$$

Это — выборочное среднее:

$$a_0 = \frac{1}{m} \sum_{i=1}^{m} x_i = \overline{X}.$$

Вариационное определение среднего (как точки, минимизирующей сумму квадратов расстояний до точек данных) очень удобно для построения статистики в произвольном метрическом пространстве и сводится к обобщённому методу наименьших квадратов).

Векторы главных компонент могут быть найдены как решения однотипных задач оптимизации:

1) централизуем данные (вычитаем среднее): $x_i := x_i - \overline{X_i}$. Теперь $\sum_{i=1}^{m} x_i = 0$;

2) находим первую главную компоненту как решение задачи:

$$a_1 = \arg \min_{|a_1|=1} \left(\sum_{i=1}^{m} |x_i - a_1 (a_1, x_i)|^2 \right).$$

Если решение не единственно, то выбираем одно из них;

3) вычитаем из данных проекцию на первую главную компоненту:

$$x_i := x_i - a_1 (a_1, x_i);$$

4) находим вторую главную компоненту как решение задачи

$$a_2 = \arg\min_{|a_2|=1} \left(\sum_{i=1}^{m} |x_i - a_2(a_2, x_i)|^2 \right).$$

Если решение не единственно, то выбираем одно из них;

5) Вычитаем проекцию на $(k-1)$-ю главную компоненту (напомним, что проекции на предшествующие $(k-2)$-главные компоненты уже вычтены):

$$x_i := x_i - a_{k-1}(a_{k-1}, x_i);$$

6) находим k-ю главную компоненту как решение задачи:

$$a_k = \arg\min_{|a_k|=1} \left(\sum_{i=1}^{m} |x_i - a_k(a_k, x_i)|^2 \right).$$

Если решение не единственно, то выбираем одно из них.

На каждом подготовительном шаге $(2k-1)$ вычитаем проекцию на предшествующую главную компоненту. Найденные векторы $\{a_1, ..., a_{n-1}\}$ ортонормированы просто в результате решения описанной задачи оптимизации, однако, чтобы не дать ошибкам вычисления нарушить взаимную ортогональность векторов главных компонент, можно включать $a_k \perp \{a_1, ..., a_{k-1}t\}$ в условия задачи оптимизации.

Неединственность в определении a_k помимо тривиального произвола в выборе знака (a_k и $-a_k$ решают ту же задачу) может быть более существенной и происходить, например, из условий симметрии данных.

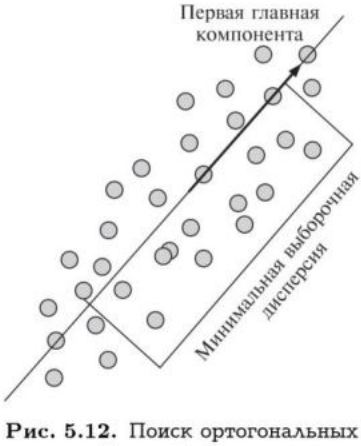

Рис. 5.12. Поиск ортогональных проекций с наибольшим рассеянием

Первая главная компонента максимизирует выборочную дисперсию проекции данных (рис. 5.12).

Пусть дан центрированный набор векторов данных $x_i \in \mathbf{R}^n$, $i = 1, ..., m$ (среднее арифметическое значение x_i равно нулю). Задача — найти такое ортогональное преобразование в новую систему координат, для которого были бы верны следующие условия:

1) выборочная дисперсия данных вдоль первой координаты максимальна (эту координату называют первой главной компонентой);

2) выборочная дисперсия данных вдоль второй координаты максимальна при условии ортогональности первой координате (вторая главная компонента);

3) выборочная дисперсия данных вдоль значений k-й координаты максимальна при условии ортогональности первым $k-1$ координатам;

Выборочная дисперсия данных вдоль направления, заданного нормированным вектором a_k, равна

$$S_m^2[(X, a_k)] = \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{j=1}^{n} x_{ij}a_{kj}\right)^2$$

(поскольку данные центрированы, выборочная дисперсия здесь совпадает со средним квадратом уклонения от нуля).

Формально, если $A = \{a_1, ..., a_n t\}^{\mathrm{T}} \in \mathbf{R}^{n\times n}$, $a_k \in \mathbf{R}^n$, — искомое преобразование, то для векторов a_k должны выполняться следующие условия:

1) находим первую компоненту как решение задачи

$$a_1 = \arg\max_{|a_1|=1} S_m^2[(X, a_1)].$$

Если решение не единственно, то выбираем одно из них;

2) вычитаем из данных проекцию на первую главную компоненту:

$$x_i := x_i - a_1(a_1, x_i);$$

в результате $x_i \perp a_1$;

3) находим вторую главную компоненту как решение задачи

$$a_2 = \arg\max_{|a_2|=1} S_m^2[(X, a_2)].$$

Если решение не единственно, то выбираем одно из них.

4) вычитаем проекцию на $(k-1)$-ю главную компоненту (напомним, что проекции на предшествующие $(k-2)$ главные компоненты уже вычтены):

$$x_i := x_i - a_{k-1}(a_{k-1}, x_i);$$

в результате $x_i \perp a_l$, $l = 1, ..., k-1$;

5) находим k-ю главную компоненту как решение задачи

$$a_n = \arg\max_{|a_k|=1} S_m^2[(X, a_k)].$$

Если решение не единственно, то выбираем одно из них.

Фактически, как и для задачи аппроксимации, на каждом шаге решается задача о первой главной компоненте для данных, из которых вычтены проекции на все ранее найденные главные компоненты. При большом числе итерации (большая размерность, много главных компонент) отклонения от ортогональности накапливаются и может потребоваться специальная коррекция алгоритма или другой алгоритм поиска собственных векторов ковариационной матрицы.

Решение задачи о наилучшей аппроксимации даёт то же множество решений $\{a_i\}$, что и поиск ортогональных проекций с наибольшим рассеянием, по очень простой причине: $|x_i - a_k(a_k, x_i)|^2 = |x_i|^2 - (a_k, x_i)^2$, и первое слагаемое не зависит от a_k. Только одно дополнение к задаче об аппроксимации: появляется последняя главная компонента a_n.

Поиск ортогональных проекций с наибольшим среднеквадратичным расстоянием между точками. Ещё одна эквивалентная формулировка следует из очевидного тождества, верного для любых m векторов x_i:

$$\frac{1}{m(m-1)}\sum_{i,j=1}^{m}(x_i - x_j)^2 = \frac{2m^2}{m(m-1)}\left[\frac{1}{m}\sum_{i=1}^{m}x_i^2 - \left(\frac{1}{m}\sum_{i=1}^{m}x_i\right)^2\right].$$

В левой части этого тождества стоит среднеквадратичное расстояние между точками, а в квадратных скобках справа — выборочная дисперсия. Таким образом, в методе главных компонент ищутся подпространства, в проекции на которые среднеквадратичное расстояние между точками максимально (или, что то же самое, его искажение в результате проекции минимально). Такая переформулировка позволяет строить обобщения с взвешиванием различных парных расстояний (а не только точек).

Аннулирование корреляций между координатами. Для заданной n-мерной случайной величины X найти такой ортонормированный базис $\{a_1, ..., a_n\}$, в котором коэффициент ковариации между различными координатами равен нулю. После преобразования к этому базису $\mathrm{cov}(X_i, X_j) = 0$ для $i \neq j$. Здесь $\mathrm{cov}(X_i, X_j) = E[(X_i - \overline{X_i})(X_j - \overline{X_j})]$ — коэффициент ковариации.

Диагонализация ковариационной матрицы. Все задачи о главных компонентах приводят к задаче диагонализации ковариационной матрицы или выборочной ковариационной матрицы. Эмпирическая, или выборочная ковариационная матрица

$$C = [c_{ij}]; \quad c_{ij} = \frac{1}{m-1}\sum_{I=1}^{m}(x_{Ii} - \overline{X_i})(x_{Ij} - \overline{X_j}).$$

Ковариационная матрица многомерной случайной величины X

$$\Sigma = [\sigma_{ij}]; \quad \sigma_{ij} = \mathrm{cov}(X_i, X_j) = E[(X_i - \overline{X_i})(X_j - \overline{X_j})].$$

Векторы главных компонент для задач о наилучшей аппроксимации и о поиске ортогональных проекций с наибольшим рассеянием — это ортонормированный набор $\{a_1, ..., a_n\}$ собственных векторов эмпирической ковариационной матрицы C, расположенных в порядке убывания собственных значений λ: $\lambda_1 \geqslant \lambda_2 \geqslant ... \geqslant \lambda_n \geqslant 0$. Эти векторы служат оценкой для собственных векторов ковариационной матрицы $\mathrm{cov}(X_i, X_j)$. В базисе из собственных векторов ковариационной матрицы она, естественно, диагональна, и в этом базисе коэффициент ковариации между различными координатами равен нулю.

Если спектр ковариационной матрицы вырожден, то выбирают произвольный ортонормированный базис собственных векторов. Он существует всегда, а собственные числа ковариационной матрицы всегда вещественны и неотрицательны.

Сингулярное разложение матрицы данных. Математическое содержание метода главных компонент — это спектральное разложение ковариационной матрицы C, т. е. представление пространства данных в виде суммы взаимно ортогональных собственных подпространств C, а самой матрицы C — в виде линейной комбинации ортогональных проекторов на эти подпространства с коэффициентами λ_i. Если $X = \{x_1, ..., x_m\}^{\mathrm{T}}$ — матрица, составленная из векторов-строк центрированных данных, то $C = \frac{1}{m-1} X^{\mathrm{T}} X$ и задача о спектральном разложении ковариационной матрицы C превращается в задачу о сингулярном разложении (Singular value decomposition) матрицы данных X.

Хотя формально задачи сингулярного разложения матрицы данных и спектрального разложения ковариационной матрицы совпадают, алгоритмы вычисления сингулярного разложения напрямую, без вычисления ковариационной матрицы и её спектра, более эффективны и устойчивы.

Теория сингулярного разложения была создана Дж.Дж. Сильвестром (J.J. Sylvester) в 1889 г. и изложена во всех подробных руководствах по теории матриц.

Матрица преобразования к главным компонентам. Матрица A преобразования данных к главным компонентам строится из векторов главных компонент: $A = \{a_1, ..., a_n\}^{\mathrm{T}}$. Здесь a_i — ортонормированные векторы-столбцы главных компонент, расположенные в порядке убывания собственных значений, верхний индекс «т»

означает транспонирование. Матрица A является ортогональной: $AA^{\mathrm{T}} = 1$.

После преобразования большая часть вариации данных будет сосредоточена в первых координатах, что даёт возможность отбросить оставшиеся и рассмотреть пространство уменьшенной размерности.

Остаточная дисперсия. Пусть данные центрированы, $\overline{X} = 0$. При замене векторов данных x_i на их проекцию на первые k главных компонент $x_i \to \sum_{j=1}^{k} a_j(a_j, x_i)$ вносится средний квадрат ошибки в расчете на один вектор данных:

$$\frac{1}{m}\sum_{i=1}^{m}\left\|x_i - \sum_{j=1}^{k} a_j(a_j, x_i)\right\|^2 = \sum_{I=k+1}^{n}\lambda_I,$$

где $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_n \geqslant 0$ — собственные значения эмпирической ковариационной матрицы C, расположенные в порядке убывания, с учетом кратности.

Эта величина называется остаточной дисперсией. Величина

$$\frac{1}{m}\sum_{i=1}^{m}\left\|\sum_{j=1}^{k} a_j(a_j, x_i)\right\|^2 = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}(a_j, x_i)^2 = \sum_{l=1}^{m}\lambda_l$$

называется объяснённой дисперсией. Их сумма равна выборочной дисперсии. Соответствующий квадрат относительной ошибки — это отношение остаточной дисперсии к выборочной дисперсии (т. е. доля необъяснённой дисперсии):

$$\delta_k^2 = \frac{\lambda_{k+1} + \lambda_{k+2} + \ldots + \lambda_n}{\lambda_1 + \lambda_2 + \ldots + \lambda_n}.$$

По относительной ошибке δ_k оценивается применимость метода главных компонент с проецированием на первые k компонент.

Замечание: в большинстве вычислительных алгоритмов собственные числа λ_i с соответствующими собственными векторами — главными компонентами a_i вычисляются в порядке от больших λ_i — к меньшим. Для вычисления δ_k достаточно вычислить первые k собственных чисел и след эмпирической ковариационной матрицы C, $\operatorname{tr} C$ (сумму диагональных элементов C, т. е. дисперсий по осям). Тогда

$$\delta_k^2 = \frac{1}{\operatorname{tr} C}\left(\operatorname{tr} C - \sum_{i=1}^{k}\lambda_i\right).$$

Специальная терминология. В статистике при использовании метода главных компонент используют несколько специальных терминов.

Матрица данных $X = \{x_1, ..., x_m\}^{\mathrm{T}}$; каждая строка — вектор предобработанных данных (центрированных и правильно нормированных), число строк равно m (количество векторов данных), число столбцов равно n (размерность пространства данных).

Матрица нагрузок (Loadings) $P = \{a_1, ..., a_k\}$; каждый столбец — вектор главных компонент, число строк — n (размерность пространства данных), число столбцов — k (количество векторов главных компонент, выбранных для проецирования).

Матрица счетов (Scores) $T = [t_{ij}]$, $t_{ij} = (x_i, a_j)$; каждая строка — проекция вектора данных на k главных компонент; число строк равно m (количество векторов данных), число столбцов равно k (количество векторов главных компонент, выбранных для проецирования).

Матрица Z-счетов (Z-scores) $Z = [z_{ij}]$; $z_{ij} = (x_i, a_j)/\sqrt{\lambda_i}$; каждая строка — проекция вектора данных на k главных компонент, нормированная на единичную выборочную дисперсию; число строк равно m (количество векторов данных), число столбцов равно k (количество векторов главных компонент, выбранных для проецирования).

Матрица ошибок (или *остатков*) (Errors or residuals) $E = X - TP^{\mathrm{T}}$.

Основная формула: $X = TP^{\mathrm{T}} + E$.

5.8.2. Сингулярный спектральный анализ

Рассмотрим вещественный временной ряд $F = (f_0 ... f_{N-1})$ длины N. Будем считать, что $N > 2$. Предположим, что ряд F — ненулевой, т. е. существует, по крайней мере, одно i, такое, что $f_i \neq 0$. Обычно считается, что $f_i = f(i\Delta)$ для некоторой функции f_t, где t — время, а Δ — некоторый временной интервал. Числа $0, ..., N-1$ могут быть интерпретированы не только как дискретные моменты времени, но и как некоторые метки, имеющие линейно упорядоченную структуру.

Здесь нумерация значения временного ряда начинается с $i = 0$, а не стандартно с $i = 1$ для удобства обозначений. Базовый алгоритм состоит из двух дополняющих друг друга этапов, разложения и восстановления.

Первый этап: разложение.

Шаг 1. Вложение. Процедура вложения переводит исходный временной ряд в последовательность многомерных векторов.

Пусть L — некоторое целое число (длина окна), $1 < L < N$. Процедура вложения образует $K = N - L + 1$ векторов вложения

$$X_i = (f_{i-1}, ..., f_{i+L-2})^{\mathrm{T}}, \quad 1 \leqslant i \leqslant K,$$

имеющих размерность L. Если нам нужно будет подчеркнуть размерность, то мы будем называть их векторами L-вложения.

L-траекторная матрица (или просто траекторная матрица) ряда F состоит из векторов вложения в качестве столбцов:

$$X = [X_1 : ... : X_k].$$

Другими словами, траекторная матрица — это матрица

$$X = (x_{ij})_{i,j=1}^{L,K} = \begin{pmatrix} f_0 & f_1 & f_2 & \cdots & f_{K-1} \\ f_1 & f_2 & f_3 & \cdots & f_K \\ f_2 & f_3 & f_4 & \cdots & f_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{L-1} & f_L & f_{L+1} & \cdots & f_{N-1} \end{pmatrix}. \tag{5.39}$$

Очевидно, что $x_{ij} = f_{i+j-2}$ и матрица X имеет одинаковые элементы на диагоналях $i + j = \text{const}$. Таким образом, траекторная матрица является ганкелевой. Существует взаимно однозначное соответствие между ганкелевыми матрицами размерности $L \times K$ и рядами длины $K = N + L - 1$.

Шаг 2. Сингулярное разложение. Результатом этого шага является сингулярное разложение траекторной матрицы ряда.

Пусть $S = XX^{\mathrm{T}}$. Обозначим $\lambda_1, ..., \lambda_L$ собственные числа матрицы S.

Пусть $d = \max\{i: \lambda_i > 0\}$. Если обозначить $V_i = X^{\mathrm{T}} U_i \sqrt{\lambda_i}$, $i = 1, ..., d$, то сингулярное разложение матрицы X может быть записано как

$$X = X_1 + ... + X_d, \tag{5.40}$$

где $X_i = \sqrt{\lambda_i} U_i V_i^{\mathrm{T}}$. Каждая из матриц X_i имеет ранг 1, поэтому их называют элементарными матрицами.

Набор $(\sqrt{\lambda_i} U_i V_i^{\mathrm{T}})$ называют i-й собственной тройкой сингулярного разложения.

Второй этап: восстановление.

Шаг 3. Группировка. На основе разложения (5.40) процедура группировки делит все множество индексов $\{1, ..., d\}$ на m непересекающихся подмножеств $I_1, ..., I_m$.

Пусть $I = \{i_1, ..., i_p\}$. Тогда результирующая матрица X_i, соот-

ветствующая группе I, определяется как

$$X_I = X_{i1} + ... + X_{ip}.$$

Такие матрицы вычисляются для $I = I_1, ..., I_m$, тем самым разложение (5.40) может быть записано в сгруппированном виде

$$X = X_{I_1} + ... + X_{I_m}. \tag{5.41}$$

Процедура выбора множеств $I_1, ..., I_m$ называется группировкой собственных троек.

Шаг 4. Диагональное усреднение. На последнем шаге базового алгоритма каждая матрица сгруппированного разложения (5.41) переводится в новый ряд длины N.

Пусть Y — некоторая $L \times K$ матрица с элементами y_{ij}, где $1 \leqslant i \leqslant L$, $1 \leqslant j \leqslant K$. Положим $L^* = \min(L, K)$, $K^* = \max(L, K)$ и $N = L + K - 1$. Пусть $y^*_{ij} = y_{ij}$, если $L \leqslant K$, и $y^*_{ij} = y_{ij}$ иначе. Диагональное усреднение переводит матрицу Y в ряд $g_0, ..., g_{N-1}$ по формуле

$$g_k = \begin{cases} \dfrac{1}{k+1} \displaystyle\sum_{m=1}^{k+1} y^*_{m,k-m+2} & \text{для } 0 \leqslant k \leqslant L^* - 1; \\ \dfrac{1}{L^*} \displaystyle\sum_{m=1}^{L^*} y^*_{m,k-m+2} & \text{для } L^* - 1 \leqslant k \leqslant K^*; \\ \dfrac{1}{N-k} \displaystyle\sum_{m=k-K+2}^{N-K^*+1} y^*_{m,k-m+2} & \text{для } K^* \leqslant k \leqslant N. \end{cases} \tag{5.42}$$

Выражение (5.42) соответствует усреднению элементов матрицы вдоль диагоналей $i + j = k + 2$: выбор $k = 0$ дает $g_0 = y_{11}$, для $k = 1$ получаем $g_1 = (y_{12} + y_{21})/2$ и т. д. Заметим, что если матрица Y является траекторной матрицей некоторого ряда $(h_0, ..., h_{N-1})$ (другими словами, если матрица Y является ганкелевой), то $g_i = h_i$ для всех i.

Применяя диагональное усреднение (5.42) к результирующим матрицам X_{Ik}, получаем ряды $F^{(k)} = (\tilde{f}^{(k)}_0, ..., \tilde{f}^{(k)}_{N-1})$, и, следовательно, исходный ряд $(f_0, ..., f_{N-1})$ раскладывается в сумму m рядов:

$$f_n = \sum_{k=1}^{m} \tilde{f}^{(k)}_n. \tag{5.43}$$

Основным параметром базового алгоритма (Singular Spectrum Analysis, SSA) является длина окна L. Выбор длины окна зависит от решаемой задачи и предварительной информации, известной о

ряде. В общем случае нет универсальных правил и безусловных рекомендаций для выбора длины окна. Однако существует несколько основных принципов для выбора длины окна L, которые имеют как теоретическое, так и практическое обоснование. Так, например, наиболее детальное разложение достигается при выборе длины окна, приблизительно равной половине длины ряда ($L \equiv N/2$). Исключением являются ряды конечного ранга, для которых при любом L, большем, чем ранг ряда d и $N > 2d-1$, число ненулевых компонент в сингулярном разложении равно d и не зависит от длины окна.

5.8.3. Метод главных компонент и обнаружение аномалий

Рассмотрим задачу сетевого обнаружения аномалий в больших распределенных системах. Метод главных компонентов (PCA) был предложен как метод для того, чтобы обнаружить аномалии, непрерывно отслеживая проекцию данных на остаточное подпространство. Этот метод доказал свою эффективность опытным путем в больших агрегированных сетях, т. е. в сетях с ограниченным количеством узлов и в грубых шкалах времени. У этого подхода, однако, есть ограничения масштабируемости. Чтобы преодолеть эти ограничения, мы разрабатываем основанный на PCA детектор аномалии, в котором адаптивные локальные фильтры данных отправляют координатору достаточное количество данных для обеспечения глобального обнаружения.

Рассмотрим систему контроля, включающую ряд локальных $M_i, ..., M_n$ узлов монитора (контролеров), каждый из которых собирает локальные наблюдения потока данных временного ряда (рис. 5.13,*а*). Например, мониторы (контролеры) могут собирать информацию о количестве TCP-запросов на установление соединения в секунду, числа транзакций в минуту DNS, или объема трафика в порту 80 за секунду. Центральный узел координатора непрерывно контролирует глобальный набор временного ряда и принимает решения относительно вопросов безопасности всей сети. Хотя эта методика применима, мы сосредотачиваемся на особенности определения объемных аномалий. Аномалия относится к необычным уровням нагрузки в сети, которые вызваны аномалиями, такими, как черви, распределенные атаки, отказ в обслуживании, отказы устройства, неверные конфигурации и так далее.

Каждый монитор собирает новую точку данных на каждом временном шаге и отправляет ее координатору. Основываясь на этих обновлениях, координатор отслеживает в скользящем окне времени размера m (т. е. m самых последних данных) для каждого временного ряда монитора, организованного в матрицу Y размера $m \times n$

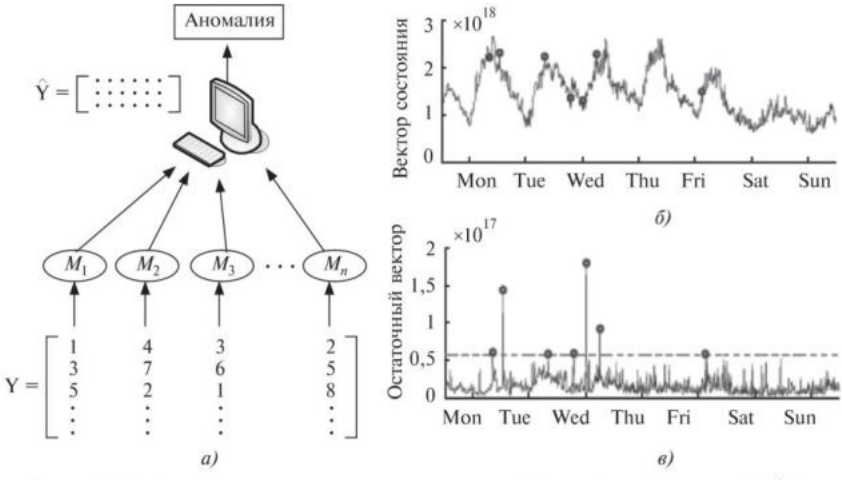

Рис. 5.13. Распределенная система контроля (*а*); выборка данных ($|y|^2$), собранная более чем за одну неделю (*б*), и его проекция в остаточном подпространстве (*в*). Пунктирная линия представляет порог для обнаружения аномалии

(где i-й столбец Y_j получает данные от монитора i, см. рис. 5.13,*а*). Затем координатор принимает свои решения исключительно на основе этой (глобальной) матрицы Y.

В алгоритме обнаружения аномалии объема всей сети [8] локальные мониторы измеряют суммарный объем трафика (в байтах) для каждого сетевого соединения и периодически (например, каждые 5 минут) централизуют данные, продвигая все недавние измерения координатору. Затем координатор выполняет метод PCA на собранной матрице Y, чтобы обнаружить аномалии объема.

Однако, чтобы гарантировать быстрое обнаружение, периоды обновления должны быть относительно маленькими. К сожалению, маленькие периоды также подразумевают увеличенные контролирующие коммуникационные издержки, которые могут быть ненужными (например, если нет никаких существенных локальных изменений в течение периода) решения.

Использование PCA для централизованного обнаружения аномалии объема. Из-за высокого уровня агрегации трафика на магистральных линиях ISP аномалии объема могут часто оставаться незамеченными, будучи «похороненными» в пределах нормального трафика (например, круговые точки, показанные в главном графике на рис. 5.13,*б*). С другой стороны хотя, результаты измерений имеют на вид высокую размерность (n — число линий), нормальные модели трафика фактически лежат в очень низкомерном

подпространстве; кроме того выделение этого нормального подпространства трафика, используя PCA (чтобы найти основные компоненты трафика), позволяет намного легче идентифицировать аномалии объема в остающемся подпространстве (рис. 5.13,*в*).

Как и прежде, Y — глобальная матрица данных временного ряда размерности $m \times n$, центрируемая для того, чтобы иметь нулевое среднее значение, и пусть $y = y(t)$ обозначает n-мерный вектор измерений (для всех линий) из одного шага по времени t. Формально, PCA — метод проекции, который отображает набор точек данных на основные компоненты, упорядоченные по объему данных, который они получают.

Набор n основных компонентов $\{V_i\}_{i=1}^{n}$, определен как

$$V_i = \arg\max_{|x|=1} \left\| \left(Y - \sum_{i=1}^{i-1} Y V_j V_i^T \right) X \right\|$$

и n — собственные векторы предполагаемой матрицы ковариации $A := \frac{1}{m} Y^{\mathrm{T}} Y$.

PCA показывает, что у места назначения источника (OD) матрицы потока магистралей ISP низкая внутренняя размерность. Таким образом, базовые нормальные потоки OD эффективно находятся в (низком) k-мерном подпространстве $\mathbf{R}^n$. Это подпространство называется нормальным подпространством трафика $S_{\text{норм}}$. Остающиеся $(n-k)$ основных компонентов составляют аварийное подпространство трафика $S_{\text{авар}}$.

Обнаружение аномалий объема полагается на разложение потока трафика $y = y(t)$ в любое время на нормальные и аварийные компоненты, $y = y_{\text{норм}} + y_{\text{авар}}$, так что $y_{\text{норм}}$ соответствует смоделированному нормальному трафику (проекция y на $S_{\text{норм}}$), а $y_{\text{авар}}$ соответствует остаточному трафику (проекция y на $S_{\text{авар}}$).

Математически $y_{\text{норм}}(t)$ и $y_{\text{авар}}(t)$ могут быть вычислены как

$$y_{\text{норм}}(t) = PP^{\mathrm{T}} y(t) = C_{\text{норм}} y(t) \text{ и } y_{\text{авар}}(t) = (I - PP^{\mathrm{T}}) y(t) = C_{\text{авар}} y(t),$$

где $P = [v_1, v_2, ..., v_k]$ сформирован первыми k основными компонентами, которые получают доминирующее различие в данных. Матрица $C_{\text{норм}} = PP^{\mathrm{T}}$ представляет собой линейный оператор, который выполняет проекцию на нормальное подпространство $S_{\text{норм}}$, и $C_{\text{авар}}$ — проекцию на аварийное подпространство $S_{\text{авар}}$.

Аномалия объема обычно приводит к большому изменению $y_{\text{авар}}$; таким образом, полезная метрика для обнаружения аварийные модели трафика является ошибкой прогноза в квадрате (SPE):

$$\text{SPE} \equiv \|y_{ab}\|^2 = \|C_{ab} y\|^2$$

(по существу, квадратная остаточная функция). Более формально предложенный алгоритм сигнализирует аномалию объема, если SPE $> Q_a$, где $Q_a = Q_a(\lambda_{k+1}, ..., \lambda_n)$ обозначает пороговую статистическую величину для остаточной функции SPE; $1 - \alpha$ — доверительный уровень.

PCA в сети для обнаружения аномалии. Перейдем к описанию детектора аномалии, который использует распределенное отслеживание и приблизительный анализ PCA. Ключевая идея состоит в том, чтобы сократить объем данных, который каждый монитор отправляет координатору. Поскольку цель работы — обнаружить аномалии, а не отследить текущее состояние, необходимо, чтобы у координатора было хорошее приближение состояния, когда аномалия рядом. То есть необязательно отслеживать глобальное состояние очень точно, когда условия нормальны. Это условие делает метод интуитивным, т. е. снижение обмена данными между мониторами и координатором становится возможным. Мы сокращаем объем данных, попадающих из мониторов координатору, устанавливая локальные фильтры в каждом мониторе. Эти фильтры поддерживают возможность локального ограничения, и монитор только отправляет координатору обновление своих данных при нарушении ограничений. Таким образом, координатор получает приблизительное, или «встревоженное» представление потока данных в каждом мониторе и, следовательно, глобального состояния. Мы используем теорию возмущений стохастической матрицы для анализа эффекта, производимого на детектор аномалии, основанный на PCA, использованием «встревоженной» глобальной матрицы. На основании вышеописанного можно выбрать параметры фильтрации (т. е. локальные ограничения), чтобы ограничить эффект возмущения на анализе PCA и на любом ухудшении в производительности детектора аномалии. Все эти идеи объединены в простой, адаптивный распределенный протокол.

5.9. Достоинства и недостатки статистических методов

Важным условием, которое является общим для всех рассмотренных методов определения изменений, является независимость наблюдений. Кроме того, наблюдения до и после изменения считаются одинаково распределенными. Данные, представляющие любые виды нестационарных и серийно коррелированных входных режимов, не выполняют эти условия. Если методы обнаружения изменений так или иначе развернуты при таких обстоятельствах, утвер-

ждения относительно вероятности ложных тревог больше не применяются.

Интернет трафик зависит от периодических изменений и серийной корреляции. Следовательно, для того чтобы применить методы определения изменений разумным способом, необходима предварительная обработка данных измерений. Более того, подозрительный трафик часто длится в течение короткого периода времени, что противоречит предположению о выходном контроле. Как следствие, короткие появления подозрительного трафика могут быть незамечены некоторыми методами определения изменений, такими, как контрольный график EWMA.

Нестационарность, исходящая из систематических временных изменений, таких, как тренд и периодические колебания, может быть смоделирована и устранена при использовании методов анализа временных рядов. Такой подход объясняется ниже. Если несколько переменных коррелированны, может помочь применение метода главных компонент. Хотя ни анализ временных рядов, ни анализ главных компонентов не могут полностью описать нормальное поведение трафика, эти методы позволяют преобразовать оригинально измеренные значения переменных трафика в новые переменные с гораздо более благоприятными статистическими свойствами.

Обнаружение зависит от того, как хорошо метод обнаружения приспособлен к специфическим изменениям. Наблюдаемые статистические свойства нуждаются в показе значительного отличия значений до и после изменения. К примеру, изменение дисперсии трудно обнаружить, если рассматривать только среднее значение образца. Кроме того, сила критериев однородности и соответствия зависит не только от величины, но также и от характеристик изменения.

Наиболее часто указываемое преимущество непараметрического CUSUM и GRSh алгоритмов есть их асимптотическая оптимальность, в результате чего минимальная задержка среднего значения обнаружения ложных тревог стремится к нулю. Однако часто пренебрегают тем, что оптимальность подходит только для определенно распределенных семейств и при предположении что величины изменений известны. Оптимальность свойств будет утрачена, если любое из этих условий не выполнится.

Системы, применяющие статистические методы, обладают целым рядом достоинств. Они не требуют постоянного обновления базы сигнатур атак, что значительно облегчает задачу сопровождения данных систем. Могут обнаруживать неизвестные атаки, сигнатуры для которых еще не написаны, что позволяет им являться

своеобразным сдерживающим буфером, пока не будет разработан соответствующий шаблон для экспертных систем. Позволяют обнаруживать более сложные атаки, чем другие методы. В частности, могут обнаруживать атаки, распределенные во времени или по объектам нападения, адаптироваться к изменению поведения пользователя и поэтому являются более чувствительными к попыткам вторжения, чем люди.

Среди недостатков систем обнаружения вторжений можно отметить трудность задания порогового значения (выбор этих значений — очень нетривиальная задача, которая требует глубоких знаний контролируемой системы). Злоумышленник может обмануть систему обнаружения атак, и она воспримет деятельность, соответствующую атаку, в качестве нормальной из-за постепенного изменения режима работы с течением времени и «приручения» системы к новому поведению. В статистических методах вероятность получения ложных сообщений об атаке является гораздо более высокой, чем при других методах. Статистические методы не очень корректно обрабатывают изменения в деятельности пользователя (например, когда менеджер исполняет обязанности подчиненного в критической ситуации). Этот недостаток может представлять большую проблему в организациях, где изменения являются частыми. В результате могут появиться как ложные сообщения об опасности, так и отрицательные ложные сообщения (пропущенные атаки). Статистические методы не способны обнаружить атаки со стороны субъектов, для которых невозможно описать шаблон типичного поведения. Системы, построенные исключительно на статистических методах, не справляются с обнаружением атак со стороны субъектов, которые с самого начала выполняют несанкционированные действия. Таким образом, шаблон обычного поведения для них будет включать только атаки. Статистические методы должны быть предварительно настроены (заданы пороговые значения для каждого параметра, для каждого пользователя); статистические методы на основе профиля нечувствительны к порядку следования событий.

Тем не менее, существуют пути решения данных проблем, и их практическая реализация является лишь вопросом времени. Очевидно, что статистический метод является чистой реализацией технологии аномального поведения. Статистический метод наследует у технологии обнаружения аномалий все так необходимые на практике достоинства.

6 ОБНАРУЖЕНИЕ АНОМАЛЬНЫХ ВЫБРОСОВ ТРАФИКА МЕТОДОМ КРАТНОМАСШТАБНОГО АНАЛИЗА

Метод кратномасштабного анализа (КМА) предполагает представление функций в различных масштабах, т. е. при различных разрешениях. Преимущество такого подхода очевидно — характерные детали, которые могут оставаться незамеченными при одном разрешении, могут быть обнаружены на другом.

6.1. Основы теории вейвлетов

Теория вейвлетов является мощной альтернативой классическому анализу Фурье и дает более гибкую технику обработки сигналов. Английское слово *wavelet* (от французского *ondelette*) можно перевести как «маленькая (короткая) волна» или «всплеск» [20]. Сам термин был введен Гроссманном (Crossmann) и Морле (Morlet) в середине 80-х г. XX века в связи с анализом свойств сейсмических и акустических сигналов. Их работа послужила началом интенсивного развития вейвлетов в последующее десятилетие рядом исследователей: Добеши (Dodechies), Мейер (Meyer), Малл (Mallat), Фарж (Farge), Чуи (Chui) и др. [26].

Вейвлеты представляют собой особые функции в виде коротких волн (всплесков) с нулевым средним значением, локализованные по оси аргументов, инвариантные к сдвигу и линейные к операции масштабирования (сжатия/растяжения). Любой из наиболее часто используемых типов вейвлетов порождает полную ортогональную систему.

Теория вейвлетов не является фундаментальной физической теорией, но она дает удобный и эффективный инструмент для решения многих практических задач. Основная область применения вейвлетных преобразований — анализ и обработка сигналов и функций, нестационарных во времени или неоднородных в пространстве, когда результаты анализа должны содержать не только частотную характеристику сигнала (распределение энергии сигнала по частотным составляющим), но и сведения о локальных координатах, на

которых проявляют себя те или иные группы частотных составляющих или на которых происходят быстрые изменения частотных составляющих сигнала.

По сравнению с разложением сигналов на ряды Фурье вейвлеты способны с гораздо более высокой точностью представлять локальные особенности сигналов, вплоть до разрывов 1-го рода (скачков). В отличие от преобразований Фурье вейвлет-преобразование одномерных сигналов обеспечивает двумерную развертку, при этом частота и координата рассматриваются как независимые переменные, что дает возможность анализа сигналов сразу в двух пространствах.

Одна из главных и особенно плодотворных идей вейвлетного представления сигналов на различных уровнях декомпозиции (разложения) заключается в разделении функций приближения к сигналу на две группы: аппроксимирующую — грубую, с достаточно медленной временной динамикой изменений и детализирующую — с локальной и быстрой динамикой изменений на фоне плавной динамики, с последующим их дроблением и детализацией на других уровнях декомпозиции сигналов. Это возможно как во временной, так и в частотной областях представления сигналов вейвлетами.

В основном вейвлет-преобразования делят на две группы: непрерывное (CWT, НВП) и дискретное (DWT, ДВП), в каждой из которых существует несколько разновидностей. Рассмотрим основные из них, которые были использованы в этой работе.

6.2. Непрерывное вейвлет-преобразование

Для понимания непрерывного вейвлет-преобразования нужно ввести понятие *свертки*. *Сверткой* функций f и g (записывается $f * g$) называется интегральное преобразование вида

$$(f * g)(t) = \int_{-\infty}^{\infty} f(u)g(t-u)\,du, \tag{6.1}$$

т. е. это интеграл от произведения двух функций после того, как одна реверсируется и смещается. Свёртка показывает «схожесть» одной функции с отражённой и сдвинутой копией другой.

Вейвлет-преобразование (ВП) одномерного сигнала в общем виде — это его представление в виде обобщенного ряда или интеграла Фурье по системе базисных функций

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right), \tag{6.2}$$

сконструированных из обладающего определенными свойствами материнского (исходного) вейвлета за счет операций сдвига во времени

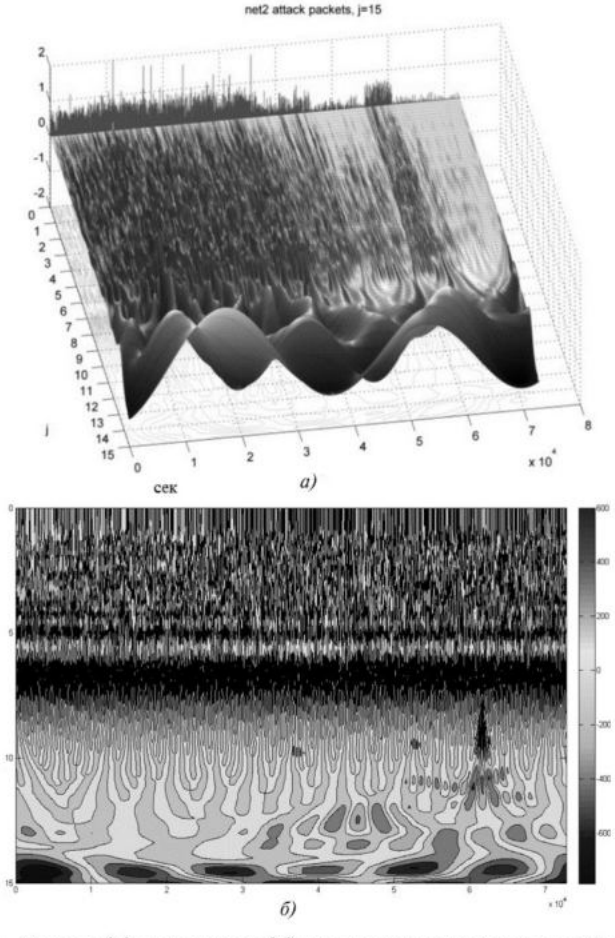

Рис. 6.1. Спектр (*а*) и изолинии (*б*) непрерывного вейвлет-преобразования реализации с аномалиями

(u) и изменения временного масштаба (s). Множитель $1/\sqrt{s}$ обеспечивает независимость нормы этих функций от масштабирующего числа s. Тогда непрерывное вейвлет-преобразование

$$W_f(u,s) = (f(t), \psi_{u,s}(t)) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} f(t)\psi\left(\frac{t-u}{s}\right) dt, \qquad (6.3)$$

где $f(t)$ — исходный сигнал; $\psi(t)$ — материнский вейвлет, смещенный по времени на величину переноса u и увеличенный/уменьшенный на величину масштаба s. Малые значения s соответствуют мелкому масштабу вейвлета или высоким частотам, большие параметры

s — крупному масштабу, т. е. растяжению материнского вейвлета и сжатию его спектра.

Исходя из выражений (6.1) и (6.2) можно записать *непрерывное вейвлет-преобразование* как результат *свертки* [26]:

$$W_f(u,s) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} f(t)\psi\left(\frac{t-u}{s}\right) dt = f(t) * \overline{\psi}_S(u), \qquad (6.4)$$

где

$$\overline{\psi}_S(t) = \frac{1}{\sqrt{s}}\psi(\frac{-t}{s}).$$

Графически результат выполнения непрерывного вейвлет-преобразования изображается в виде спектра, который может быть представлен как в виде плоской картинки, так и в виде объемной поверхности (рис. 6.1,*а*).

На рис. 6.1,*а* изображен спектр непрерывного вейвлет-преобразования для реализации сетевого трафика с аномалиями размером 72700 значений. Декомпозиция выполнена при помощи вейвлета Морле по 15 уровням. На рис. 6.1,*б* изображены изолинии спектра вейвлет-преобразования. На обоих рисунках хорошо видно, что типичная DDoS-атака, проявляющаяся в виде аномального выброса в районе $6 \cdot 10^4$, может быть обнаружена при помощи вейвлет-преобразования на уровне характерных для нее частот (0...12), т. е. на уровнях разложения, начиная примерно с 12, где размер вейвлета соизмерим или больше, чем продолжительность самой аномалии, она перестает вносить существенные изменения в спектр преобразования.

6.3. Дискретное вейвлет-преобразование. Алгоритм Малла

При непрерывном изменении параметров s и u для расчета вейвлет-спектра необходимы большие вычислительные затраты. Множество функций $\psi_{us}(t)$ избыточно. Необходима дискретизация этих параметров при сохранении возможности восстановления сигнала из его преобразования. Дискретизация, как правило, осуществляется через степени двойки:

$$\begin{aligned} &s = 2^j; \quad u = k2^j; \\ &\psi_{j,k}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right) = \frac{1}{\sqrt{2^j}}\psi(2^{-j}t - k), \end{aligned} \qquad (6.5)$$

где j и k — целые числа. В этом случае плоскость u, s превращается в соответствующую сетку j, k. Параметр j называется *параметром*

масштаба, или уровнем декомпозиции, а вейвлет-преобразование, выполненное с таким параметром масштаба, называется *диадным*.

Наиболее быстрым и часто используемым дискретным вейвлет-преобразованием является быстрое вейвлет-преобразование (ВВП), или алгоритм Малла [26]. Он позволяет представить сигнал в виде совокупности последовательных приближений грубой (аппроксимирующей) $A_j(t)$ и уточненной (детализирующей) $D_j(t)$ составляющих:

$$S(t) = A_j(t) + \sum_{j=1}^{m} D_j(t), \tag{6.6}$$

с последующим их уточнением итерационным методом. Каждый шаг уточнения соответствует определенному масштабу 2^j (т. е. уровню j) анализа (декомпозиции) и синтеза (реконструкции) сигнала. Такое представление каждой составляющей сигнала вейвлетами можно рассматривать как во временной, так и в частотной областях.

На первом шаге алгоритма исходный сигнал $S(t)$ декомпозируется на две составляющие:

$$S(t) = A_1(t) + D_1(t) = \sum_k a_1\varphi_1(t) + \sum_k d_1\psi_1(t),$$

где $\psi_{1k}(t)$ — вейвлет; $\varphi_{1k}(t)$ — порождающая функция вейвлета; a_1, d_1 — коэффициенты аппроксимации и детализации на уровне 1 соответственно.

Число коэффициентов a_{1k} и d_{1k} имеют половинную длину по сравнению с исходным сигналом. Следующий шаг итерации для уровня два выполняется с полученной на уровне 1 аппроксимаций аналогичным способом. На практике наибольший уровень разложения определяется числом $n_0 - 1$ дискретных значений сигнала ($N = 2^{n_0}$).

Таким образом, на каждом уровне разложения j будем иметь последовательности коэффициентов аппроксимации a_j и детализации d_j длиной $N/2^j$ каждая, а исходный сигнал можно восстановить из выражения

$$S(t) = a_j(t)\varphi(t)\sum_{j=1}^{m} d_j(t) + \psi_1(t). \tag{6.7}$$

Число операций умножения при прямом ВВП равно $2LN$, где $L = 2n$. Столько же операций необходимо и для реконструкции сигнала. Таким образом, для анализа-синтеза сигнала в базисе вейвлетов необходимо выполнить $4LN$ операций, что не превышает (и

даже меньше) числа операций для быстрого преобразования Фурье $(N \log_2 N)$.

Благодаря свойствам вейвлетов для сформированного из них ортогонального базиса можно записать исходную функцию сигнала $X(t)$ в виде

$$X(t) = \sum_k a_X(J,k)\varphi_{J,k}(t) + \sum_{j=1}^{J}\sum_k d_X(j,k)\psi_{j,k}(t), \qquad (6.8)$$

где первая сумма представляет собой приближения (среднее) для функции сигнала, сумма по j — шкалу J; сумма по k — отклонения между полученными приближениями; $d_X(j,k)$ — вейвлет-коэффициенты.

Применительно к объекту исследования модель (6.8) после КМА примет вид:

$$s(t_i) = \sum_{k=-\infty}^{\infty} a_{m,k}\varphi_{m,k}(t_i) + \sum_{m=m'}^{\infty}\sum_{k=-\infty}^{\infty} d_{m,k}\psi_{m,k}(t_i), \quad m,k \in I, \qquad (6.9)$$

где $\varphi_{m,k}(t)$ — базисная масштабирующая функция; $\psi_{m,k}(t)$ — базисная вейвлет-функция; $a_{m,k}, d_{m,k}$ — аппроксимирующие и детализирующие коэффициенты; m,k — параметры масштаба и сдвига в пространстве целых чисел I.

При конечном числе уровней разложения M любую последовательность дискретных отсчетов радиосигнала $s(t_i)$ можно представить в виде упорядоченной совокупности коэффициентов разложения по системе масштабирующих функций и вейвлет-функций:

$$s(t_i) = \sum_{k=1}^{2^{N-M}} a_{m,k}\varphi_{m,k}(t_i) + \sum_{m=1}^{M}\sum_{k=1}^{2^{N-M}} d_{m,k}\psi_{m,k}(t_i).$$

Масштабирующие и вейвлет-функции определяются в соответствии с теорией кратномасштабного анализа (КМА) [5]:

$$\varphi_{m,k}(t) = \sqrt{2^m}\varphi(2^m t - k); \quad \psi_{m,k}(t) = \sqrt{2^m}\psi(2^m t - k).$$

Здесь $\sqrt{2^m}$ — нормирующий множитель; $k = 0, \pm 1, \pm 2, ...;\ m \in Z$.

а практике для быстрого расчета значений коэффициентов $a_{m,k}$ и $d_{m,k}$ применяют схему последовательного деления, называемую пирамидой, или алгоритмом Малла, которая интерпретируется как последовательная двухполосная фильтрация входного сигнала при помощи каскадно соединенных блоков фильтров низкой (H) и высокой (G) частот (рис. 6.2).

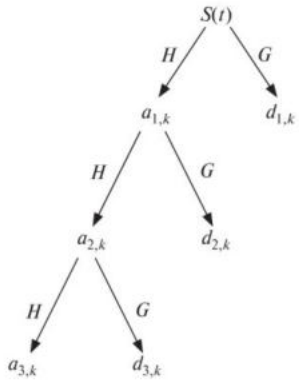

Рис. 6.2. Схема преобразований сигнала по алгоритму Малла

На рис. 6.2 для вейвлет-коэффициентов $a_{m,k}$ и $d_{m,k}$ первый индекс m соответствует номеру уровня разложения, а второй индекс $k = 0, 1, ..., 2^m - 1$ — порядковому значению вейвлет-коэффициента на уровне разложения m.

Согласно теории КМА значения $a_{m,k}$ и $d_{m,k}$ можно получить, базируясь на коэффициентах, рассчитанных на предыдущих этапах декомпозиции сигнала:

$$a_{m,k} = \frac{1}{\sqrt{2}} \sum_n a_{m-1,n} h_{n+2k};$$

$$d_{m,k} = \frac{1}{\sqrt{2}} \sum_n a_{m-1,n} g_{n+2k},$$

где h_m и g_m — последовательности, определяющие характеристики фильтров H и G на m-м уровне разложения.

Первая сумма в (6.9)

$$q_{\text{т}}(t_i) + q_{\text{ц}}(t_i) = \sum_{k=-\infty}^{\infty} a_{m,k} \varphi_{m,k}(t_i)$$

содержит усредненные (с весовыми функциями $a_{m,k}$) значения $f(t_i)$ по диадным интервалам $[2k - m, 2(k + 1) - m]$ характеризует тренд и циклические составляющие трафика (суточные и недельные), а вторая

$$\varepsilon_{\text{а}}(t_i) + \varepsilon_{\text{ф}}(t_i) = \sum_{m=m'}^{\infty} \sum_{k=-\infty}^{\infty} d_{m,k} \psi_{m,k}(t_i)$$

— локальные особенности сетевого трафика на фоне случайной шумовой составляющей (флуктуаций).

Применение вейвлет-функций дает возможность анализа частотно-временного представления сигнала, а также разделения его на высокочастотные (ВЧ-детали) и низкочастотные компоненты (НЧ-приближения), что обеспечивает возможность локализации аномалий сигнала различных видов.

Критерии выбора вейвлет-функции. В задачах выделения и идентификации локальных особенностей временного ряда определяющими при выборе вейвлет-функции будут следующие характеристики:

- гладкость;

- размер носителя;
- число нулевых моментов.

Число нулевых моментов. Вейвлет Y имеет m нулевых моментов, т. е.

$$\int_{-\infty}^{+\infty} t^k \psi(t)\, dt = 0, \quad k = \overline{0, m-1}. \tag{6.10}$$

Число нулевых моментов m характеризует способность вейвлета выявлять особенность вида $a < m$. Также из свойства (6.10) следует, что Y ортогонален любому многочлену степени $m-1$, т. е. «пропускает» полиномы степени выше $m-1$. Поэтому в окрестностях, где f гладкая функция, большое число нулевых моментов Y обеспечит большое число пренебрежимо малых вейвлет-коэффициентов. В этом случае мы получаем аппроксимирующую схему с минимальным числом слагаемых при заданном уровне точности аппроксимации.

Размер носителя. Вейвлет-преобразование порождает искусственные «скачки» на краях функции f, находящие отражения в коэффициентах разложения. Поскольку размер окрестности на масштабном уровне j, содержащей возникающий при этом краевой эффект, зависит от размера носителя функции Y и определяется по формуле

$$h_j = 2^j q,$$

где q — размер носителя используемой базисной функции. Получаем, чем меньше размер носителя, тем меньшую погрешность мы имеем на краях f.

Если f имеет локальную особенность в некоторой точке n и n находится внутри носителя $\psi_{j,n}$, то вейвлет-коэффициенты имеют большую амплитуду. Чем больше носитель, тем большее число таких коэффициентов на каждом масштабном уровне мы получаем. Чтобы минимизировать их число, мы должны использовать функции с наименьшим размером носителя Y.

Гладкость вейвлета. Аналогично числу нулевых моментов, гладкость вейвлета характеризует его способность выявлять особенность вида $\alpha \leqslant m$. Что касается погрешности аппроксимации, то гладкость вейвлета Y оказывает в основном косметическое влияние на нее: если Y — гладкий вейвлет, то и погрешность будет гладкая.

Таким образом, при выборе вейвлет-функции мы приходим к выбору между числом нулевых моментов и размером носителя. Для ортогональных вейвлет-функций доказано, что если вейвлет Y имеет m нулевых моментов, то его наименьший носитель равен $2m-1$. Поэтому:

- если функция f имеет несколько локальных особенностей и очень гладкая между этими особенностями, необходимо использовать вейвлет с большим числом нулевых моментов;
- если число особенностей нарастает, лучше уменьшить размер носителя ценой уменьшения числа нулевых моментов.

Доказано также, что ортогональные вейвлеты Добеши с компактным носителем — это единственное семейство базисных вейвлет-функций, которые имеют минимальный размер носителя при заданном числе нулевых моментов.

Известным фактом является то, что число нулевых моментов и гладкость вейвлетов связаны друг с другом, но характер связи может быть различным в зависимости от вида рассматриваемого семейства базисов. В частности, показано, что, например, для фильтров Добеши характерно следующее свойство: гладкость вейвлета возрастает с возрастанием числа нулевых моментов.

В большинстве практических задач необходимо, чтобы крупномасштабные аппроксимирующие компоненты позволяли получить наилучшее приближение функции, а с помощью мелкомасштабных составляющих идентифицировались отдельные локальные особенности.

Наилучшее приближение аппроксимируемой функции f в крупномасштабных компонентах обеспечивает скэйлинг-функция f [26] с достаточным числом нулевых моментов m. Поэтому в этом случае важно, чтобы нулевые моменты имел не только Y, но и f. Семейство ортогональных базисов, удовлетворяющее этому требованию и имеющее носитель наименьшего размера, называют койфлетами.

Койфлеты порядка m удовлетворяют условиям:

$$\int_{-\infty}^{+\infty} t^k \psi(t)\, dt = 0, \quad k = \overline{0, m};$$

$$\int_{-\infty}^{+\infty} \varphi(t)\, dt = 1; \quad \int_{-\infty}^{+\infty} t^k \varphi(t)\, dt = 0 \quad k = \overline{1, m};$$

Если функция f принадлежит C^k в окрестности $2^J n$ с $k \leqslant m$, то

$$2^{J/2} \langle f, \varphi_{J,n} \rangle \approx f(2^J n) + (2^{(k+1)J}). \tag{6.11}$$

Порядок приближения возрастает с ростом m, результирующий койфлет имеет носитель размера $3m - 1$ вместо $2m - 1$ для вейвлета Добеши.

Преимущества и применение. Вейвлет-функции с компактными носителями, например вейвлеты Добеши и койфлеты, наиболее качественно выделяют локальные особенности сигналов.

Койфлеты более симметричны чем, например, вейвлеты Добеши, что дает лучшую аппроксимацию при изучении симметричных сигналов.

Наличие у койфлетов нулевых моментов скейлинг-функции приводит к лучшей сжимаемости.

Вейвлеты Добеши и койфлеты индуцируются общей 2π-периодической функцией $m_0(\omega) \in L_2(0, 2\pi)$, но для койфлетов к ней добавляется набор условий, определяющих равенство нулю моментов соответствующей скейлинг-функции.

6.4. Анализ методов обнаружения аномалий трафика с помощью вейвлетов

Техника вейвлет-анализа широко используется в системах обнаружения вторжений, благодаря присущему ей частотно-временному свойству, которое позволяет раскладывать сигнал на несколько частотных компонент. Уже достаточно много работ было опубликовано на эту тему и много систем внедрено на практике.

Показательной в этом плане можно считать работу Barford [55]. В качестве анализируемого трафика в ней был взят трафик реальной сети, записанный в течение 6 месяцев на пограничном маршрутизаторе. После сбора и анализа трафика, авторы выделяют четыре типа возникших в реальной сети аномалий:

- проблемы оборудования: отказ оборудования или временная неверная настройка оборудования, отключения;
- атаки: в основном DDoS, обычно типа flood;
- перегрузки: перегрузки на сети, например увеличение величины исходящего трафика ftp-сервера вследствие появления на нем сверхпопулярного контента;
- прочие аномалии, которые не относятся ни к проблемам на сети, ни к атакам и перегрузкам. Например: компьютер в общежитии, обменивающийся круглосуточно большими объемами информации с ресурсом в интернете, как часть исследовательского проекта. Также проблемы, связанные с ошибкой записи трафика, например пропуск некоторых потоков в записи из-за перегрузки маршрутизатора.

Примечательно, что предложенный метод не рассматривается как готовая полноценная система для онлайн обнаружения аномалий, но некоторые идеи и методы, предложенные в ней, использовались во многих работах и в дальнейшем.

Ключевой идеей в работе является выделение из данного сигнала x (который представляет собой средние 5-минутные значения) трех сигналов следующим образом:

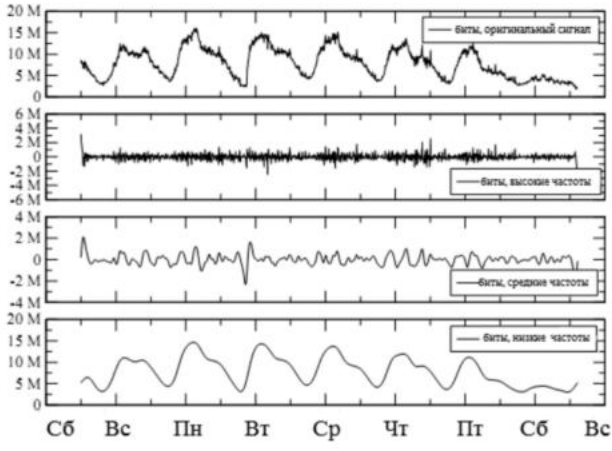

Рис. 6.3. Исходный трафик, записанный в течение недели (вверху), и его частотные составляющие, полученные при помощи вейвлет-декомпозиции (сверху-вниз: ВЧ, СЧ и НЧ)

НЧ-часть исходного сигнала получена реконструкцией всех НЧ-вейвлет-коэффициентов, начиная с уровня 9 и выше. НЧ-часть сигнала должна захватывать особенности и аномалии очень высокой продолжительности (от нескольких дней и больше). Сигнал здесь очень рассеянный (число элементов данных примерно 0,4 % от исходного сигнала) и захватывает недельные особенности очень хорошо. Для многих других типов данных Интернета НЧ-часть сигнала показывает большую степень регулярности трафика, что может помочь захватывать аномалии большой продолжительности.

СЧ-часть сигнала получена реконструкцией вейвлет-коэффициентов частотных уровней 6, 7 и 8. Полученный здесь сигнал имеет нулевое среднее и предназначен для захвата в основном дневных колебаний сигнала. Число элементов данных здесь около 3 % от исходного сигнала.

ВЧ-часть сигнала получена пороговой обработкой вейвлет-коэффициентов первых 5 частотных уровней, т. е. все коэффициенты, чье абсолютное значение *ниже* определенного порога, обнуляются (жесткий порог). Необходимость в пороговой обработке исходит из того факта, что ВЧ-часть состоит из небольших краткосрочных изменений, которые мы в общем считаем *шумом*, что никак не помогает нам в объективном определении аномалий.

Выделение этих трех компонент показано на рис. 6.3.

После выделения трех частотных составляющих сигнала авторы вычисляют локальную дисперсию для каждой из этих состав-

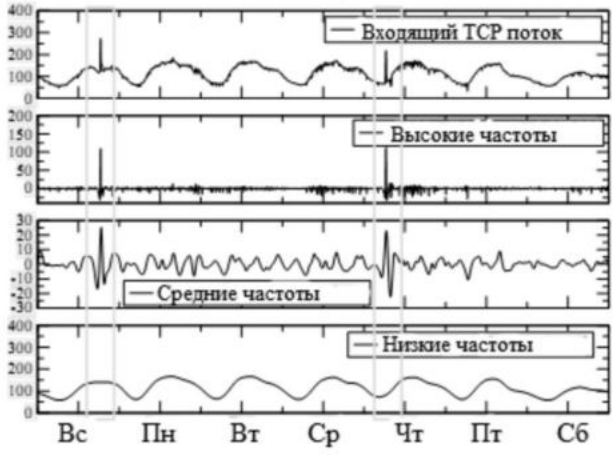

Рис. 6.4. Исходный трафик с DOS-атакой и его вейвлет-декомпозиция по трем составляющим. Атака выделена серыми вертикальными полосами

ляющих при помощи скользящего окна, получая на выходе график изменения дисперсии. Далее, применяя пороговый анализ к этому графику, по превышению порогов принимается решение о наличии или отсутствии аномалии. Результатом вейвлет-анализа по методу Барфорда явился важный вывод о том, что различные типы аномалий могут быть обнаружены на различных, присущих только им, частотных уровнях вейвлет-разложения.

Например, SYN-flood DOS-атака, представляющая собой кратковременную высокочастотную аномалию, может быть обнаружена только на ВЧ и СЧ-частях, в то время как на крупнозернистой НЧ-составляющей она не видна, что иллюстрируется на рис. 6.4.

Исследования показывают, что аномалии сетевого трафика можно разделить на два больших класса — кратковременные (напри-

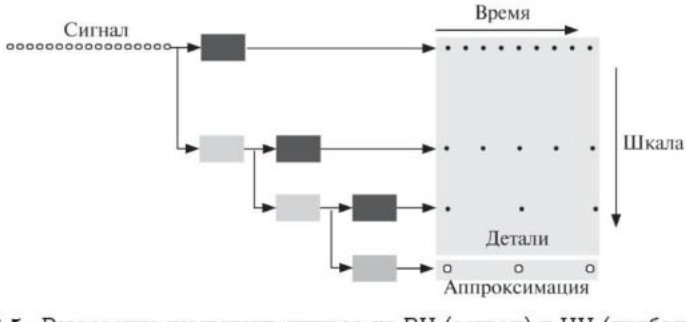

Рис. 6.5. Разделение компонент сигнала на ВЧ (детали) и НЧ (приближения) компоненты

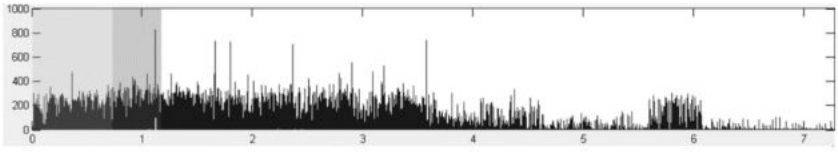

Рис. 6.6. Скользящие окна в потоке трафика

мер, DDoS-атаки) и длительные (например, сканирование портов — portscan). Инструментарий вейвлет-анализа позволяет выделять данные аномалии путем разделения трафика на ВЧ и НЧ компоненты, как показано на рис. 6.5.

6.5. Алгоритм обнаружения аномалий методом дискретного вейвлет-преобразования

Рассмотрим обнаружение аномалий сетевого трафика на основе дискретного вейвлет-преобразования с применением статистических критериев [27, 28]. Для адаптации этого способа к анализу трафика в реальном времени используется техника двух скользящих окон W_1 и W_2, двигающихся во времени с определенным шагом, фиксируя значения трафика, которые находятся во временных границах каждого окна.

Применение «скользящих окон» позволяет увеличить надежность обнаружения даже незначительных аномалий. Известно, что плотность спектральной мощности временного ряда «трафик — время» при наличии аномалий имеет пики на определенных частотах. Вейвлет-анализ позволяет обнаружить аномалии трафика на основании различий спектров обычного и аномального трафика. На рис. 6.6 изображены реализация трафика и два окна. Будем считать окно W_1 «окном сравнения», а окно W_2 — «окном обнаружения». Пусть размер каждого окна w_1 и w_2 выбранных временных единиц соответственно, причем $w_1 > w_2$. Тогда в произвольный момент времени t начало окна W_2 будет находиться в точке t, в нем будут содержаться w_2 значений трафика от $t - w_2$ до t, а в окне W_1 — w_1 значений от $t - w_2 - w_1$ до $t - w_2$.

Выполняя БВП для выборок внутри каждого из окон в каждый момент времени t_i, будем получать на некотором масштабном уровне j набор коэффициентов для окна W_1 — аппроксимации $\{a_{1x}, a_{2x}, a_{3x}, ..., a_{nx}\}_{t,j}$ и детализации $\{d_{1x}, d_{2x}, d_{3x}, ..., d_{nx}\}_{t,j}$; для окна W_2 — аппроксимации $\{a_{1y}, a_{2y}, a_{3y}, ..., a_{my}\}_{t,j}$ и детализации $\{d_{1y}, d_{2y}, d_{3y}, ..., d_{my}\}_{t,j}$. Причем количество коэффициентов n на уровне j в окне W_1 будет определяться выражением $n = w_1/2^J$, в окне W_2 — выражением $m = w_2/2^J$. Эти коэффициенты будут проверяться по

статистическим критериям, и на основе принятия или отклонения статистических гипотез будет выноситься решение о кардинальном различии в анализируемых параметрах между окнами W_1 и W_2, а следовательно, наличии аномалий или же наоборот — их отсутствии.

Анализ статистических характеристик коэффициентов аппроксимации и детализации показывает, что плотность распределения вероятностей (ПРВ) мгновенных значений коэффициентов детализации хорошо описывается гауссовским законом с параметрами $N(0, Md)$, а ПРВ аппроксимирующих коэффициентов хорошо описывается распределением

$$f(x) = \frac{\lambda p}{2\Gamma(1/p)} \mathrm{e}^{-\lambda|x-m|^p},$$

где $0 \leqslant p < \infty$ — параметр формы; λ — дисперсия распределения; m — среднее значение и

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} \mathrm{e}^{-t}\, dt, \quad z > 0.$$

Параметр формы p определяет вид распределения. При различных p распределение имеет следующий вид: $p = 0$ — дельта-функция Дирака; $p = 1$ — распределение Лапласа; $p = 2$ — гауссовское распределение; $p = +\infty$ — равномерное распределение.

Корреляцией коэффициентов детализации и аппроксимации можно пренебречь в силу ортогональности вейвлет-базиса.

Опишем алгоритм обнаружения аномальных выбросов на основе статистических критериев, используемых для определения изменений дисперсии и среднего в коэффициентах вейвлет-преобразования.

Для обнаружения аномалий, выражающихся в изменении дисперсии предлагается использовать критерий Фишера, обнаружения изменении величины среднего значения — критерий Кохрана [56].

6.5.1. Алгоритм обнаружения аномалий по критерию Фишера для выбросов дисперсий

Критерий Фишера предложен для обнаружения изменений в дисперсиях выборок окон W_1 и W_2. Распределение выборок считается гауссовким. В каждый момент времени (положении окон) t на масштабном уровне j выдвигаются две статистические гипотезы о равенстве дисперсий двух выборок $\{d_{1x}, d_{2x}, d_{3x}, ..., d_{nx}\}_{t,j}$ и $\{d_{1y}, d_{2y}, d_{3y}, ..., d_{my}\}_{t,j}$: нулевая H_0: $\sigma^2_{1,t,j} = \sigma^2_{2,t,j}$ и альтернативная H_1: $\sigma^2_{1,t,j} \neq \sigma^2_{2,t,j}$.

Алгоритм обнаружения выбросов в гауссовском процессе на основе анализа аномального изменения дисперсий записывается как

$$z_{t,j} = \frac{\frac{1}{m-1}\sum_{i=1}^{m}(d_{iy} - \overline{d_y})^2}{\frac{1}{n-1}\sum_{i=1}^{n}(d_{ix} - \overline{d_x})^2}. \tag{6.12}$$

Введем обозначения:

$S^2_{1,t,j} = \frac{1}{n-1}\sum_{i=1}^{n}(d_{ix} - \overline{d_x})^2$ — выборочная дисперсия выборки последовательности деталей на масштабном уровне j в окне W_1;

$S^2_{2,t,j} = \frac{1}{m-1}\sum_{i=1}^{m}(d_{iy} - \overline{d_y})^2$ — выборочная дисперсия выборки последовательности деталей на масштабном уровне j в окне W_2;

$\overline{d}_x = \frac{1}{n}\sum_{i=1}^{n} d_{ix}$ — выборочное среднее выборок последовательности деталей на масштабном уровне j в окне W_1;

$\overline{d}_y = \frac{1}{m}\sum_{i=1}^{m} d_{iy}$ — выборочное среднее выборок последовательности деталей на масштабном уровне j в окне W_2.

С учетом сделанных обозначений перепишем алгоритм (6.12) в виде

$$Z_{t,j} = \frac{S^2_{2,t,j}}{S^2_{1,t,j}}. \tag{6.13}$$

Нулевая гипотеза опровергается в пользу альтернативной, в случае если $Z > F_p(v_1, v_2)$, где $F_p(v_1, v_2)$ — p-квантиль распределения Фишера с $v_1 = n - 1$ и $v_2 = m - 1$ степенями свободы.

Случайная величина, определяемая отношением (6.13), имеет распределение Фишера–Снедекора [4].

6.5.2. Алгоритм обнаружения аномалий на основе критерия Кохрана-Кокса

Критерий Кохрана предложен для обнаружения изменений среднего значения выборок аппроксимаций $\{a_{1x}, a_{2x}, a_{3x}, ..., a_{nx}\}_{t,j}$ и $\{a_{1y}, a_{2y}, a_{3y}, ..., a_{my}\}_{t,j}$.

Алгоритм обнаружения выбросов на основе анализа аномально-

го изменения среднего значения выборки записывается как

$$Y = \frac{1}{s}\left(\frac{1}{n}\sum_{i=1}^{n} a_{ix} - \frac{1}{m}\sum_{i=1}^{m} a_{iy}\right). \tag{6.14}$$

Введем, как и прежде, обозначения:

$S^2_{1,t,j} = \dfrac{1}{n-1}\sum_{i=1}^{n}(a_{ix} - \overline{a_x})^2$ — выборочная дисперсия выборки последовательности аппроксимаций на масштабном уровне j в окне W_1;

$S^2_{2,t,j} = \dfrac{1}{m-1}\sum_{i=1}^{m}(a_{iy} - \overline{a_y})^2$ — выборочная дисперсия выборки последовательности аппроксимаций на масштабном уровне j в окне W_2;

$S^2_{t,j} = \dfrac{S^2_{1,t,j}}{n} + \dfrac{S^2_{2,t,j}}{m}$ — суммарная взвешенная дисперсия выборок деталей в окнах W_1 и W_2;

$\overline{a}_x = \dfrac{1}{n}\sum_{i=1}^{n} a_{ix}$ и $\overline{a}_y = \dfrac{1}{m}\sum_{i=1}^{m} a_{iy}$ — выборочные средние выборок последовательностей деталей на масштабном уровне j в окне W_1 и W_2 соответственно.

С учетом введенных обозначений алгоритм (6.14) может быть преобразован к виду

$$Y_{t,j} = \frac{1}{S_{t,j}}(\overline{a_y} - \overline{a_x}). \tag{6.15}$$

Критические (пороговые) значения статистики вычисляются по формуле:

$$t'_p = \frac{f_1 t_p(v_1) + f_2 t_p(v_2)}{f_1 + f_2},$$

где $f_1 = S_1^2/n$, $f_2 = S_2^2/m$; $t_p(v)$ — p-квантиль распределения Стьюдента с v степенями свободы ($v_1 = n-1$ и $v_2 = m-1$).

6.5.3. Алгоритм обнаружения аномалий по критерию Фишера для выбросов средних значений

В случае, если статистика анализируемых последовательностей имеет распределение экспоненциального вида для обнаружения аномальных выбросов, в средних значениях выборок может быть использована критерий Фишера для выбросов средних значений. Рассмотрим применение данного критерия для обнаружения аномальных выбросов среднего значения коэффициентов аппроксимации

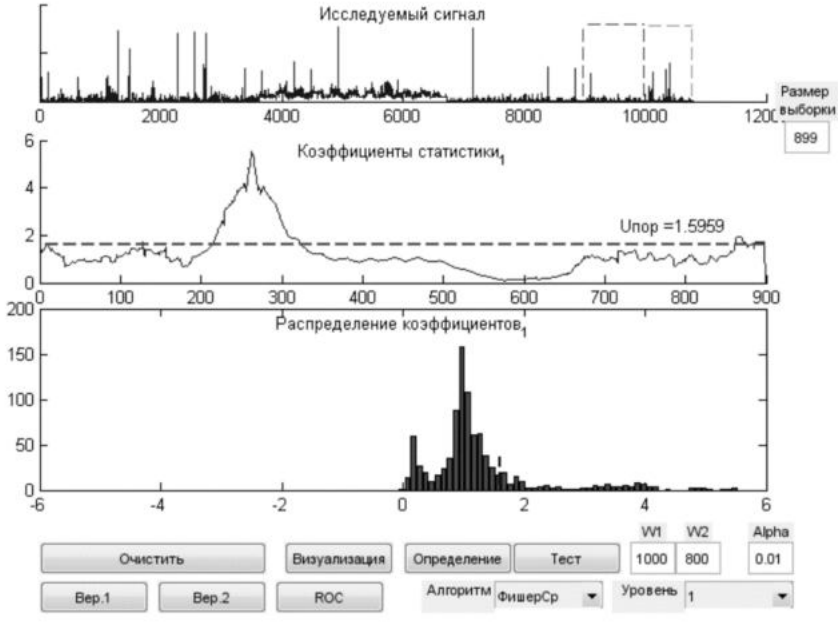

Рис. 6.7. Реализация алгоритма (6.16)

$\{a_{1x}, a_{2x}, a_{3x}, ..., a_{nx}\}_{t,j}$ и $\{a_{1y}, a_{2y}, a_{3y}, ..., a_{my}\}_{t,j}$, распределение которых имеет экспоненциальный характер.

В каждый момент времени (положении окон) t на масштабном уровне j выдвигаются две статистические гипотезы о равенстве средних значений двух выборок $\{a_{1x}, a_{2x}, a_{3x}, ..., a_{nx}\}_{t,j}$ и $\{a_{1y}, a_{2y}, a_{3y}, ..., a_{my}\}_{t,j}$: нулевая H_0: $\mu_{1,t,j} = \mu_{2,t,j}$ и альтернативная H_1: $\mu_{1,t,j} \neq \mu_{2,t,j}$. Статистика критерия записывается как

$$M = \frac{m \sum_{i=1}^{n} a_{ix}}{n \sum_{i=1}^{m} a_{iy}}. \tag{6.16}$$

Нулевая гипотеза опровергается в пользу альтернативной в случае, если $M > F_p(v_1, v_2)$, где $F_p(v_1, v_2)$ — p-квантиль распределения Фишера с $v_1 = 2n$ и $v_2 = 2m$ степенями свободы.

Пример реализации алгоритма (6.16) для первого уровня разложения показан на рис. 6.7.

Критерии (6.14), (6.15) и (6.16) аномального поведения потока событий не эквивалентны. Факт появления аномалии по одному критерию может соответствовать нормальному поведению потока согласно другому критерию. Это связано с тем, что используе-

мые критерии введены для разных статистических характеристик. В случае критерия (6.13) оценивается отклонение выборочных дисперсий, в то время как при использовании критериев (6.15) и (6.16) сравниваются математические ожидания.

Так как критерий Фишера предполагается для обнаружения быстрых высокочастотных выбросов, характеризующихся изменением дисперсии, значения выборок для него будем передавать как коэффициенты детализации. Критерий Кохрана оперирует средними и предлагается для обнаружения долговременных низкочастотных аномалий. Коэффициенты деталей имеют нулевое среднее значение, поэтому не подойдут для данного критерия. При этом предлагается в качестве выборок использовать коэффициенты аппроксимации. Критические (пороговые) значения зависят от доверительной вероятности p и размеров выборок n и m в первом случае, а во втором от доверительной вероятности p, размеров n и m и дисперсий выборок S^2. Для согласования используемых критериев необходимо рассмотреть вопрос выбора порогов обнаружения на каждом уровне декомпозиции.

6.5.4. Выбор порогов обнаружения

Применяемые критерии Фишера (для дисперсий) и Кохрана (для средних значений) предполагают, что анализируемые выборки имеют нормальное распределение [56].

Для каждого статистического критерия предлагается использовать два порога — верхний и нижний. Нижний порог вычисляется исходя из доверительной вероятности $p = 0{,}95$, а верхний порог — для доверительной вероятности $p = 0{,}999$. Причем превышение верхнего порога какой-либо из этих статистик обнаружения в момент времени t означает наличие в этой точке изменения дисперсии или среднего, что означает наличие аномалии. Превышение же нижнего порога означает высокую вероятность наличия аномалии, но не является достаточным критерием для принятия такого решения. В этом случае производится дальнейшая декомпозиция по следующему уровню разложения, и для коэффициентов этого уровня опять будут проверяться статистические критерии.

В случае превышения верхнего порога на каком-либо уровне разложения энергия аномального сигнала в основном будет сосредоточена на этом уровне, поэтому проводится реконструкция коэффициентов и дальнейшая проверка восстановленного сигнала по статистическим критериям.

В случае превышения статистическими критериями верхних пороговых значений окончательно фиксируется аномалия.

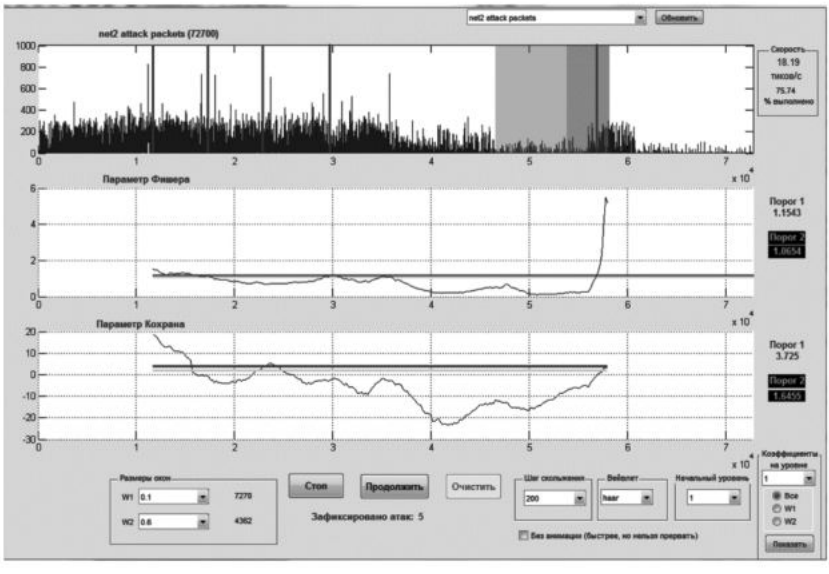

Рис. 6.8. Интерфейс программы анализа последовательности

Главное окно программы в процессе анализа последовательности изображено на рис. 6.8. На верхнем графике показывается реализация трафика сети с атаками и процесс движения окна. Два графика ниже — вычисляемые в реальном времени параметры Фишера и Кохрана соответственно и красный — верхний порог, желтый — нижний порог. На этих графиках для экономии места на экране отображается только первый уровень ВП. В случае выполнения условий, описанных в алгоритме выше, фиксируется атака и момент ее первого вхождения. Атаки отображаются в виде красных горизонтальных линий на графике трассы сверху, количество зафиксированных атак во всей последовательности отображается внизу окна.

6.6. Дискретное вейвлет-пакетное преобразование

При рассмотрении дискретношл вейвлет-пакетнго преобразования (ДВП) по алгоритму Малла на каждом шаге происходит октавополосное «расщепление» (splitting) сигнала на ВЧ и НЧ составляющие и «отсечение» ВЧ составляющей. Причина такого подхода заключена в неявном предположении, что НЧ область содержит больше информации об исходном сигнале, чем ВЧ область. В результате получается «однобокое» дерево (рис. 6.9,*а*). Такое предположение оправдано для многих реальных сигналов, однако для некоторых оно не выполняется.

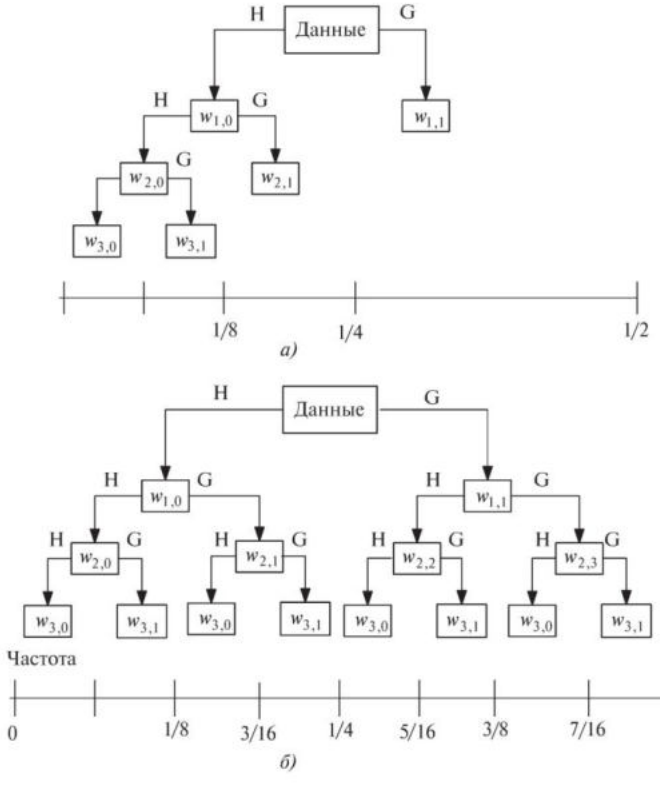

Рис. 6.9. Деревья вейвлет-преобразований: *а* — ДВП по алгоритму Малла; *б* — ДВПП (справа)

Можно усовершенствовать алгоритм Малла, предложив «дерево>, которому будет соответствовать набор подпространств сигнала с базисами, применив процесс расщепления как для НЧ, так и ВЧ составляющих сигнала. В результате получается «полное» (бинарное или сбалансированное) дерево, представленное на рис. 6.9,*б*.

Ветвям дерева будет соответствовать набор подпространств сигнала с базисами, построенными, как и для однобокого дерева согласно кратномасштабному анализу. Функции и фильтры, порождающие эти базисы, называются соответственно вейвлет-пакетами и пакетными фильтрами.

Пакетные вейвлеты предоставляют более широкую часть диапазона частот (см. рис. 6.9). В пакетном преобразовании при вычислении каждого следующего уровня коэффициенты аппроксимации и детализации и проходят через НЧ и ВЧ фильтры. В стандартном ДВП коэффициенты аппроксимации соответствуют полосе частот

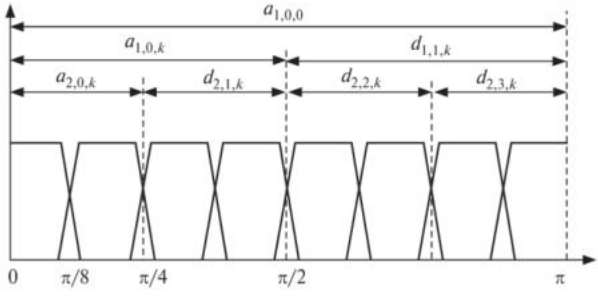

Рис. 6.10. Спектральные характеристики ВПП

$[0, 1/2j + 1]$ на уровне разложения j, а коэффициенты детализации на уровне j описывают полосу частот $[1/2j+1, 1/2j]$. Пакетная трансформация, напротив, делит всю полосу частот [0, 1/2] на полосы частот эквивалентной длины, например на данном уровне j есть $2j$ частотных «частей» соответствующей длины (см. рис. 6.8,*б*). Лучшее частотное разрешение пакетных преобразований предоставляет лучшие декорреляционные (декомпозиционные) свойства. В основу ДВПП заложены рекуррентные соотношения вида

$$a_{m+1,2p,k} = \sum_n h_n a_{m,p,2k+n}; \quad d_{m+1,2p,k} = \sum_n g_n a_{m,p,2k+n};$$

$$a_{m+1,2p+1,k} = \sum_n h_n d_{m,p,2k+n}; \quad d_{m+1,2p+1,k} = \sum_n g_n d_{m,p,2k+n};$$

где m — номер масштабного уровня; p — номер узла в пределах масштабного уровня; $k = 0, ..., N/2^m$ — номер коэффициента в пределах узла.

Сущность алгоритмов ДВПП отражена на рис. 6.10.

На первом этапе преобразования первый цифровой фильтр h_n из числового ряда $f_k = a_{0,0,k}$ выделяет децимацией* аппроксимирующих коэффициентов $a_{m,p,k}$, а фильтр g_n — децимацией детализирующих коэффициентов $d_{m,p,k}$. При переходе с масштабного уровня m на уровень $m+1$ как аппроксимирующие $a_{m,p,k}$, так и детализирующие коэффициенты $d_{m,p,k}$ разделяются вновь на низкочастотные $(a_{m+1,p,k})$ и высокочастотные $(d_{m+1,p,k})$ части спектрального диапазона. Дополнительная декомпозиция высокочастотных составляющих спектра трафика позволяет выделить локальные особенности (аномалии) и оценить флуктуации. Из множества возможных базисов вейвлет-разложения — от <минимального» (алгоритм Малла) до полного ДВПП на всех уровнях детализации экспериментально с

* Децимация — уменьшение частоты дискретизации.

учетом временных ограничений выбираются те, на которых аномальное состояние трафика проявляется наиболее четко. В качестве критерия выбора оптимального базиса ДВПП предложено использовать критерий минимума энтропии, характеризующей уровень усреднения и определяющей количество существенных коэффициентов модели трафика. Дополнительными ограничениями являются ресурсные затраты.

Критерий, по которому проводится обнаружение аномалий, представляет собой отношение дисперсий и среднего коэффициентов пакетного преобразования. Предложено два пороговых значения для каждого из отношений. Адаптация выбора уровня разложения заключается в следующем. Если на каком-либо уровне пакетного преобразования есть превышение высшего порога, выносится решение о наличии аномалии. Если же на этом уровне происходит превышение нижнего порога, значит, в этом месте возможно имеет место быть аномалия и тогда проводится дальнейшая вейвлет-декомпозиция до следующего уровня, на котором снова проводится анализ. Так происходит до того момента, пока значение отношений либо не превысит высший порог, что будет говорить об аномалии, либо перестанет превышать пороги вообще, что будет говорить об отсутствии аномалий.

Рассматривая коэффициенты деталей пакетного преобразования, ищутся высокочастотные аномалии. Рассматривая коэффициенты аппроксимации, происходит поиск низкочастотных аномалий.

Нулевая гипотеза определяется как H_0: $\sigma_1^2 = \sigma_2^2 = ... = \sigma_T^2$.

Альтернативная гипотеза H_a: $\sigma_1^2 = ... = \sigma_k^2 \neq \sigma_{k+1}^2 = ... = \sigma_T^2$.

Статистический критерий определяется как

$$D = \max(D^+, D^-),$$

где

$$D^+ = \max_{1 \leqslant k \leqslant T-1} \left(\frac{k+1}{T} - P_k \right); \quad D^- = \max_{1 \leqslant k \leqslant T-1} \left(P_k - \frac{k}{T} \right);$$

$$P_k = C_k / C_T, \quad k = 1, ...T-1.$$

Величина $C_k = \sum_{t=1}^{k} x_t^2$ является кумулятивной суммой квадратов серии некоррелированных случайных величин $\{x_t\}$ с математическим ожиданием равным нулю и дисперсией σ_t^2, $t = 1, ..., T$. Точка изменения дисперсии расположена в $k^* = \arg\max_k D$: когда абсолютное максимальное значение D достигнет некоторой заранее определенной величины, тогда k^* считается точкой изменения.

Методика обоснования порогового уровня аномального состояния сетевого трафика. В основу обоснования положен метод статистических решений для задачи проверки двух альтернативной гипотезы: H_0 и H_1 выражают предположения об отсутствии или наличии аномалии на текущем уровне сетевого трафика $f_{ad}(t)$.

Для того чтобы задача обнаружения аномалий обрела математическую содержательность, введены показатели — вероятности ложной тревоги $P_{\text{лт}}$ и пропуска аномалии $P_{\text{па}}$, понимая под ложной тревогой факт решения $\hat{H}_1$ об обнаружении аномалии при условии, что в наблюдаемом $f_{ad}(t)$ аномалия отсутствует, а под пропуском аномалии — принятие решения $\hat{H}_0$ о том, что аномалии в $f_{ad}(t)$ нет при условии, что в действительности она имеет место.

С целью обоснования применимости методов проверки статистических гипотез на основе экспериментальных данных по критерию Пирсона доказана нормальность закона распределения случайных погрешностей определения состояния сетевого трафика. Отсюда выведены зависимости для расчета вероятностей $P_{\text{па}}, P_{\text{лт}}$:

$$P_{\text{лт}} = \frac{1}{\sqrt{2\pi}} \int_{z_n}^{\infty} \frac{1}{\sqrt{D(z)}} \exp\left(-\frac{z^2}{2D(z)}\right) dz = 1 - \Phi(h);$$

$$P_{\text{па}} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z_n} \frac{1}{\sqrt{D(z)}} \exp\left(-\frac{(z-\overline{z})^2}{2D(z)}\right) dz = \Phi(h-q),$$

где

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{k^2}{2}\right) dk$$

— интеграл вероятности при $k = z(\sqrt{D(z)})^{-1}$;

$$z = \int_0^T f_a(t_i)\mathcal{E}_a(t_i)\, dt$$

— корреляционный интеграл, определяющий степень сходства наблюдаемой реализации $f_a(t_i)$ с ожидаемой аномалией $\mathcal{E}_a(t_i)$; z_n — пороговый уровень аномальности сетевого трафика; $h = z_n(\sqrt{D(z)})^{-1}$ — нормированный пороговый уровень; $q = z(\sqrt{2\overline{z}/N_0})^{-1}$ — параметр обнаружения, равный соотношению сигнал/шум.

Пороговый уровень аномальности сетевого трафика z_n рассчитывается в соответствии с принятым критерием оптимальности. В СОВ может быть использован критерий Неймана–Пирсона в форме задачи условной оптимизации целевой функции $P_{\text{па}} + \mu(0{,}05 - P_{\text{лт}})$ в следующей формулировке: минимизировать $P_{\text{па}}$ при ограничении на величину $P_{\text{лт}}$, т. е. найти нормированный порог h, и путем подстановки h в (6.20) определить минимальную величину $P_{\text{па}}$.

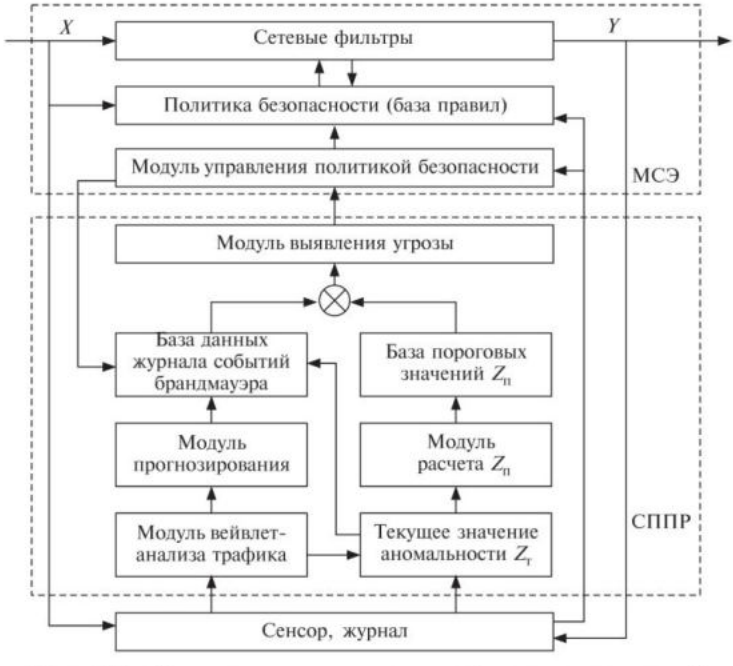

Рис. 6.11. Контур управления системы обнаружения аномалий

В основу решения задачи условной оптимизации положен метод нелинейного программирования с использованием теоремы Куна–Таккера. Для этого составлена функция Лагранжа с ограничением $0,05 - P_{лт} \geqslant 0$ вида

$$L(h, \mu) = P_{па} + \mu(0{,}05 - P_{лт}) = \Phi(h - q) + \mu(\Phi(h) - 0{,}95), \quad (6.21)$$

где μ — неопределенный множитель Лагранжа.

Расчеты показали, что пороговый уровень аномальности сетевого трафика пропорционален квадратному корню из дисперсии, т. е. $z_n = 1{,}644685\sqrt{D(z)}$, при этом обеспечивается минимум среднего риска принятия неверного решения.

Прототип системы выявления и блокирования аномалий представлен на рис. 6.11, он является развитием архитектуры межсетевого экранирования с поддержкой функции автоматического управления базой правил брандмауэра при обнаружении аномалий трафика ККС.

Исходя из анализа известных работ, можно сделать вывод, что методы обнаружения аномалий, основанные на вейвлет-преобразовании, представляются одними из наиболее эффективных.

6.7. Обнаружение DoS- и DDoS-атак методами мультифрактального анализа

6.7.1. Фрактальные свойства телекоммуникационного трафика

Исследования [21] показывают, что сетевой трафик является самоподобным на некоторых временных масштабах.

Для исследования структуры трафика и иллюстрации его фрактального (самоподобного) характера в [53] использовался процесс агрегирования профилей сетевого трафика. В качестве примера были получены временные ряды с временем агрегации: 0,1 с; 1 с; 10 с (рис. 6.12).

Агрегирование профиля сетевого трафика с временем агрегации 60 с привела к существенным искажениям полученного временного ряда. Процесс агрегирования производился в соответствии с методикой по формуле

$$Y_k^{(m)} = \frac{1}{m} \sum_{i=(k-1)m+1}^{km} Y_i,$$

где Y_i — номер отсчета в полученном профиле; m — размер блока или интервал времени агрегирования; k — номер блока.

Для моделирования атаки в канале связи была выбрана DoS-атака ICMP-flooding, осуществляющая затопление узла-жертвы служебными ICMP-пакетами. IP-адрес машины, с которой была осуществлена атака, — 192.168.1.191 (в отправляемых пакетах был использован поддельный IP-адрес отправителя — 192.168.1.10). Узел-жертва, на который была направлена лавина ICMP пакетов, имел IP-адрес 192.168.1.207. Время начала захвата пакетов — 16 января 2012 г. в 19:43:05.874968000 (по московскому времени). Время прихода последнего захваченного пакета — 16 января 2012 г. в 19:47:12.551204000 (по московскому времени). Атака ICMP-flooding продолжалась одну минуту, время ее начала 19:45:00.

На рис. 6.13 показаны временные последовательности пакетов, наблюдаемые в канале связи (в случае DoS атаки ICMP-flooding) при различном времени агрегирования.

При времени агрегирования 0,1 с произведено 2468 измерений. Минимальное значение числа пакетов, наблюдаемых в канале связи, равно 0, а максимальное — 148. Количество нулевых значений 174, а ненулевых — 2294.

Число интервалов для построения гистограммы равно максимальному количеству наблюдаемого скопления пакетов и состави-

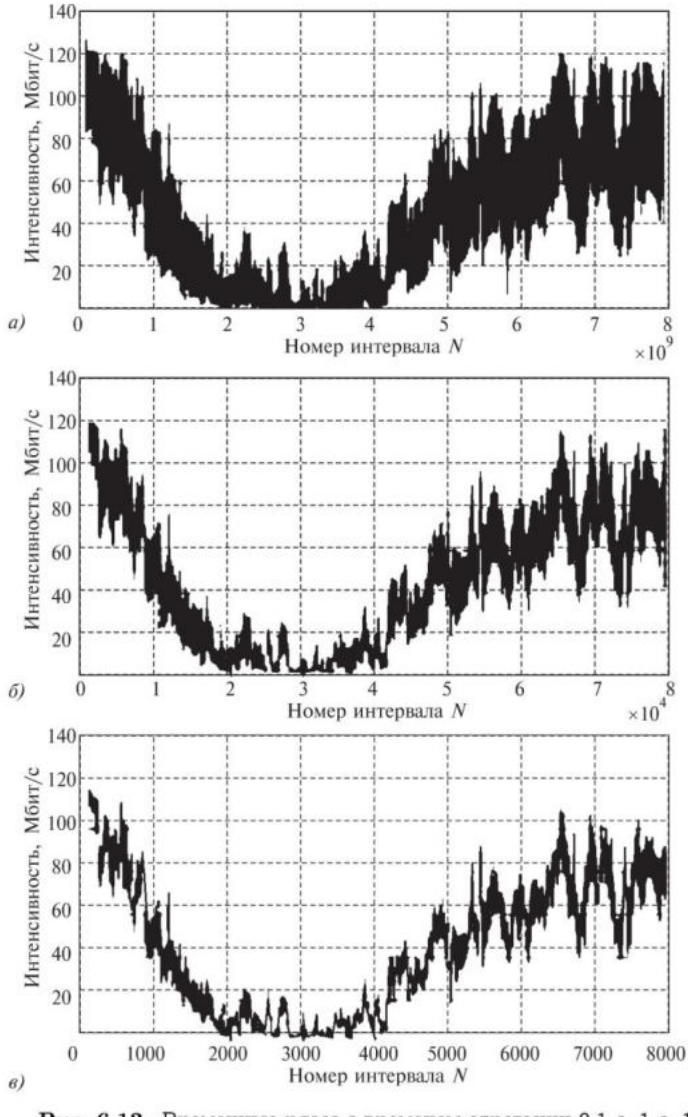

Рис. 6.12. Временные ряды с временем агрегации 0,1 с, 1 с, 10 с

ло, соответственно, 148. Отсюда ширина каждого интервала равна 1 пакету.

При времени агрегирования 0,5 с произведено 495 измерений. Максимальное значение числа наблюдаемых в канале пакетов 0, максимальное — 204. Количество нулевых значений — 1, ненулевых — 494.

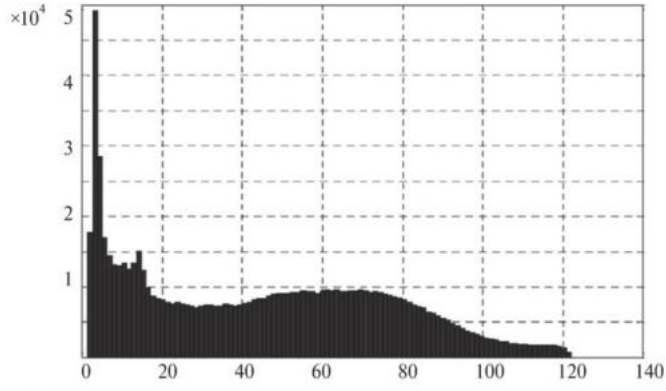

Рис. 6.13. Гистограмма изучаемой реализации, время агрегации 0,1 с

При времени агрегирования 1 с произведено 248 измерений. Минимальное значение числа наблюдаемых пакетов в канале связи 0, а максимальное 248. Количество нулевых значений — 1, ненулевых — 247.

Визуально выявить присутствие самоподобного процесса, а также долговременной зависимости (ДВЗ) можно, исследуя частотную область. Если рассматривать самоподобные процессы в частотной области, то явление ДВЗ приводит к степенному характеру спектральной плотности вблизи нуля:

$$S(w) = \omega^{-\gamma} L(\omega) \quad \text{при } \omega \to 0$$

где $0 < \gamma < 1$; L — медленно меняющаяся функция в нуле; $S(\omega)$ — функция спектральной плотности.

Таким образом, спектральная плотность стремится к $+\infty$, когда ω приближается к нулю, подобное явление называется $1/f$-шумом.

Из рис. 6.14 видно, что функция спектральной плотности достигает максимального значение в нуле, а ее форма напоминает гиперболическую функцию. Таким образом, визуальный анализ частотной области временного ряда показывает присутствие ДВЗ процессов.

Для более точной оценки самоподобных свойств трафика вводят понятие коэффициента Херста H [21]. Если данный коэффициент находится в пределах $0{,}5 < H < 1$, то исследуемый процесс проявляет самоподобные свойства. Приближение H к 1 говорит о высокой самоподобности данного процесса и о том, что поведение процесса является персистентным или процесс обладает длительной памятью. То есть, если на некотором временном промежутке в прошлом

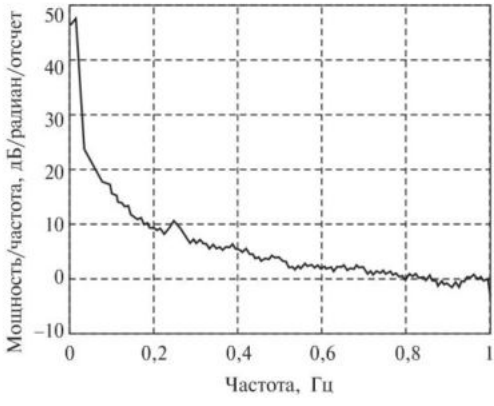

Рис. 6.14. График функции спектральной плотности

наблюдалось положительное приращение процесса, другими словами, его увеличение, то и в будущем в среднем будет происходить увеличение. При $H = 0{,}5$ отклонение процесса от среднего является случайными и не зависит от предыдущих значений.

При $0 < H < 0{,}5$ процесс является переменчивым, т. е. увеличение относительно среднего в прошлом в будущем сменится в противоположном направлении.

На практике проверка присутствия ДВЗ и оценка параметра Херста является непростой задачей. Главная проблема в том, что весьма трудно найти различие между стационарным и нестационарном ДВЗ процессом вследствие присутствия локальных трендов, циклов и т. д., что свойственно нестационарным процессам. Если имеется большая выборка, то получение более точной оценки параметра Херста становится более легкой задачей, но можно быть уверенным, что в больших выборках нестационарные эффекты присутствуют благодаря дневным циклам трафика. Гипотеза о стационарности с высокой достоверностью может быть принята только в периоды высокой загрузки канала. Важной частью статистического анализа временных рядов является идентификация и удаление тренда.

Очевидным подходом для решения проблемы стационарности является выбор интервалов времени, где предположение о стационарности трафика справедливо (локальная стационарность).

Для пульсирующих данных, таких, как измеренный трафик, необходимо использовать инструментальное средство, основанное на методе определения точки измерения, которое заключается в том, чтобы перемещать окно по данным и сравнивать распределение выборок в двух половинах окна. Если два распределения существенно

различны, то предположение о стационарности для окна отвергается. Сравнение распределений двух рядов равного размера выполняются по критерию Колмогорова–Смирнова.

На основании проведенных тестов на стационарность для анализа можно выбрать несколько подмножеств из всех измеренных данных, каждый из которых получает оценку Херста H. Более подробно данный процесс можно описать следующим образом.

Оценим показатель Херста для блоков данных D. Рассмотрим K сегментов ряда, каждый длиной N. Показатель Херста H оценивается в каждом сегменте, S_i, $i = 1, 2, ..., K = DN$ с использованием, в нашем случае, RS-анализа, анализа изменения дисперсий, периодограмного анализа и анализа абсолютных значений. Если оценки в i-м блоке обозначены как $\hat{H}_i$, то для соответствующего N оценку показателя Херста можно найти в виде

$$\hat{H}_N = \frac{N}{D} \sum_{i=1}^{D/N} \hat{H}_i.$$

Таким образом, если выбрать N достаточно большим, то можно обеспечить приемлемую сходимость оценки так, чтобы для стационарного процесса оценка $\hat{H}_N$ не зависела от N. Самоподобные свойства трафика позволяют с достаточной степенью достоверности прогнозировать появление на сегменте сети временных периодов с перегрузкой.

В то же время трафик также показывает мультифрактальные свойства в меньших масштабах времени (порядка миллисекунд). Таким образом, можно говорить о том, что свойство самоподобия отражает долгосрочное поведение измеряемого сигнала, а мультифрактальные свойства отображают мгновенное поведение сигнала.

Поиск распределения сингулярностей (особенностей) в мультифрактальном сигнале имеет особенно важное значение для анализа его свойств. В настоящее время известно несколько методов для определения спектра сингулярностей мультифрактального сигнала на основе вейвлет-преобразования [21]. В [25] к задачам обнаружения аномалий сетевого трафика применен мультифрактальный анализ.

6.7.2. Обнаружение DoS- и DDoS-атак методом мультифрактального анализа

В [22, 23] для обнаружения аномальных выбросов трафика предлагается использовать *метод максимумов модулей вейвлет-пре-*

образования (ММВП), позволяющий выявить сингулярности сигнала.

В качестве анализируемых последовательностей были взяты наборы данных, предоставленные Линкольнской лабораторией Масачусетского технологического института (1999 DARPA Intrusion Detection Evaluation) [30], представляющие собой сетевой трафик, собранный на граничном маршрутизаторе университетской сети. Каждая последовательность длиной около 24 часов с шагом дискретизации в 1 с. Представлен как «чистый» трафик сети без атак, так и с различными видами аномалий: относящиеся к атакам типа «отказ в обслуживании» (DoS) и к различным типам сканирования. DoS-атаки также включают в себя распределенные DoS-атаки (DDoS), которые использует много хостов для запуска атак против одной жертвы.

Алгоритм оценки параметров мультифрактального спектра по методу ММВП следующий.

Шаг 1. Производим декомпозицию исходного сигнала $f(t)$ при помощи непрерывного диадного вейвлет-преобразования материнским вейвлетом $\psi(t)$ на коэффициенты:

$$W_f(u,j) = (f(t), \psi_{u,s}(t)) = 2^{-j/2} \int_{-\infty}^{\infty} \frac{t-u}{2^j}\, dt.$$

Шаг 2. В получившемся массиве вейвлет-коэффициентов находим положения локальных максимумов $\{u_p(j)\}_{p \in Z}$ и находим их абсолютное значения, формируя массив локальных максимумов $|Wf(u_p, j)|$;

Шаг 3. Вычисляем функцию разбиения

$$S(q,j) = \sum_p |W_f(u_p, j)|^q.$$

Шаг 4. Для каждого $q \in R$ вычисляется масштабный показатель

$$\tau(q,j) = \lim \inf_{j \to 0} \frac{\ln S(q,j)}{\ln 2^j}.$$

Шаг 5. Вычисляется мультифрактальный спектр $f_L(\alpha)$ при помощи преобразования Лежандра:

$$f_L(\alpha) = \min_{q \in R}[q(\alpha + 1/2) - \tau(q)].$$

Шаг 6. Для каждой октавы j вычисляются мультифрактальные

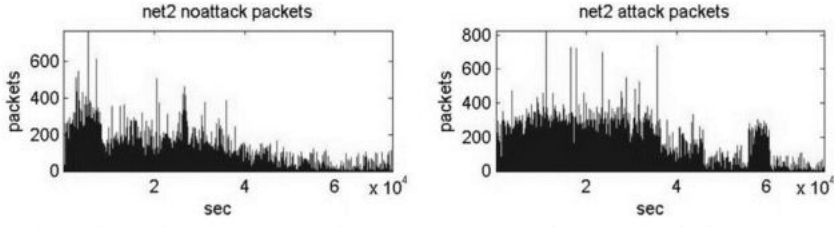

Рис. 6.15. Реализации трафика сети: *а* — сеть без аномалий; *б* — сеть с аномальной активностью

ные размерности порядка q:

$$D_{q,j} = \frac{1}{q-1}[q(\alpha(q,j) - f(\alpha(q),j)].$$

При $q < 0$ значение $S(q,j)$ зависит, в основном, от малых максимумов амплитуды $|Wf(u_p,j)|$. Поэтому вычисления могут быть неустойчивы. Чтобы избежать появления ложных максимумов модуля, созданных вычислительными погрешностями в областях, где f почти константа, вейвлет-максимумы объединяются в цепочку, чтобы образовать кривую максимумов в зависимости от масштаба. Если $\psi = (-1)^P \Theta^{(p)}$, где $\Theta = (1/\sqrt{2\pi})\mathrm{e}^{-t^2/2}$ — функция Гаусса, то все линии максимумов $u_p(j)$ определяют кривые, которые распространяются до предела $j = 0$. Поэтому все линии максимумов, которые не распространяются до наименьшего масштаба, удаляются при вычислении $S(q,j)$.

Результаты применения метода ММВП при обнаружении аномалий в трафике [25, 29] данвы на рис. 6.15. На рис. 6.15,*а* представлена реализация «чистого» трафика сети без атак длиной 72700 с (около 22 ч) с шагом дискретизации 1 с, а на рис. 6.15,*б* — та же реализация с аномальными выбросами. Визуально заметно, что реализация с атаками отличается наличием аномальных выбросов в трафике.

Выполним $n = 16$ (максимальное целое $\log_2 72700$) непрерывных вейвлет-преобразований исходных реализаций. Спектрограммы для некоторых октав показаны на рис. 6.16. На спектрограммах четко прослеживаются частотно-временные локализации всех особенностей сигнала. Так, аномальный выброс, происходящий в области $6 \cdot 10^4$ с (рис. 6.16,*б*, справа) проявляет себя в виде резких возмущений в поле спектрограммы, чего не наблюдается в этой области на рисунках слева. Анализ спектрограмм позволяет сделать вывод, что некоторые особенности сигнала могут проявлять себя на одном уровне разложения, но никак не проявлять на другом, поэто-

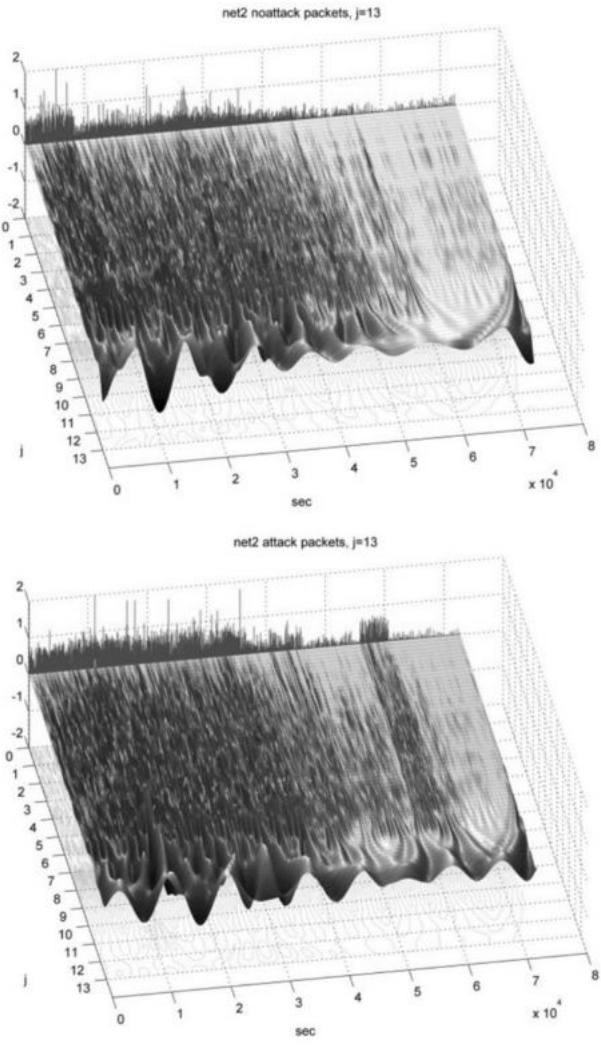

Рис. 6.16. Спектрограммы вейвлет-преобразований: вверху — для сети без атак, внизу — для сети с аномалиями, для октавы $j = 13$

му для выявления всех особенностей сигнала он анализируется по всем октавам.

На рис. 6.17 наглядно иллюстрируется различие мультифрактальных спектров реализаций с аномалиями и без аномалий на каждому масштабном уровне разложения j. От октавы к октаве спектры нормальной и «атакуемой» сети практически не отличаются хаусдорфовой размерностью в связи с тем, что анализируемые реализа-

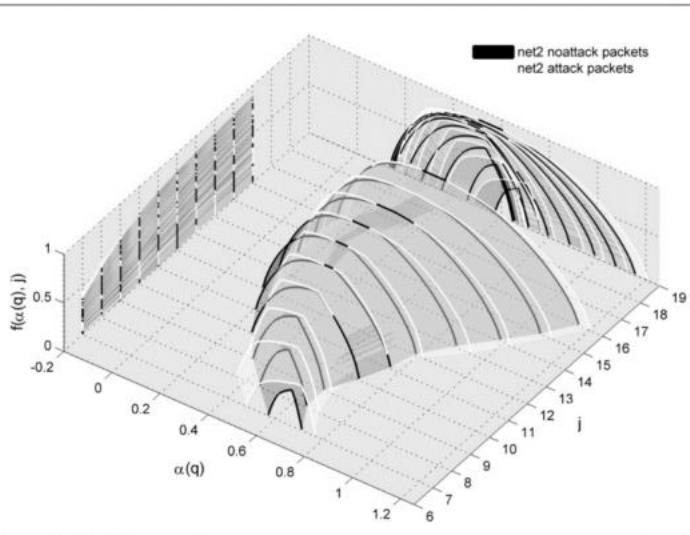

Рис. 6.17. Мультифрактальные спектры в сравнении, черный — без аномалий, белый — с аномалиями

ции имеют одинаковые длины. Однако интервалы задания и другие размерности существенно различны. Особенно большие различия проявляются в правом крыле спектра при $q < 0$.

Таким образом, различия в особенностях трафика с аномалиями и без аномалий четко отражаются на графиках их спектров сингулярностей, которые могут быть найдены с использованием метода ММВП. Видно, что поскольку последовательности имеют одинаковую длину, хаусдорфова размерность мультифракталов $f(\alpha_0) = D_0$ практически не меняется (максимумы функций одинаковы). Зато различны информационная D_1 и корреляционная размерность D_2. Так же различны границы, в которых заданы функции $\alpha_{\min}, \alpha_{\max}$.

Формализуя отличие спектров друг от друга, можно сравнить фрактальные размерности D_1, корреляционные размерности D_2 и интервалы, характеризующие «ширину» спектра Лежандра для каждой из реализаций по каждой октаве разложения.

Анализ полученных зависимостей показывает, что различия двух реализаций проявляются в их мультифрактальных спектрах, построенных при помощи разработанного программного обеспечения на основе метода ММВП, независимо от количества вовлеченных в анализ масштабных уровнях разложения (октав) j. Характеристики спектра на каждом уровне разложения могут выявить локальные особенности сигнала, позволяющие обнаружить их путем анализа мультифрактальных спектров реализаций на данном уровне разложения.

Хаусдорфовы размерности сравниваемых реализаций D_0 и D_1, определяющие количество найденных локальных максимумов при данном количестве уровней разложения, наиболее отличаются при малом числе уровней (октав) разложения. Информационные размерности сравниваемых реализаций D_1, отвечающие за разницу в левых склонах мультифрактального спектра, различаются на небольшую, но постоянную величину и практически не зависят от количества уровней разложения. Можно говорить о том, что наличие в сигнале многих и продолжительных атак и аномальной активности изменяет самоподобную природу трафика, что отражается в различии в размерностей D_1.

7 МЕТОДЫ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ В СИСТЕМАХ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ

7.1. Методы Data Mining

В настоящее время ищутся новые методы анализа данных в системах обнаружения вторжений (СОВ), и все больше внимания уделяется применению методов интеллектуального анализа данных (ИАД, Data Mining). ИАД — это процесс выявления значимых корреляций, образцов и тенденций в больших объемах данных [19, 43].

Data Mining — это процесс обнаружения в сырых данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности.

Суть и цель технологии Data Mining можно охарактеризовать так: это технология, которая предназначена для поиска в больших объемах данных неочевидных, объективных и полезных на практике закономерностей. Неочевидных — это значит, что найденные закономерности не обнаруживаются стандартными методами обработки информации или экспертным путем. Объективных — это значит, что обнаруженные закономерности будут полностью соответствовать действительности, в отличие от экспертного мнения, которое всегда является субъективным. Практически полезных — это значит, что выводы имеют конкретное значение, которому можно найти практическое применение.

Традиционные методы анализа данных (статистические методы) в основном ориентированы на проверку заранее сформулированных гипотез (verification-driven data mining) и на «грубый» разведочный анализ, составляющий основу оперативной аналитической обработки данных (OnLine Analytical Processing, OLAP), в то время как одно из основных положений Data Mining — поиск неочевидных закономерностей. Инструменты Data Mining могут находить такие закономерности самостоятельно и также самостоятельно строить гипотезы о взаимосвязях. Поскольку именно формулировка гипотезы

относительно зависимостей является самой сложной задачей, преимущество Data Mining по сравнению с другими методами анализа является очевидным.

К методам и алгоритмам Data Mining относятся следующие:

- искусственные нейронные сети;
- деревья решений;
- символьные правила;
- методы ближайшего соседа и k-ближайшего соседа;
- метод опорных векторов;
- байесовские сети;
- линейная регрессия;
- корреляционно-регрессионный анализ;
- иерархические методы кластерного анализа;
- неиерархические методы кластерного анализа, в том числе алгоритмы k-средних и k-медианы;
- методы поиска ассоциативных правил, в том числе алгоритм Apriori;
- метод ограниченного перебора;
- эволюционное программирование и генетические алгоритмы, разнообразные методы визуализации данных и множество других методов.

Data Mining может состоять из трех стадий.

1) выявление закономерностей (осуществляется исследование набора данных с целью поиска скрытых закономерностей; также должна осуществляться валидация закономерностей, т. е. проверка их достоверности на части данных, которые не принимали участие в формировании закономерностей);

2) использование выявленных закономерностей для предсказания неизвестных значений (обнаруженные закономерности используются непосредственно для прогнозирования — решаются задачи классификации и прогнозирования);

3) анализ исключений; стадия предназначена для выявления и объяснения аномалий, найденных в закономерностях (анализируются исключения или аномалии, выявленные в найденных закономерностях).

Статистические методы Data Mining классифицированы на четыре группы:

1) дескриптивный анализ и описание исходных данных;

2) анализ связей (корреляционный и регрессионный анализ, факторный анализ, дисперсионный анализ);

3) многомерный статистический анализ (компонентный анализ, дискриминантный анализ, многомерный регрессионный анализ, канонические корреляции и др.);

4) анализ временных рядов (динамические модели и прогнозирование).

Кибернетические методы Data Mining:

- искусственные нейронные сети (распознавание, кластеризация, прогноз);
- эволюционное программирование (в том числе алгоритмы метода группового учета аргументов);
- генетические алгоритмы (оптимизация);
- ассоциативная память (поиск аналогов, прототипов);
- нечеткая логика;
- деревья решений;
- системы обработки экспертных знаний.

Основная идея этих методов применительно к СОВ основана на предположении о том, что активность пользователей и программ в системе может быть отслежена и построена ее математическая модель. Для прикладного применения в СОВ методы ИАД можно рассмотреть с двух позиций: методы обнаружения нарушений (misuse detection), которые строят модель атаки, а в процессе обнаружения используют методы классификации, и методы обнаружения аномалий (anomaly detection), которые строят модель нормальной активности, а в процессе обнаружения используют методы поиска исключений.

Формально, будем использовать следующую модель задачи классификации:

Ω — множество анализируемых объектов (в терминах предмета распознавания образов — пространство образов);

ω: $\overline{\omega} \in \Omega$ — объект классификации (распознавания) – образ;

$g(\omega)$: $\Omega \to M$, $M = \{1, 2, \ldots, m\}$ — индикаторная функция, разбивающая пространство образов Ω на m непересекающихся классов $\Omega^1, \Omega^2, ..., \Omega^m$. Индикаторная функция неизвестна наблюдателю;

X — пространство наблюдений, воспринимаемых наблюдателем — пространство признаков (признак — некоторое количественное измерение объекта);

$x(\omega)$: $\Omega \to X$ — функция, ставящая в соответствие каждому объекту ω точку $x(\omega)$ в пространстве признаков. Вектор $x(\omega)$ — это образ объекта, воспринимаемый наблюдателем;

в пространстве признаков определены непересекающиеся множества точек, соответствующих образам одного класса;

$\hat{g}(x)$: $X \to M$ — решающее правило — оценка для $g(\omega)$ на основании $x(\omega)$, т. е. $\hat{g}(x) = \hat{g}(x(\omega))$. (Решающим правилом называют правило отнесения образа к одному из классов на основании его вектора признаков.)

Пусть $x_j = x(\omega_j)$, $j = 1, 2, \ldots, N$, — доступная наблюдателю информация о функциях $g(\omega)$ и $x(\omega)$ (сами эти функции наблюдателю неизвестны). Тогда (g_j, x_j), $j = 1, 2, \ldots, N$, есть множество прецедентов (образов, правильная классификация которых известна).

Задача заключается в построении такого решающего правила $\hat{g}(x)$, чтобы распознавание проводилось с минимальным числом ошибок.

Обычный случай — считать пространство признаков евклидовым, т. е. $X = R^1$. Качество решающего правила измеряют частотой появления правильных решений. Обычно его оценивают, наделяя множество объектов Ω некоторой вероятностной мерой. Тогда задача записывается в виде

$$\min P\{g(x(\omega)) \neq g(\omega)\}. \tag{7.1}$$

Методы ИАД, применяемые в IDS, делятся на две группы: методы *обнаружения нарушений* (misuse detection) [3], которые строят модель атаки, а в процессе обнаружения используют ИАД методы классификации, и методы *обнаружения аномалий* (anomaly detection) [4–6], которые строят модель нормальной активности, а в процессе обнаружения используют ИАД методы поиска исключений.

Сформулируем проблемы, которые должен решать метод ИАД для задачи выявления вторжений:

- необходимость работы с *разнородными сложно структурированными данными большого объема,* поскольку источниками информации являются различные журналы, протоколы и т. д.;
- *наличие «шума»* в тренировочных данных, т. е. редких нетипичных нормальных событий, редких нетипичных атак, а также ошибок эксперта, допущенных при подготовке тренировочного набора;
- необходимость распознавания атак в режиме в *реальном времени.*

7.2. Метод опорных векторов

Метод опорных векторов (Support Vector Machine, SVM) был описан в работах В.Н. Вапника [31, 32]. SVM — это математический

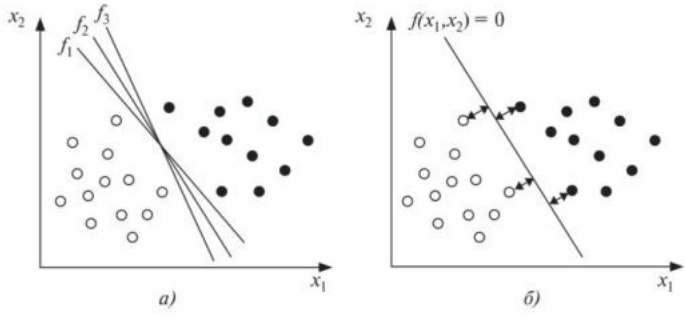

Рис. 7.1. Иллюстрация основной идеи SVM

метод получения функции, решающей задачу классификации. Идея метода возникла из геометрической интерпретации задачи классификации. Пусть два множества точек можно разделить плоскостью (в двумерном пространстве — прямой). Тогда таких плоскостей будет бесконечное множество (рис. 7.1,*а*). Выберем в качестве оптимальной такую плоскость, расстояния до которой ближайших точек обоих классов равны (рис. 7.1,*б*). Ближайшие точки-векторы называются *опорными*. Поиск оптимальной плоскости приводит к задаче квадратичного программирования при множестве линейных ограничений-неравенств. В 90-х гг. прошлого века метод SVM был усовершенствован: разработаны эффективные алгоритмы поиска оптимальной плоскости, найдены способы обобщения на нелинейные случаи и ситуации с числом классов, большим двух [31, 32].

Рассмотрим задачу классификации для 2-х классов X_1 и X_2 (векторы пространства $\mathbf{R}^n$). То есть на практике располагаем исходными данными, характеризующими нормальную активность пользователей, и некоторыми примерами атак. Будем считать, что эти классы не пересекаются. Тогда существует единичный вектор φ и число c, такие, что $(x_1, \varphi) > c$ при $x_1 \in X_1$ и $(x_2, \varphi) < c$ при $x_2 \in X_2$. В таком случае говорят, что X_1 и X_2 разделимы гиперплоскостью.

Обозначим $c_1(\varphi) = \min\limits_{x_1 \in X_1} (x_1, \varphi)$ и $c_2(\varphi) = \max\limits_{x_2 \in X_2} (x_2, \varphi)$. Тогда $(x_1, \varphi) > c_1(\varphi)$ при $x_1 \in X_1$, а $(x_2, \varphi) > c_2(\varphi)$ при $x_2 \in X_2$. Если $c_1(\varphi) \geqslant c_2(\varphi)$, то гиперплоскость

$$\Pi(x_1, \varphi) = \frac{c_1(\varphi) + c_2(\varphi)}{2} \tag{7.2}$$

разделяет X_1 и X_2. Существует множество разделяющих гиперплоскостей в силу непрерывности $c_1(\varphi)$ и $c_2(\varphi)$. Задача состоит в нахождении оптимальной разделяющей гиперплоскости, формально соответствующей вектору $\varphi_{\text{опт}}$, при котором достигается макси-

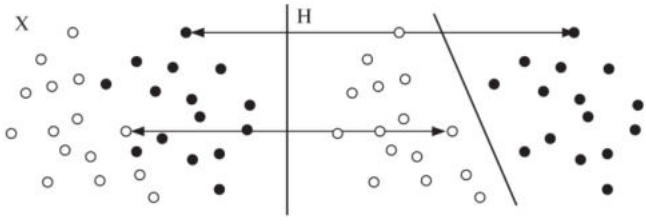

Рис. 7.2. Метод SVM

мум $\Pi(\varphi)$ (логично, что разделяющая гиперплоскость должна быть расположена максимально далеко от ближайших к ней точек обоих классов). Доказана теорема, что если два множества X_1 и X_2 разделимы гиперплоскостью, то оптимальная разделяющая гиперплоскость существует и единственна.

Однако на практике выборка редко является линейно разделимой. Поэтому можно применять следующий удобный в СОВ подход — осуществить переход от исходного пространства признаковых описаний объектов X к новому пространству H с помощью некоторого преобразования ψ: $X \to H$. Если выборка в X не противоречива и H имеет достаточно высокую размерность, то всегда найдется пространство, в котором она разделима.

Пространство H называют спрямляющим.

Функция K: $X \times X \to \mathbf{R}$ называется ядром, если она представима в виде $K(x_1, x_2) = \langle \psi(x_1), \psi(x_2) \rangle_H$ при некотором отображении ψ: $X \to H$, где H — пространство со скалярным произведением (пространство евклидово, в общем случае гильбертово).

В качестве варианта метода решения задачи, необязательно заниматься подбором отображения ψ и строить H, а достаточно только подобрать ядро (так называемую kernel function).

Основная идея метода SVM проиллюстрирована на рис. 7.2. Исходные объекты (в левой части рисунка) преобразуются при помощи ядерных функций. После этого новый набор преобразованных объектов (в правой части рисунка) уже линейно разделим. Таким образом, вместо построения сложной кривой требуется лишь провести оптимальную прямую, которая отделит объекты типа GREEN от объектов типа RED (т. е. образы «нормальной» активности от атак).

В качестве ядерных функций предлагается использовать потенциальные функции (ПФ). В этом случае каждая точка (образ) образует в пространстве признаков X некоторое поле притяжения. Например, можно рассматривать каждую точку как точечный электрический заряд. Электрическое поле описывается потенциалом,

создаваемым системой зарядов во всем пространстве. Изменение потенциала электрического поля по мере удаления от заряда обратно пропорционально квадрату расстояния. Потенциал, таким образом, может служить мерой удаления точки от заряда. Когда поле образовано несколькими зарядами, потенциал в каждой точке этого поля равен сумме потенциалов, создаваемых в этой точке каждым из зарядов. Если заряды, образующие поле, расположены компактной группой, потенциал поля будет иметь наибольшее значение внутри группы зарядов и убывать по мере удаления от нее. Тогда $K(x, y)$, $x, y \in X$, — потенциальная функция, такая, что $K(x, y) > 0$ при $x \neq y$,

$$K(x, y) = K(x, x + \mu(y - x)) = \tilde{K}(\mu), \tag{7.3}$$

где $\tilde{K}(\mu)$ — монотонно убывающая функция и $\tilde{K}(0)$ — ее максимальное значение.

При использовании указанных выше методов и в результате при построении оптимальной канонической гиперплоскости $\langle w, p\rangle$ в пространстве характеристик H, приходим к решению следующей оптимизационной задачи:

$$\min_{w \in H, \xi_i \in \mathbf{R}} \frac{1}{2}\|w\|^2 + \frac{1}{vl}\sum_{i}^{l} \xi_i - 1 \tag{7.4}$$

при $(w\varphi(x_i) - p) \geqslant 1 - \xi_i$, $\xi_i \geqslant 0 \forall i \in [1, N]$. Здесь v — параметр регуляризации; l — размерность пространства $\mathbf{R}$.

После решения оптимизационной задачи решающая функция для каждой точки x имеет вид

$$f(x) = \operatorname{sgn}(w\varphi(x) - 1). \tag{7.5}$$

Если использовать метод Лагранжа и ввести дополнительные переменные (множители Лагранжа) a_i, тогда можно представить оптимизационную задачу как

$$\min\left\{\frac{1}{2}\sum_{i,j} a_i a_j K_\varphi(x_i, x_j)\right\}$$

при $0 \leqslant a_i \leqslant 1/vl$, $\sum_i a_i = 1$.

Решающая функция принимает вид

$$f(x) = \operatorname{sgn}\left(\sum_i a_i K_\varphi(x_i, x) - 1 + p\right). \tag{7.6}$$

Параметр регуляризации v, $0 < v < 1$, задает компромисс меж-

ду точностью модели, определяемой величиной тренировочной ошибки $\sum \xi_i$, и способностью модели к обобщению, определяемой величиной границы $(1/2)\|w\|^2$. Параметр устанавливается априори.

Достоинствами SVM с использованием ПФ являются:

- получение функции классификации с минимальным уровнем ошибки классификации;
- возможность использования линейного классификатора для работы с нелинейно разделяемыми данными, сочетая простоту с эффективностью;
- возможность работы с разнородными сложно структурированными данными за счет использования различных ПФ;
- в случае изменения структуры анализируемых данных, достаточно заменить только используемую ПФ без замены самого алгоритма;
- по сути в SVM решается главным образом задача квадратичного программирования, имеющая единственное решение, и для нее существует множество изученных эффективных методов оптимизации, что позволяет работать в режиме реального времени.

Однако SVM имеет и некоторые незначительные недостатки, а именно:

- решающая функция $f(x)$ зависит от параметра v, устанавливаемого априори;
- SVM чувствителен к наличию «шума» в тренировочном наборе.

Для преодоления этих недостатков в качестве одного из вариантов предлагается использовать математический аппарат нечетких множеств.

Подход к формализации понятия нечеткого множества состоит в обобщении понятия принадлежности. В обычной теории множеств существует несколько способов задания множества. Одним из них является задание с помощью характеристической функции, определяемой следующим образом. Пусть U — так называемое универсальное множество, из элементов которого образованы все остальные множества, рассматриваемые в данном классе задач, например множество всех целых чисел, множество всех гладких функций и т. д. Характеристическая функция множества $A \subseteq U$ — это функция μ_A, значения которой указывают, является ли $x \in U$ элементом множества A:

$$\mu_A(x) = \begin{cases} 1, & x \in A; \\ 0, & x \notin A. \end{cases} \tag{7.7}$$

Особенностью этой функции является бинарный характер ее значений.

С точки зрения характеристической функции нечеткие множества есть естественное обобщение обычных множеств, когда мы отказываемся от бинарного характера этой функции и предполагаем, что она может принимать любые значения на отрезке [0,1]. В теории нечетких множеств характеристическая функция называется функцией принадлежности, а ее значение $\mu_A(x)$ — степенью принадлежности элемента x нечеткому множеству A.

Более строго нечетким множеством A называется совокупность пар

$$A = \{\langle x, \mu_A(x)\rangle \mid x \in U\},$$

где μ_A — функция принадлежности, т. е. μ_A: $U \to [0,1]$.

Итак, в оптимизационную задачу необходимо включить нечеткую функцию принадлежности элементов тренировочного набора $\mu_A(x)$. В результате «шумы» будут иметь меньшую степень принадлежности, чем корректные значения, и суть задачи в необходимости построения гиперплоскости H, разделяющей два нечетких множества.

Поскольку традиционные сигнатурные методы не обеспечивают должного уровня защиты, использование Data Mining методов в СОВ является активно развивающимся направлением.

Для определения атак нужно сформировать вектор признаков, подобный вектору, который формируется для искусственной нейронной сети. Затем с помощью специального программного обеспечения, например SVM Light [33], произвести обучение SVM-классификатора. В результате получится функция, которая будет производить классификацию векторов-признаков, т. е. распознавать, к какому классу относится текущее действие ПО или пользователя — правомерному или запрещенному. Методы использования и обучения SVM в сфере сетевой безопасности еще до конца не изучены. Ясно только, что данный подход обладает существенной мощью и имеет большие перспективы развития, в том числе в задаче обеспечения защиты компьютерных сетей.

Основное отличие SVM от нейросетей заключается в том, что для нейросети количество настраиваемых коэффициентов должно априорно задаваться пользователем на основании некоторых эвристических соображений. В методе опорных векторов количество настраиваемых параметров автоматически определяется во время настройки и обычно меньше, чем число векторов в обучающей последовательности. Ненулевыми остаются коэффициенты у опорных векторов, с помощью которых строится разделяющая гиперплоскость.

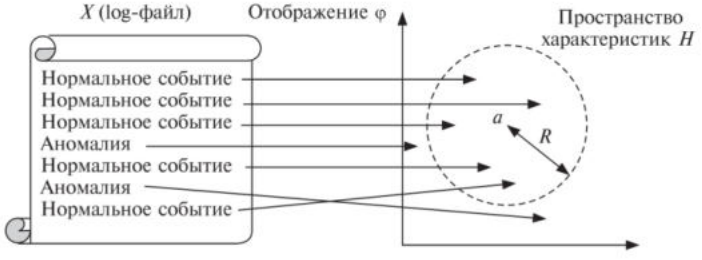

Рис. 7.3. Идея метода Single-class SVM применительно к задаче обнаружения вторжений в режиме обнаружения аномалий

Метод опорных векторов позволяет получить функцию классификации с минимальной верхней оценкой ожидаемого риска (уровня ошибки классификации). Он также делает возможным использовать линейный классификатор для работы с нелинейно разделяемыми данными [34].

Недостатком метода опорных векторов является неустойчивость по отношению к шуму в исходных данных. Шумовые выбросы обучающей выборки будут существенным образом учтены при построении разделяющей гиперплоскости [35].

Методы потенциальных функций для обнаружение аномалий. Рассмотрим обнаружение аномалий с помощью метода Single-class SVM. В этом методе определяется ПФ K, задающая отображение исходного множества анализируемых объектов в бесконечномерное пространство характеристик H. Далее в H ищется гиперсфера с центром в a минимального радиуса R, включающая «основную часть» υ образов объектов из исходного пространства. Это приводит к следующей оптимизационной задаче:

$$\min_{\xi\in\mathbf{R},\, R\in\mathbf{R},\, a\in H} \left[R^2 + \frac{1}{\upsilon N} \sum_{i=1}^{N} \xi_i \right],$$

при $\|\varphi(x_i) - a\|^2 \leqslant R^2 + \xi_i \;\forall i \in [1, N]$, которая может быть сведена к задаче квадратичного программирования. Здесь $\xi_i \geqslant 0$ — набор дополнительных переменных; вектор φ описывает отображение пространства параметров x в пространство характеристик H.

Идея метода визуально представлена на рис. 7.3 и опирается на гипотезу о компактности: в пространстве характеристик находится положение компактной области образов исходных объектов, которая отделяется от образов аномалий с помощью гиперсферы (a, R).

Для методов потенциальных функций существуют эффективные методы упрощения решающих функций, что позволяет им работать в реальном времени [7].

Методы ПФ имеют ряд преимуществ с точки зрения применимости для задач обнаружения вторжений:

- за счет использования различных потенциальных функций позволяют работать с разнородными сложно структурированными данными;
- имеют геометрическую интерпретацию и может быть произведена «подмена» потенциальных функций без замены самого алгоритма (так называемыми kernel trick), это означает, что в случае изменения структуры анализируемых данных достаточно заменить только используемую ПФ.

Основной недостаток метода — бинарная решающая функция, которая существенно зависит от параметра v, устанавливаемого априори. Переборные методы определения v неприемлемы для больших объемов данных, поскольку изменение v требует перестроения модели.

7.3. Обнаружение аномалий трафика с применением нейронных сетей

Среди параллельных методов распознавания, использующих евклидово пространство описаний образов, можно выделить широкий класс методов, заслуживающий отдельного рассмотрения, — нейросетевые методы. В их основе лежат нейронные сети — вычислительные модели, принцип функционирования которых сходен с сетями биологических нейронов головного мозга. Благодаря заимствованию принципов организации биологических структур мозга нейросети демонстрируют многие их свойства, такие, как обучение на основе предыдущего опыта, извлечение существенных свойств из поступающей информации, обобщение имеющихся прецедентов на новые случаи. Возможности, предоставляемые нейронными сетями, были использованы для решения задач распознавания и классификации образов во множестве исследований и прикладных разработок. В данном разделе рассматриваются некоторые основные разновидности нейронных сетей, их возможности и способы обучения применительно к обнаружению аномалий трафика.

Применение нейронных сетей обусловливается самой неформальной постановкой задачи — обнаружить аномальное поведение. Идея состоит в том, чтобы, получив некоторое «тренировочное» множество параметров вход-выход, характеризующее поведение системы, дать сети «привыкнуть» к ним. Выходом может быть некоторый коэффициент «нормальности» поведения или один из параметров системы. Если исходные данные имеют закономерности, то делается предположение, что сеть способна «научиться» на них. Если

в процессе работы предложенный нейронной сетью выход, при условии что он является некоторым коэффициентом, попадает в опасную область или отличается от имеющегося в реальной системе, если это один из параметров системы, то делается вывод, что в системе имеется аномалия.

Для построения шаблона поведения пользователя могут использоваться такие параметры, как время, когда он обычно работает, набор узлов, с которых он начинает рабочую сессию, характеристики использования ресурсов системы и т. п. Эти параметры оцифровываются и служат входом в нейронную сеть обратного распространения ошибки (backpropagation neuralnetwork, BPNN), а выходом является коэффициент, равный нулю для пользователя с нормальным поведением и равный единице — с аномальным. Иными словами, сеть тренируется на парах типа («нормальные» параметры, 0) и («аномальные» параметры, 1).

Поскольку для получения «ненормального» поведения надо было бы вынудить пользователя вести себя не так, как он привык, то аномальные данные генерируются случайно, что осложняет интерпретацию результатов относительно работы на реальных данных.

7.3.1. Выявление аномалий сетевой активности с применением аппарата искусственных нейронных сетей

В основе аппарата нейронных сетей лежит принцип подобия искусственного нейрона (ИН) биологическому прототипу. В простейшем варианте ИН — это бинарный пороговый элемент, который вычисляет сумму входных сигналов $z(t)$ и формирует на выходе сигнал 1, если эта сумма превышает определенный порог, и сигнал 0 в противном случае. График пороговой функции имеет вид, показанный на рис. 7.4.

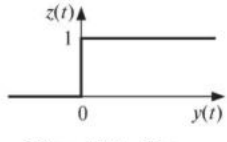

Рис. 7.4. График функции $z(t) = \text{sign}[y(t)]$

Пороговая функция является наиболее упрощенной активационной функцией ИН. Наиболее точную работу ИНС обеспечивают функция гиперболического тангенса (*а*), сигмоидальная (*б*) и логистическая (*в*) функции активации (рис. 7.5).

Совокупность ИН, сумматора и порогового элемента называется искусственной нейронной сетью (Artificial Neural Network, ANN) (рис. 7.6).

Сигнал поступает на входной слой рецепторных нейронов, каждый из которых связан со всеми элементами выходного слоя с определенными значениями весовых коэффициентов $w(i)$. Их сумми-

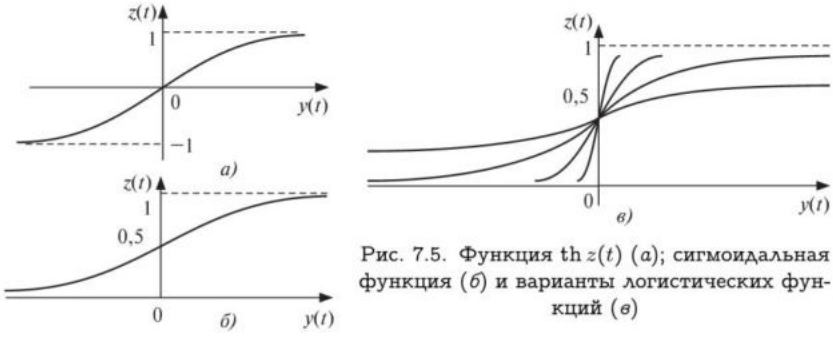

Рис. 7.5. Функция th $z(t)$ ($а$); сигмоидальная функция ($б$) и варианты логистических функций ($в$)

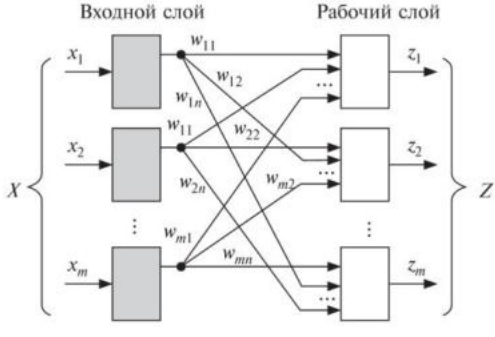

Рис. 7.6. Модель однослойной ИНС

рование приводит к возбуждению тех нейронов рабочего слоя, значение активационной функции которых превысило пороговое значение.

Существует множество различных моделей ИНС: с разным количеством рабочих (промежуточных) слоев, с обратными связями или без них, с полносвязными или произвольно связанными нейронами скрытых слоев.

Главным преимуществом нейронных сетей является возможность их обучения с целью создания гибких адаптивных систем управления. В том числе аппарат ИНС может успешно решать задачу выявления аномалий сетевой активности: обнаружение вторжений, нарушения правил работы в сети и любых нехарактерных для данной сети действий, которые невозможно выявить, используя распространенные системы обнаружения вторжений. Применение адаптивной системы предполагает построение профиля сети и отдельных пользователей в режиме нормального функционирования (отсутствие атак и других непредусмотренных действий). В этот период происходит обучение ИНС, сеть настраивается на желательные параметры и запоминает свое состояние. После введения систе-

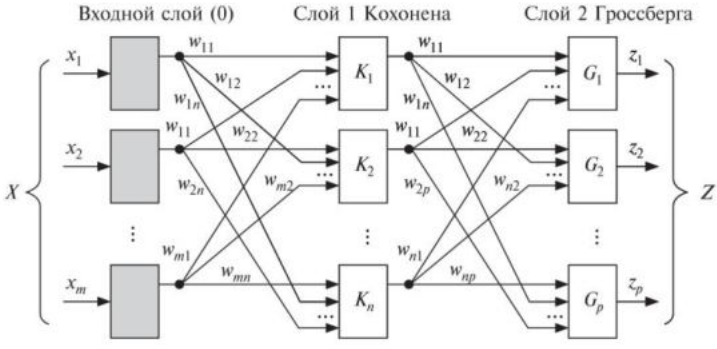

Рис. 7.7. Упрощенная версия сети встречного распространения

мы управления в эксплуатацию искусственная нейронная сеть будет выявлять все отклонения и нарушения в работе компьютерной сети, превышающие заданное при настройке ИНС пороговое значение.

Ключевой проблемой применения нейросетей для решения задач подобного типа является выбор параметров и метода обучения ИНС. Так как некорректный выбор структуры может привести к неполному или неточному обучению сети, попаданию ее в локальные минимумы или к отсутствию сходимости. Для создания систем обнаружения вторжений и выявления сетевых аномалий одним из возможных вариантов является использование ИНС встречного распространения, состоящей из двух слоев: слой ИН Кохонена и слой ИН Гроссберга. Первый слой обладает способностью извлекать из данных статистические свойства, второй обобщает и уточняет результаты. Важной особенностью является то, что подобные сети используют смешанное обучение: слой Кохонена обучается без учителя, нейроны слоя Гроссберга обучаются на результатах первого слоя с помощью комбинирования обратного распространения с обучением Коши. Нормализация векторов входных значений позволяет получить более точные результаты. Процесс нормализации представляет собой деление каждой компоненты входного вектора на его длину. Такая операция превращает входной вектор X в вектор единичной длины X_n в m-мерном пространстве.

В данной ИНС используется механизм латерального торможения (обострения входного сигнала). Если на слой нейронов, содержащий латеральные связи, подать входной вектор, имеющий небольшой максимум, то в процессе релаксации сети осуществляется повышение его контрастности (обострение). При большом значении максимума происходит сглаживание контрастности активационной функцией.

Такая структура нейронной сети позволяет создавать эффективные средства защиты компьютерных сетей с возможностью адаптации к изменяющимся условиям функционирования самой сети и требованиям к ее работе.

Работа с нейронной сетью предполагает наличие следующих этапов:

- сбор и подготовка исходных данных;
- построение и обучение сети;
- тестирование сети и анализ результатов.

7.3.2. Применение нейронных сетей в задачах обнаружения вторжений

Процесс обработки информации в СОВ приведен на рис. 7.8. Он включает три этапа.

Рис. 7.8. Процесс обнаружения

На первом этапе осуществляется захват трафика сети (feature selection). Сбор необходимых данных выполняет специальное программное средство (sniffer). Эти данные поступают в виде сетевых пакетов, заголовки которых содержат важную первичную информацию. Результаты первого этапа не могут быть сразу использованы классификатором, поскольку они представлены в «сыром» виде и нуждаются в предварительной обработке. Поэтому второй этап (feature preprocessing) связан с вычислением (на основе входных данных) параметров, характеризующих активность сети и представленных в той форме, в которой их сможет принять классификатор.

Для решения поставленных нами задач можно использовать базу данных KDD-99. Эта база содержит около 5 000 000 записей о соединениях. Каждая запись в этой базе представляет собой образ сетевого соединения. Соединение — последовательность TCP-пакетов за некоторое конечное время, моменты начала и завершения которого четко определены, в течение которого данные передаются от IP-адреса источника на IP-адрес приемника (и в обратном направлении), используя некоторый определенный протокол. Отдельная запись состоит из около 100 байтов, включает 41 параметр сетевого трафика и промаркирована как «атака» или «не атака». Например, первый параметр определяет длительность соединения, второй указывает используемый протокол, третий — целевую службу и т. д.

Третий этап состоит в обнаружении и распознавании атак (classification).

Мы предлагаем применять в качестве классификатора различные нейронные сети. После обучения нейронной сети такая СОВ способна выявлять возникающие в сети угрозы.

В базе KDD-99 представлены 22 типа атаки. При этом атаки делятся на четыре основные категории: DoS, U2R, R2L и Probe.

Атака DoS — отказ в обслуживании, характеризуется генерацией большого объема трафика, что приводит к перегрузке и блокированию сервера.

Атака U2R предполагает получение зарегистрированным пользователем привилегий локального суперпользователя (администратора).

Атака R2L характеризуется получением доступа незарегистрированного пользователя к компьютеру со стороны удаленной машины.

Атака Probe заключается в сканировании портов с целью получения конфиденциальной информации.

Выходные значения соответствуют четырем классам атак и нормальному состоянию сети.

7.3.3. Архитектурные решения СОВ

Рассмотрим различные архитектурные решения для построения систем обнаружения атак [44]. Они основаны на применении модулярных нейронных сетей. Основной задачей СОВ является обнаружение и распознавание атак. Для этих целей используется *многослойный персептрон* (Multilayer Perceptron, MLP), обучение которого осуществляется по правилу обратного распространения ошибки.

Второй важный вопрос, касающийся структуры СОВ: какие параметры входного вектора наиболее значимы для успешного обнаружения того или иного типа атаки. Обычно используют *рециркуляционную нейронную сеть* (Recirculation Neural Network, RNN) для получения главных компонент. Она представляется многослойным персептроном, который осуществляет линейное или нелинейное сжатие входных данных через «узкое горлышко» в скрытом слое. Такая сеть состоит из трех слоев (рис. 7.9).

Скрытый слой осуществляет сжатие входных образов. Значение j-го элемента скрытого слоя определяется по формулам

$$y_i = F(S_j); \quad S_j = \sum_{i=1}^{41} w_{ij} x_i,$$

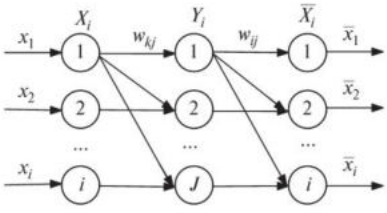

Рис. 7.9. Архитектура RNN

где F — функция активации; S_j — взвешенная сумма j-го нейрона; w_{ij} — весовой коэффициент между i-м нейроном входного и j-м нейроном скрытого слоя; x_i — i-й входной элемент.

Значения нейронных элементов выходного слоя определяются следующим образом:

$$\overline{x}_i = F(S_i); \quad S_i = \sum_{i=1}^{12} w'_{ji} y_j,$$

где w'_{ji} — весовой коэффициент между j-м нейроном скрытого и i-м нейроном выходного слоя; $\overline{x}_i$ — i-й выходной элемент.

Для обучения нелинейной RNN используются, как правило, два алгоритма — линейное правило обучения и алгоритм обратного распространения ошибки. Так, при алгоритме обратного распространения ошибки весовые коэффициенты модифицируются по следующим выражениям:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha\gamma_j F'(S_j) x_i;$$
$$w'_{ij}(t+1) = w'_{ij}(t) - \alpha\gamma_j F'(S_j)(\overline{x}_i - x_i),$$

где $F'(S_j)$ — производная нелинейной функции активации по взвешенной сумме;

$$\gamma_j = \sum_{i=1}^{41} (\overline{x}_i - x_i) F'(S_i) w'_{ji}$$

— ошибка j-го нейрона.

В процессе обучения весовые коэффициенты скрытого слоя ортонормируются в соответствии с процедурой Грамма–Шмидта.

Рассмотрим отображение входного пространства образов для нормального состояния и компьютерной атаки (тип атаки neptune) на плоскость двух первых главных компонент. Из рис. 7.10 видно, что данные, соответствующие разным классам, концентрируются в разных областях.

Комбинируя RNN и MLP нейронные сети, можно получать различные архитектуры систем обнаружения атак.

На рис. 7.11 приведена система обнаружения атак, которая состоит из рециркуляционной нейронной сети и многослойного персептрона, которые соединены последовательно. Задачей RNN является

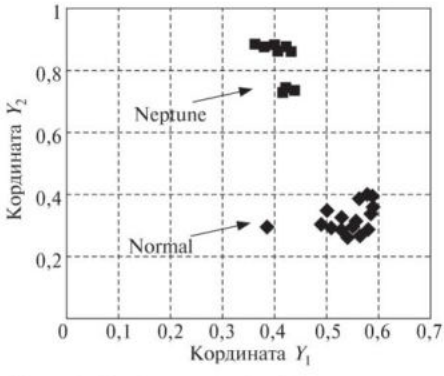

Рис. 7.10. Данные, обработанные нелинейной RNN

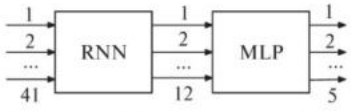

Рис. 7.11. Первый вариант СОВ

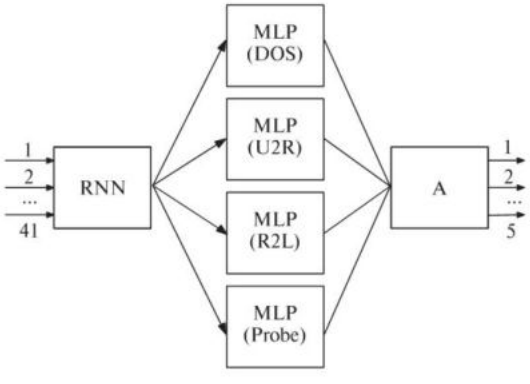

Рис. 7.12. Второй вариант СОВ

сжатие входного 41-размерного вектора в 12-размерный выходной вектор. Многослойный персептрон осуществляет обработку сжатого пространства входных образов (главных компонент) с целью распознавания класса атаки.

На рис. 7.12 приведена вторая схема системы обнаружения атак. Она характеризуется тем, что главные компоненты с выходов RNN одновременно поступают на четыре отдельных многослойных персептрона, каждый из которых соответствует определенному классу атаки: DoS, U2R, R2L и Probe. С выходов MLP данные поступают на арбитр, который и принимает окончательное решение о состоянии системы. В качестве арбитра может использоваться линейный или многослойный персептрон. Тогда обучение его будет производиться после обучения RNN и MLP. Такая схема может осуществлять иерархическую классификацию атак. В этом случае арбитр опреде-

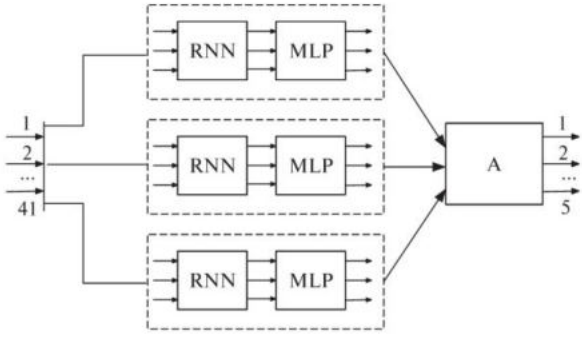

Рис. 7.13. Третий вариант СОВ

ляет один из пяти классов атаки, а соответствующий многослойный персептрон — тип атаки.

Следующий вариант структуры СОВ (рис. 7.13) основан на применении модулярной нейронной сети. Под модулярностью понимается разбиение сложной вычислительной задачи на множество небольших и простых задач, которые решаются отдельными модулями системы (экспертами). Далее заключения всех экспертов интегрируются в общее решение, которое имеет приоритет над решением каждого отдельного эксперта.

В качестве эксперта использовали модель 1 (см. рис. 7.11). Обучение каждого эксперта происходит на отдельном множестве данных, т. е. данные для обучения последующего эксперта формируются с учетом результатов обучения предыдущих экспертов. Алгоритм, используемый для такого обучения, называют *алгоритмом усиления за счет Фильтрации* (Boosting by Filtering). После обучения нейронные сети способны обнаруживать атаки.

7.3.4. Результаты экспериментов

Чтобы оценить эффективность предложенных подходов обнаружения вторжений, был проведен ряд экспериментов. База данных KDD Cup 99 использовалась для обучения и тестирования нейросетевых моделей. Это одна из тех немногих баз в области обнаружения вторжений, которая привлекает внимание исследователей благодаря своей хорошо продуманной структуре и доступности.

Для изучения характеристик предложенных систем мы задались тремя основными показателями: доля обнаруженных, доля распознанных атак по каждому классу и число ложных срабатываний системы. Доля обнаруженных атак определяется как число образов атак отдельного класса, обнаруженных системой, деленное на общее

Таблица 7.1

Результаты тестирования модели 1

Класс	Всего	Обнаружено	Распознано
DoS	391458	391441 (99,99 %)	370741 (94.71%)
U2R	52	48 (92,31 %)	42 (80,77%)
R2L	1126	1113 (98,85 %)	658 (58,44%)
Probe	4107	4094 (99,68 %)	4081 (99,37 %)
Нормальное состояние			
Normal	97277	–	50831 (52,25%)

Таблица 7.2

Сводные данные по результатам тестирования каждой модели

Модель	Обнаруженные атаки	Распознанные атаки	Ложные атаки	Общая доля распознанных, %
1	396696 (99,98 %)	375522 (94,65 %)	46446 (47,75 %)	86,30
2	395949 (99,80 %)	375391 (94,61 %)	13398 (13,77 %)	92,97
3	396549 (99,95 %)	375730 (94,70 %)	12549 (12,90 %)	93,21

количество записей об атаках этого класса в базе данных. Подобным образом определяется и доля распознанных. Ложные срабатывания указывают общее число образов нормальной работы сети, ошибочно классифицированных как атаки.

Рассмотрим функционирование системы на примере модели 1. Эта модель достаточно проста. Результаты тестирования в режиме распознавания класса атаки приведены в табл. 7.1.

Таким образом, наилучший результат был достигнут для атак класса DoS и Probe (почти однозначная распознаваемость). Несколько хуже определяются U2R и R2L, соответственно 80,77% и 58,44%. Кроме того, существует процент ложных срабатываний системы. Сводные данные по каждому из вариантов построения системы обнаружения атак приведены в табл. 7.2.

Таким образом, модель 3 характеризуется высокой точностью (93,21 %) и наименьшим числом ложных срабатываний. При использовании модели 1 были распознаны 86,3 % входных образов, а модели 2 — 92,97 %. Модели 2 и 3 могут успешно применяться для работы с большими наборами сложных по структуре данных.

Недостатки:

- топология сети и веса узлов определяются только после огромного числа проб и ошибок;
- размер окна — еще одна величина, которая имеет огромное значение при разработке; если сделать окно маленьким, то сеть будет не достаточно производительной, слишком большим — будет страдать от неуместных данных.

Преимущества:

- успех данного подхода не зависит от природы исходных данных;
- нейронные сети легко справляются с зашумленными данными;
- автоматически учитываются связи между различными измерениями, которые, несомненно, влияют на результат оценки.

7.4. Методы искусственного интеллекта в задачах обеспечения безопасности компьютерных сетей

7.4.1. Многоагентные системы

Долгое время ведутся разработки персональных средств защиты, предназначенных для одной единицы оборудования [35–38]. Так как защита сети — задача комплексная, то помимо средств персональной защиты (антивирусы, сетевые экраны и т. д.) ведутся разработки многоагентных систем. Каждый агент отвечает за определенную часть задания, и общее решение возникает в результате их скоординированных действий. Агенты могут иметь реализованные элементы интеллекта, а могут и не иметь таковых. В процессе работы агенты обмениваются сообщениями по специальным протоколам. Обычно существует агент, который управляет действиями других. Перечислим особенности многоагентных систем, которые позволяют эффективно их использовать в системах сетевой безопасности:

Гибкость. Агенты могут создавать себе подобных и размещать их на новых узлах сети, поэтому многоагентные системы легко адаптируются к любой сетевой архитектуре и адекватно отвечают на изменения в конфигурации сетевого оборудования.

Экономичность. Система равномерно распределена по всему периметру защиты. Эта особенность обеспечивает оптимальное распределение вычислительных ресурсов сети.

Повышенная отказоустойчивость. Так как агенты могут существовать самостоятельно и распределены на всех узлах сети, т. е. система защиты не имеет центра, то атаковать ее будет сложнее, нежели сеть с централизованным сервером защиты. Распределенная по сети информация и распределенная защита требуют от злоумышленника проводить атаку многих узлов одновременно.

Возможность централизованного администрирования. Внесение изменений в работу агентов может производиться централизованно и по протоколам взаимодействия агентов передаваться на все точки обеспечения безопасности.

Многоагентный подход подробно рассмотрен в [36, 37]. В [35, 38] рассмотрены многоагентные системы защиты сети от внешних

угроз. Особенности подхода, описанного в этих работах, обеспечивают защиту от сложных угроз и позволяют наглядно представить текущее состояние всех агентов и сети в целом. Агенты разделены по роду деятельности и объединены в команды. Например, в [38] выделены следующие классы агентов команд защиты: обработки информации (*сэмплеры*), обнаружения атаки (*детекторы*), фильтрации (*фильтры*), *агенты расследования*. Сэмплеры осуществляют сбор данных для последующего обнаружения сетевых аномалий или злоупотреблений детектором. Фильтры ответственны за фильтрацию трафика по правилам, представленным детектором. Агент расследования пытается обезвредить агентов атаки. Команда агентов защиты совместно реализует механизм защиты и может взаимодействовать по различным схемам. В одной из схем при обнаружении начала атаки действует детектор той команды, на чью сеть направлена атака. Он посылает запрос агентам-сэмплерам других команд с целью получения информации, которая может быть релевантной указанной атаке. Сэмплеры других команд отвечают на запрос, посылая необходимые данные. Эта информация существенно повышает шансы на обнаружение атаки. В случае обнаружения вероятного источника атаки детектор сети-жертвы посылает информацию об адресе агента атаки детектору команды, в сети которой может находиться этот агент, с целью его деактивации. Таким образом, использование многоагентного подхода и интеллектуальных алгоритмов обработки данных при разработке систем обеспечения безопасности компьютерных сетей значительно повышает их качественные характеристики.

7.4.2. Системы анализа защищенности

Необходимым элементом системы безопасности компьютерной сети является регулярный анализ ее защищенности. В зависимости от требуемого качества проводимой проверки можно проводить сканирование либо зондирование системы [39]. Объем рабочей памяти может изменяться, т. е. увеличиваться (это происходит чаще) или уменьшаться по мере применения правил. *Механизм вывода* служит для реализации логического вывода путем просмотра правил и фактов, нахождения соответствия между ними и изменения рабочей памяти. В случае, если левая часть продукции оказывается истинной, происходит *срабатывание* продукции и возникает событие одного из двух типов: 1) получение нового знания — в рабочую память добавляется факт из правой части продукции; 2) выполнение некоторого действия по изменению конфигурации компьютерной сети. В области сетевой безопасности продукционные системы стали

использоваться для обнаружения известных уязвимостей в проверяемой системе по формальным признакам, выявленным экспертами. Производители экспертных систем по сетевой безопасности формируют и поддерживают базу данных эвристик в Интернете для поддержания системы в актуальном состоянии.

Например, в бесплатной утилите *AVZ* [39] реализована функция эвристической проверки системы, которая позволяет проводить поиск известных Spyware* и вирусов по косвенным признакам — на основании анализа реестра, файлов на диске и в памяти. Регулярно выходят обновления программы, в которых содержится обновленный набор эвристик. Приведем пример продукционного правила:

ЕСЛИ процесс использует библиотеки для работы с сетью И количество обнаруженных сигнатур, типичных для отправки почты > X, ТО записать в рабочую память факт «программа работает с электронной почтой».

В данном случае выполнение составного условия приводит к занесению в рабочую память нового факта, который может быть использован другими правилами. К преимуществам продукционного подхода можно отнести устойчивость таких систем и точность обнаружения известных типов угроз сетевой безопасности. К недостаткам относят поражение сети атаками *zero-day* (атака через уже обнаруженные, но еще не закрытые бреши в программных продуктах). Кроме того, в связи с частым появлением новых угроз эксперты должны ежедневно работать над качественным пополнением базы знаний. Поскольку в области сетевой безопасности продукционные системы применяются для обнаружения известных угроз, количество которых возрастает с каждым днем, то подход становится все менее эффективным, ввиду необходимости регулярной передачи набора новых прав.

7.5. Методы искусственных иммунных систем и нейронных сетей для обнаружения компьютерных атак

Идея создания искусственных иммунных систем появилась в результате изучения процессов биологического иммунитета, который защищает организм от болезнетворных бактерий и вирусов, обнаруживая и уничтожая их.

---

* Spyware — шпионское программное обеспечение, предназначенное для слежения за действиями пользователя на его компьютере.

Биологическая иммунная система представляет собой сложную адаптивную структуру, состоящую из различных органов и компонентов, которая для защиты биологического организма от внешних бактерий и вирусов использует разнообразные иммунные механизмы, такие, как производство иммунокомпетентных клеток; их обучение и отбор; обнаружение вредоносных бактерий и вирусов; уничтожение обнаруженных вирусов; механизмы адаптации, механизмы иммунной памяти и т. д. Основной целью иммунной системы является распознавание чужеродных клеток и бактерий в организме и уничтожение их. Иммунная реакция заключается в стимуляции различных механизмов при обнаружении вредоносных бактерий, направленных на их уничтожение. Следует отметить, что иммунная система способна распознавать не только уже известные ей бактерии и вирусы, но и также не известные, ранее не встречающиеся [19].

В результате проведенного анализа биологической иммунной системы был сделан вывод, что данная система является надежным механизмом обнаружения аномалий в виде болезнетворных бактерий и вирусов. Такая система характеризуется способностью к классификации объектов различного класса, а также наличием механизмов борьбы с обнаруженными инфекциями. Благодаря своим особенностям и характеристикам, иммунная система представляет большой интерес в области обработки массивов данных и защиты информации. Перечисленные характеристики и возможности доказывают перспективность использования основных концепций иммунитета в решении сложных компьютерных задач, таких, например, как задач обеспечение информационной безопасности.

7.5.1. Построения искусственной иммунной системы для обнаружения компьютерных атак

При построении искусственной иммунной системы для обнаружения и классификации сетевых атак на компьютерные системы мы основывались на базовых принципах и механизмах биологической иммунной системы, а также на типовой схеме искусственной иммунной системы, предложенной в [40]. Это такие механизмы, как генерация и обучение иммунных детекторов, отбор детекторов, которые по каким-либо причинам генерируют ложные решения, функционирование детекторов (рис. 7.14).

Иммунные детекторы генерируются по случайному алгоритму, что дает возможность создания большого количества разнообразных по своей структуре детекторов, которые способны реагировать на любую аномалию. Далее детекторы проходят стадию обучения,

Рис. 7.14. Жизненный цикл детекторов иммунной системы

на которой они приобретают способность корректно реагировать на чужеродные объекты или явления. Для того чтобы детекторы не генерировали ложные срабатывания, они тщательно отбираются. Те из них, которые не обучились корректно классифицировать объекты, удаляются. Отобранные детекторы допускаются к выполнению функций по классификации объектов.

Каждому детектору выделяется некоторое лимитированное количество времени (время жизни), на протяжении которого он может существовать.

Если на протяжении этого времени детектор не обнаруживает аномалий, то он удаляется, а на его место приходит новый, структурно отличный детектор. Если детектор обнаружил аномалию, происходит так называемая стадия активации. На этой стадии происходит информирование об обнаруженной аномалии и ее уничтожение. Детектор, обнаруживший аномалию, трансформируется в детектор иммунной памяти. Детекторы иммунной памяти характеризуются большим временем жизни и уровнем доверия.

Как видно из рис. 7.14, искусственная иммунная система моделирует основные процессы биологической иммунной системы, а также их взаимодействие. Отличие заключается в способе представления информации и структуре иммунного детектора.

7.5.2. Метод функционирования иммунных нейросетевых детекторов

Рассмотрим основные положения адаптированного механизма искусственной иммунной системы с целью повышения качества обнаружения компьютерных атак.

В [41–43] представлена методика обучения и функционирования нейронной сети для обнаружения сетевых атак на компьютерные се-

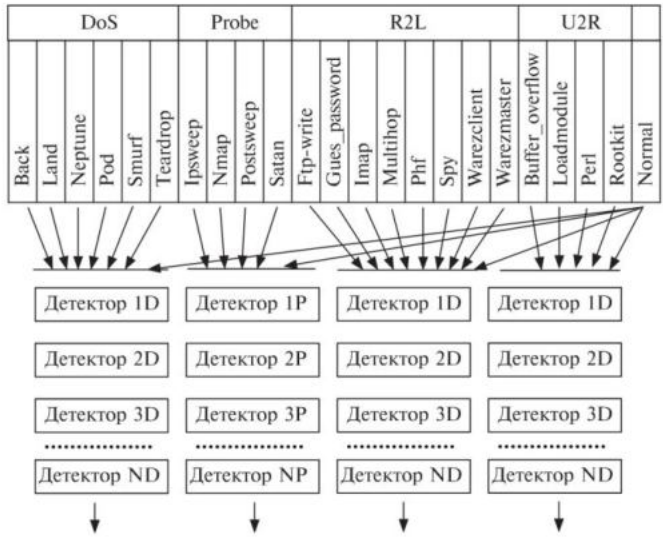

Рис. 7.15. Метод обучения иммунных детекторов

ти. В качестве иммунного детектора выберем разработанную нейронную сеть. Такая нейронная сеть характеризуется разделением нейронов в скрытом слое Кохонена, отвечающих за разные классы — сетевые атаки и нормальные соединения. Итак, иммунные детекторы, проходят следующие этапы в своем развитии.

Генерация иммунных детекторов. На данном этапе происходит создание нейронных сетей с начальной инициализацией весовых коэффициентов согласно случайному распределению. Также, как и в биологическом иммунитете, искусственные иммунные детекторы должны пройти процесс обучения для того, чтобы корректно выполнять задачи классификации сетевого трафика.

Обучение иммунных детекторов. На данном этапе сгенерированные нейронные сети подвергаются процессу обучения. Как было показано в [41], для обучения разработанной нейронной сети осуществляется контролируемое конкурентное обучение в соответствии с правилом «победитель берет все» [44, 45]. Однако методика обучения иммунного детектора несколько отличается от той, которая была представлена нами ранее. Рассмотрим подробнее предлагаемый метод обучения для иммунных детекторов (рис. 7.15), в основе которых лежит выбранная нейронная сеть.

Как известно, в сетевом трафике можно выделить набор параметров. Для обучения нейронной сети и для анализа сетевого трафика посредством обученной нейронной сети выделяем 41 параметр

из сетевого трафика [46]. Однако, если для обучения нейросетевого детектора используется обучающая выборка, состоящая из какого-то отдельного подкласса компьютерной атаки (например back или land, принадлежащих классу DoS-атак, или, например, satan, который принадлежит к классу Probe-атак) и обучается столько детекторов, сколько существует разновидностей сетевых атак (22 детектора на каждый из подкласс компьютерных атак), то в случае обучения иммунного нейросетевого детектора обучающая выборка формируется из параметров всего класса атак (например, DoS-атаки или R2L-атаки). Кроме того, для обнаружения атак определенного класса в данном случае используется не один обученный иммунный нейросетевой детектор, а несколько.

В общем случае методику создания и обучения иммунных нейросетевых детекторов можно представить как последовательность шагов в следующем виде:

Шаг 1. Создаем нейронную сеть со случайной инициализацией весовых коэффициентов.

Шаг 2. Случайным образом из набора параметров сетевых соединений выбираем N соединений, относящихся к определенному классу компьютерных атак, и M соединений, относящихся к классу нормального трафика.

Шаг 3. Последовательно подаем выбранные параметры соединений на нейронную сеть и в зависимости от поданных данных и данных на выходе нейронной сети корректируем весовые коэффициенты.

Весовые коэффициенты корректируются согласно следующему:

- вычисляется евклидово расстояние между входным образом и весовыми векторами нейронных элементов слоя Кохонена:

$$D_i = |X - w_i| = \sqrt{(X_1 - \omega_{1i})^2 + (X_2 - \omega_{2i})^2 + (X_n - \omega_{ni})^2}, \tag{7.8}$$

  где $i = \overline{1, m}$. Определяется нейронный элемент победитель с номером k:

$$D_k = \min_j D_j; \tag{7.9}$$

- производится модификация весовых коэффициентов нейрона в соответствии с выражением

$$\omega_{ck}(t+1) = \omega_{ck}(t) + \gamma(X_c - \omega_{ck}(t)). \tag{7.10}$$

Если при подаче на вход сети параметров сетевой атаки победителем является один из первых p нейронов нейронной сети или при

подаче на вход сети параметров нормального соединения победителем является один из r последних нейронов сети Кохонена, то

$$\omega_{ck}(t+1) = \omega_{ck}(t) - \gamma(X_c - \omega_{ck}(t)). \tag{7.11}$$

В противном случае процесс повторяется, начиная с шага 3 для всех входных образов.

Обучение нейронной сети производится до желаемой степени согласования между входными и весовыми векторами, т. е. до тех пор, пока значение суммарной квадратичной ошибки не станет равной нулю:

$$E_i = \frac{1}{2}\sum_{k=1}^{L}\sum_{j=1}^{2}(z_{ij}^k - l_{ij}^k)^2, \tag{7.12}$$

и весь процесс повторяется до тех пор, пока количество обученных иммунных детекторов не станет равным заданному значению L.

Таким образом, создается набор детекторов для анализа сетевого трафика с целью обнаружения компьютерных атак. Однако перед тем, как выполнять анализ сетевого трафика, обученные детекторы необходимо проверить на корректность классификации с целью предотвращения возникновения ложных срабатываний. Для этого все обученные детекторы проходят стадию отбора.

Отбор иммунных детекторов. Для минимизации возникновения ложных срабатываний, когда нормальное соединение принимается за компьютерную атаку, все обученные иммунные нейросетевые детекторы проходят проверку на корректность классификации. Для этого на нейронную сеть подается заранее созданная тестовая выборка, состоящая из параметров нормального соединения. Если i-й детектор классифицирует одно из тестовых соединений как атаку, то он уничтожается, а вместо него генерируется и обучается новый детектор. Если i-й детектор не генерирует ложные срабатывания на тестовой выборке, то он считается корректным и допускается к анализу входящего и исходящего сетевого трафика.

Функционирование иммунных детекторов. Детекторы, которые допущены к анализу сетевого трафика, образуют систему обнаружения компьютерных атак. Весь трафик, получаемый компьютером, сначала анализируется совокупностью иммунных детекторов, и, если ни один из детекторов не обнаруживает аномалию, то трафик обрабатывается операционной системой и соответствующим программным обеспечением. Описанная система анализа сетевого трафика, состоящая из совокупности иммунных детекторов, изображена на рис. 7.16.

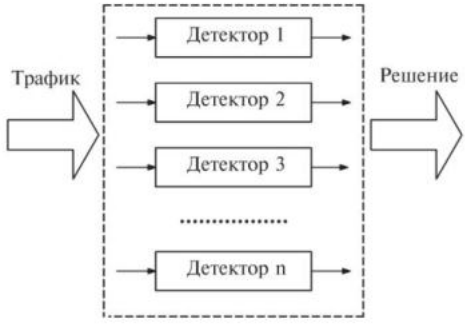

Рис. 7.16. Набор детекторов для анализа сетевого трафика с целью обнаружения компьютерных атак

Каждый детектор наделяется временем жизни, на протяжении которого он анализирует сетевой трафик. Если по окончании выделенного времени детектор не обнаружил аномалию, он уничтожается, а на его место приходит новый детектор. Механизм наделения детекторов временем жизни позволяет избавляться от детекторов, которые хоть и прошли успешно стадии обучения и отбора, однако из-за своей структурной особенности (набор весовых коэффициентов) являются малопригодными.

Активация иммунных детекторов. Активация детекторов подразумевает обнаружение детектором сетевой атаки. В случае, когда сетевое соединения классифицируется одним или несколькими детекторами как компьютерная атака, происходит его блокировка, т. е. оно не допускается к обработке операционной системой и специализированным программным обеспечением. Также выдается сообщение пользователю о попытке атаки компьютерной системы.

Формирование иммунной памяти. При обнаружении и блокировании сетевой атаки целесообразно сохранять ее параметров с целью изучения и детального анализа. Дело в том, что иммунные нейросетевые детекторы обучаются на ограниченном наборе данных, которые не могут включать в себя все вероятные компьютерные атаки. Для того чтобы повысить качество обнаружения, а также наделить систему обнаружения вторжений гибкостью и позволить ей адаптироваться под современные реалии, параметры сетевого соединения, классифицированного как атака, сохраняются и заносятся в обучающую выборку, тем самым пополняя ее актуальными данными. Детекторы, которые будут создаваться с целью заменить «устаревшие» иммунные детекторы, будут уже обучаться также и на новых данных, что позволит значительно увеличить качество обнаружения. Кроме этого, создается новая нейронная сеть, которая

обучается исключительно на данных, выделенных из обнаруженной сетевой атаки, и которая вводится в систему анализа сетевого трафика. Это позволит более точно выделить данную атаку при повторной подобной атаке на защищаемую компьютерную систему со стороны злоумышленника. Совокупность детекторов иммунной памяти будет хранить в себе информацию обо всех сетевых атаках, направленных в прошлом на защищаемую компьютерную систему, и обеспечивать высокий уровень реагирования на повторные попытки атак.

7.5.3. Алгоритм функционирования системы обнаружения вторжений на базе искусственных иммунных систем и нейронных сетей

Общий алгоритм функционирования СОВ на базе искусственных иммунных систем и нейронных сетей можно представить в следующем виде: создание иммунного детектора на базе многослойной нейронной сети с одним скрытым слоем Кохонена и начальная инициализация весовых коэффициентов.

Формируем обучающую выборку для созданного детектора. Обучающая выборка формируется выбором случайным образом n соединений, относящихся к определенному классу сетевой атаки, и m соединений, относящихся к нормальному трафику.

Последовательно подаем данные из обучающей выборки на нейронную сеть и обучаем ее согласно правилу «победитель берет все» и формулам (7.8)–(7.12).

Обученный детектор проверяется на тестовой выборке, состоящей из параметров нормальных соединений. Если детектор корректно классифицирует предоставленные данные, то он «допускается» к анализу сетевого трафика в реальном режиме. Если же детектор классифицирует предоставляемые тестовые данные как сетевую атаку, то он уничтожается. Внедрение обученного и отобранного детектора в подсистему анализа сетевого трафика.

Если отведенное время жизни детектора истекло и детектор не обнаружил аномалию в сетевом трафике, то он уничтожается и процесс начинается с шага 1 (см. стр. 198).

Если детектор обнаружил аномалию, то происходит его активация. Сетевое соединение, классифицируемое как атака, блокируется, а пользователю выдается сообщение.

Выделение и анализ параметров сетевого соединения, классифицированного как атака и занесение выделенных параметров в базу для обучения новых иммунных детекторов.

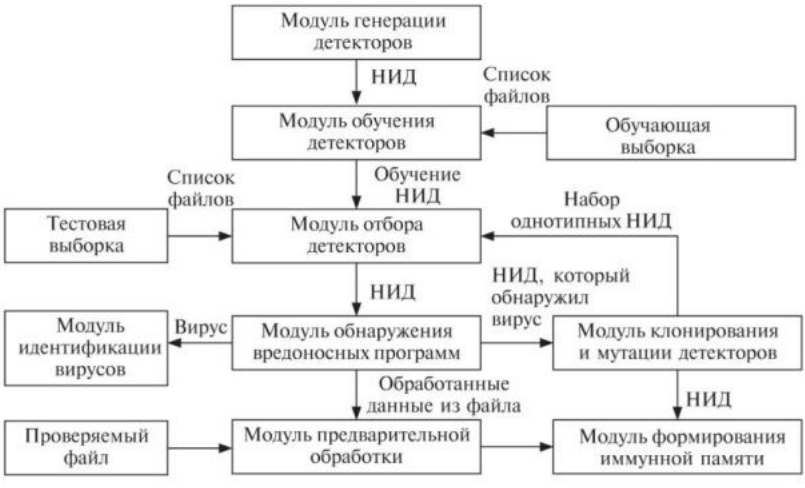

Рис. 7.17. Структура нейросетевой искусственной иммунной системы

Обучение детектора иммунной памяти на параметрах обнаруженной сетевой атаки с целью повышения качества обнаружения возможных аналогичных атак на защищаемую систему и внедрение данного детектора в подсистему анализа сетевого трафика.

Рассматривается интеллектуальная система обнаружения компьютерных вирусов [47, 48], которая базируется на применении искусственных иммунных систем и нейронных сетей. Такая система использует основные принципы функционирования биологической иммунной системы, где в качестве отдельного детектора используется нейронная сеть.

На рис. 7.17 представлена структура нейросетевой искусственной иммунной системы для обнаружения вредоносных программ. Она состоит из следующих основных модулей: модуль генерации детекторов, модуль обучения иммунных детекторов, модуль отбора детекторов, модуль уничтожения детекторов, модуль обнаружения вредоносных программ, модуль идентификации вирусов, модуль клонирования и мутации детекторов, модуль формирования иммунной памяти.

Нейросетевые иммунные детекторы играют ключевую роль в обнаружении вредоносных программ. Пройдя стадии обучения и отбора, детекторы приобретают способность реагировать на вредоносные программы, сканируя их структуру, и в то же время игнорировать чистые файлы.

Функционирование нейросетевого иммунного детектора заключается в последовательности следующих действий:

1) случайным образом из списка существующих файлов нейросетевой иммунный детектор выбирает файл для проверки;

2) данные из файла проходят предварительную обработку, связанную с удалением «дыр» и незначащих нулей;

3) по методу скользящего окна детектор сканирует файл и принимает решение о принадлежности проверяемого файла к классу чистых или к классу вредоносных программ;

4) если нейросетевой иммунный детектор принимает решение о принадлежности проверяемого файла к классу чистых программ, детектор выбирает новый файл для проверки;

5) если нейросетевой иммунный детектор принимает решение о принадлежности проверяемого файла к классу вредоносных программ, он подает сигнал об обнаружении компьютерного вируса и нейросетевая иммунная система переходит в особый режим функционирования.

7.6. Визуальный анализ данных

7.6.1. Анализ методов визуализации

Существует множество способов поиска скрытых закономерностей в данных машиной, алгоритмами, но также не стоит упускать из вида возможности человека по анализу данных [51, 52]. Полезно сочетать огромные вычислительные ресурсы современных компьютеров с творческим и гибким человеческим мышлением. Визуальный анализ данных призван вовлечь человека в процесс отыскания знаний в данных.

Основная идея заключается в том, чтобы представить большие объёмы данных в такой форме, где человек мог бы увидеть то, что трудно выделить алгоритмически. Чтобы человек смог погрузиться в данные, работать с их визуальным представлением, понять их суть, сделать выводы и напрямую взаимодействовать с данными. Из-за сложности информации это не всегда возможно и в простейших графических видах представления знаний, таких, как деревья решений, дейтаграммы, двумерные графики и т. п. В связи с этим возникает необходимость в более сложных средствах отображения информации и результатов анализа.

С помощью новых технологий пользователи способны оценивать: большие объекты и маленькие, далеко они находятся или близко. Пользователь в реальном времени может двигаться вокруг объектов или кластеров объектов и рассматривать их со всех сторон. Это позволяет использовать для анализа естественные че-

ловеческие перцепционные навыки в обнаружении неопределённых образцов в визуальном трёхмерном представлении данных.

Визуальный анализ данных особенно полезен, когда о самих данных мало что известно и цели исследования до конца не понятны. За счёт того, что пользователь напрямую работает с данными, представленными в виде визуальных образов, которые он может рассматривать с разных сторон и под любыми углами зрения, в прямом смысле этого слова, он может получить дополнительную информацию, которая поможет ему более чётко сформулировать цели исследования. Таким образом, визуальный анализ данных можно представить как процесс генерации гипотез. При этом сгенерированные гипотезы можно проверить или автоматическими средствами (методами статистического анализа или методами Data Mining), или средствами визуального анализа. Кроме того, прямое вовлечение пользователя в визуальный анализ имеет два основных преимущества перед автоматическими методами:

- визуальный анализ данных позволяет легко работать с неоднородными и зашумлёнными данными, в то время как не все автоматические методы могут работать с такими данными и давать удовлетворительные результаты;
- визуальный анализ данных интуитивно понятен и не требует сложных математических или статистических алгоритмов.

Визуальный анализ данных обычно выполняется в три этапа:

1) беглый анализ — позволяет идентифицировать интересные шаблоны и сфокусироваться на одном или нескольких из них;

2) увеличение и фильтрация — идентифицированные на предыдущем этапе шаблоны отфильтровываются и рассматриваются в большем масштабе;

3) детализация по необходимости - если пользователю нужно получить дополнительную информацию, он может визуализировать более детальные данные.

Характеристики средств визуализации данных существует достаточно большое количество средств визуализации данных, предоставляющих различные возможности.

Для выбора таких средств рассмотрим более подробно три основные характеристики средств визуализации данных:

1) характер данных, которые нужно визуализировать с помощью данного средства;

2) методы визуализации и образцы, в виде которых могут быть представлены данные;

3) возможности взаимодействия с визуальными образами и методами для лучшего анализа данных.

Выделяют следующие виды данных, с которыми могут работать средства визуализации:

- одномерные данные — одномерные массивы, временные ряды и т. п.;
- двумерные данные — точки двумерных графиков, географические координаты и т. п.;
- многомерные данные — финансовые показатели, результаты экспериментов и т. п.;
- тексты и гипертексты — газетные статьи, веб-документы и т. п.;
- иерархические и связанные — структура подчинённости в организации, электронная переписка людей, гиперссылки документов и т. п.;
- алгоритмы и программы — информационные потоки, отладочные операции и т. п.

Для визуализации перечисленных типов данных используются различные визуальные образы и методы их создания. Очевидно, что количество визуальных образов, которыми могут представляться данные, ограничиваются только человеческой фантазией. Основное требование к ним — наглядность и удобство анализа данных, которые они представляют. Методы визуализации могут быть как самые простые (линейные графики, диаграммы, гистограммы и т. п.), так и более сложные, основанные на сложном математическом аппарате. Кроме того, при визуализации могут использоваться комбинации различных методов. Выделяют следующие типы методов визуализации:

- стандартные 2D/3D-образы — гистограммы, линейные графики и т. п.;
- геометрические преобразования — диаграмма разброса данных, параллельные координаты и т. п. (рис. 7.18, 7.19);
- отображение иконок — линейчатые фигуры (needle icons) и звёзды (star icons);
- методы, ориентированные на пиксели, — рекурсивные шаблоны, циклические сегменты и т. п.;
- иерархические образы — древовидные карты и наложение измерений.

К простейшим методам визуализации относятся графики, диаграммы, гистограммы и т. п. Основным их недостатком является невозможность приемлемой визуализации сложных данных и большого количества данных. Методы геометрических преобразований

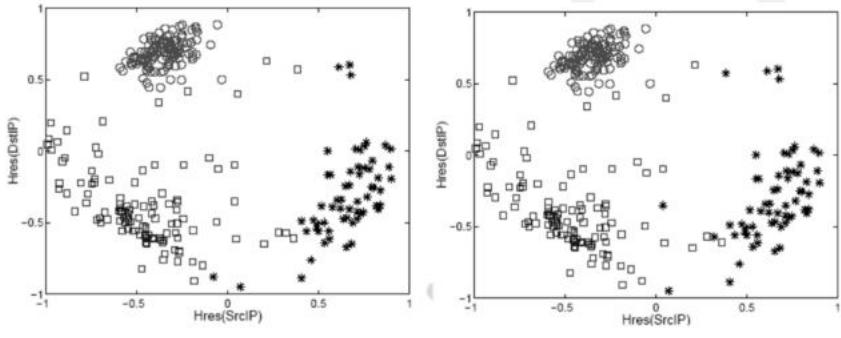

Рис. 7.18. Кластеризация типов аномалий (2-D): круги — сканирование; квадраты — DoS-атаки; звездочки — DDoS-атаки

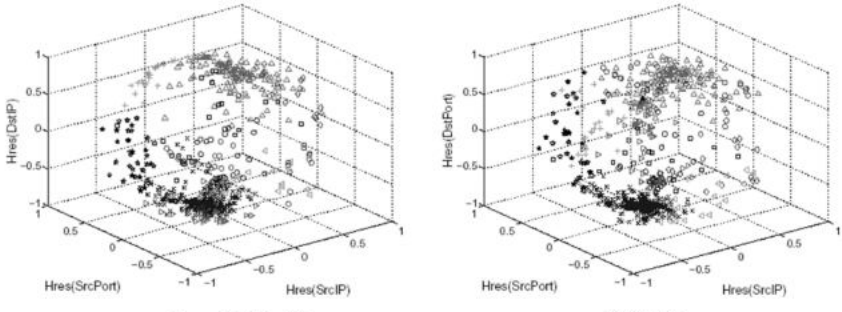

Рис. 7.19. Кластеризация типов аномалий (3-D)

визуальных образов направлены на трансформацию многомерных наборов данных с целью отображения их в декартовом и в недекартовом геометрических пространствах.

Данный класс методов включает в себя математический аппарат статистики. Другим классом методов визуализации данных являются методы отображения иконок. Их основной идеей является отображение значений элементов многомерных данных в свойства образов. Такие образы могут представлять собой: человеческие лица, стрелки, звёзды и т. п. Визуализация генерируется отображением атрибутов элементов данных в свойства образов. Такие образы можно группировать для целостного анализа данных. Результирующая визуализация представляет собой шаблоны текстур, которые имеют различия, соответствующие характеристикам данных. Основной идеей методов, ориентированных на пиксели, является отображение каждого измерения значения в цветной пиксель и из группировки по принадлежности к измерению. Так как один пиксель используется для отображения одного значения, то, следовательно, данный ме-

тод позволяет визуализировать большое количество данных (свыше одного миллиона значений).

Методы иерархических образов предназначены для представления данных, имеющих иерархическую структуру. В случае многомерных данных должны быть правильно выбраны измерения, которые используются для построения иерархии.

7.6.2. Использование преобразования Хафа для обнаружения аномалий трафика

В задачах компьютерного зрения при анализе изображений зачастую бывает необходимо выделять в изображении различные кривые (прямые, окружности и т. д.). Для решения данной задачи существует целое семейство методов, основанных на преобразовании Хафа. Теоретические возможности этого преобразования позволяют не ограничиваться плоскостью и дискретными кривыми, его можно применять для поиска кривых в облаке точек на плоскости или в многомерном пространстве [51, 52].

Преобразование Хафа — это метод, позволяющий выделять в двоичном изображении графические объекты теоретически любой формы. Он основывается на представлении искомого объекта в виде параметрического уравнения, поэтому на практике используется для поиска простейших форм — прямых, окружностей, эллипсов и т. д. Вычислительная сложность алгоритма преобразования быстро растет с увеличением сложности графического объекта, т. е. его аналитического представления.

Алгоритм метода. Исходное изображение должно представлять собой множество точек двух типов: фоновых точек и точек интереса. Рассмотрим семейство таких кривых, заданных параметрическим уравнением $F(a_1, a_2, ..., a_n, x, y) = 0$, где x, y — координаты на плоскости, a — параметры семейства кривых. Эти параметры образуют фазовое пространство, каждая точка которого соответствует некоторой кривой. В нем вводится дискретная сетка, разбивающая его на ячейки, каждая из которых соответствует набору кривых с близкими значениями параметров.

Финальным шагом является обход ячеек пространства Хафа и выбор максимальных значений, за которые «проголосовало» больше всего точек изображения, что и даёт параметры для уравнений искомого объекта.

В качестве примера рассмотрим параметрическое уравнение прямой в полярной системе координат:

$$F(R, \theta, x, y) = x \cos\theta + y \sin\theta - R,$$

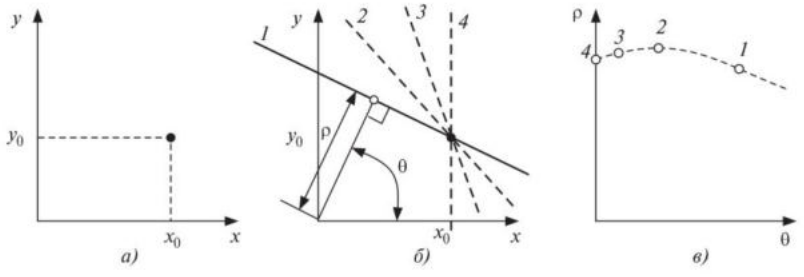

Рис. 7.20. Принцип преобразования Хафа: *а* — координаты точки на плоскости изображения; *б* — преобразование декартовой системы координат в полярную; *в* — характер преобразования исходной точки в кривую в новой системе координат

где R — длина перпендикуляра, опущенного на прямую из начала координат; θ — угол между перпендикуляром к прямой и осью x (рис. 7.20,*а*). Точка (x_0, y_0) на плоскости изображения (рис. 7.20,*а*) соответствует множеству линий, каждую из которых можно выразить через разные ρ и θ (рис. 7.20,*б*).

Каждая из этих линий соответствует точкам на плоскости (ρ, θ), которые, будучи собранными, вместе образуют кривую формы (рис. 7.20,*в*). Анализируя исходное изображение и соответствующее ему фазовое пространство и выбрав ячейку с максимальным значением счетчика попавших кривых, можно выделить соответствующую ей прямую на исходном изображении.

Использование метода при обнаружении сетевых аномалий. В задаче обнаружения аномалий трафика IP-сетей преобразование Хафа имеет прикладное значение. В основе одного из методов обнаружения аномалий лежит вейвлет-преобразование трафика, при котором он раскладывается на масштабные уровни, каждый из которых содержит свой диапазон частот. Анализируя каждый из этих уровней статистическими методами, находятся точки изменения таких параметров, как дисперсия, среднее значение и характеристики самоподобия. Если такие точки изменения найдены на нескольких масштабных уровнях и примерно одинаково локализованы во времени, то можно сделать вывод о наличии аномалии в данной области. Для автоматизированного принятия решения о наличии аномалии в этом случае и используется преобразование Хафа. Для этого найденные точки изменения представляются в виде точек интереса на бинарном изображении, и задача сводится к обнаружению вертикальных прямых по определенному количеству найденных на разных уровнях точек, примерно одинаково локализованных во времени.

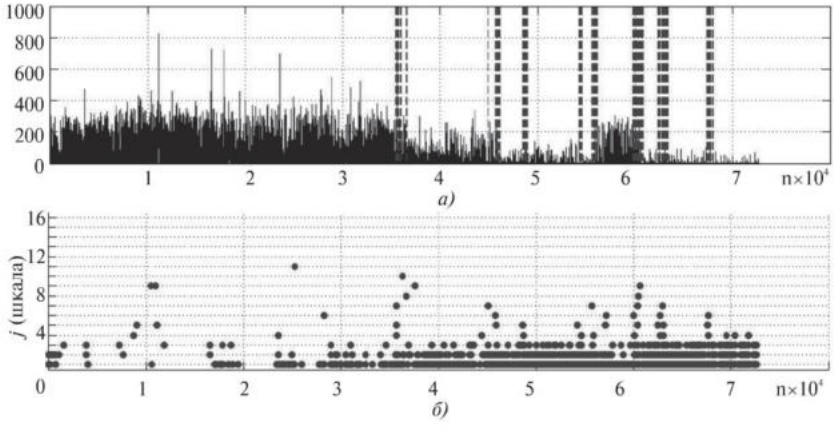

Рис. 7.21. Трасса с аномалиями

На рис. 7.21,*а* представлена трасса, содержащая аномалии. На рис. 7.21,*б* показаны точки изменения, найденные на различных уровнях вейвлет-разложения от 1 до 16. Жирными пунктирными линиями на верхней трассе показан результат работы алгоритма Хафа для нахождения вертикальных линий, проходящих через как минимум 5 точек изменения, найденных примерно в один промежуток времени.

7.7. Достоинства и недостатки методов обнаружения аномалий

У каждого из рассмотренных выше методов обнаружения аномалии есть уникальные достоинства и недостатки. Важно знать, какой метод обнаружения аномалии лучше всего подходит для данной проблемы. Однако, учитывая сложность проблемы, не представляется возможным обеспечить понимание всех методов в обнаружении аномалий. Так, например, методы «ближайшего соседа» и основанные на кластеризации не удобны, когда число размерностей высоко, потому что меры расстояния при высоком числе размерностей не в состоянии различать нормальные и аномальные экземпляры.

Спектральные методы решают проблему высокой размерности, отображая данные в виде проекции с более низкой размерностью. Но их производительность сильно зависит от того, насколько нормальные экземпляры и аномалии различимы в пространстве проекций.

Основанные на классификации методы могут быть лучшим выбором при таком сценарии. Но большинство эффективных основанных на классификации методов требуют меток как для нормальных,

так и для аномальных экземпляров, что не всегда доступно. Даже если метки и для нормальных, и для аномальных экземпляров доступны, неустойчивость в распределении двух меток часто делает изучение классификатора довольно сложным.

В полуконтролируемом режиме методы «ближайшего соседа» и основанные на кластеризации при условии использования нормальных меток, обычно оказываются более эффективными, чем методы основанные на классификации.

Статистические методы, хотя и приемлют бесконтрольный режим, эффективны только в случаях, когда размерность данных низка и имеются статистические предположения.

Теоретическо-информационные методы требуют метрику, которая достаточно чувствительна, чтобы обнаружить эффекты даже единственной аномалии. В противном случае такие методы могут обнаружить аномалии только, когда там имеется значительное число аномалий.

Методы «ближайшего соседа» и основанные на кластеризации требуют вычисления расстояния между парой экземпляров данных. Таким образом, такие методы предполагают, что мера по расстоянию может достаточно хорошо установить различия между аномалиями и нормальными экземплярами данных. В ситуациях, где идентификация хорошей меры по расстоянию трудная, основанные на классификации или статистические методы могли бы быть лучшим выбором.

Вычислительная сложность метода обнаружения аномалии — ключевой аспект, особенно когда метод применен к реальной области применения. В то время как методы классификации, кластеризации и статистические методы обладают большим временем для обучения, тестирование, как правило, дешевое. Часто это приемлемо, так как модели могут обучаться до реального использования, в то время как тестирование требуется в режиме реального времени. Напротив, методы «ближайшего соседа», информационно-теоретические и спектральные методы, у которых нет учебной фазы, имеют дорогая фаза тестирования, которая может быть ограничением в реальном режиме.

Методы обнаружения аномалии обычно предполагают, что аномалии в данных редки по сравнению с нормальными экземплярами. Хотя аномалии не всегда редки. Например, при контакте с обнаружением червя в компьютерных сетях аномальный (червь) трафик фактически более частый, чем нормальный трафик.

Есть несколько перспективных направлений для дальнейших исследований в области обнаружения аномалий. Так, например, начинают находить все большее применение в ряде областей методы обнаружения контекстных и коллективных аномалий. Наличие распределенных данных побудило к развитию и исследованию методов обнаружения распределенных аномалий [57].

В ситуациях, когда данные распределены, когда информация хранится сразу на нескольких сайтах и есть необходимость ее защиты на каждом из них, имеется проблема сохранения конфиденциальность и применения тех или иных методов обнаружения аномалий [58]. С появлением сенсорных сетей, стала необходима обработка данных по мере их поступления в режиме реального времени. Многие методы, рассмотренные в данном исследовании, требуют полного набора данных перед тем, как начать анализ на имеющиеся аномалии. В последнее время были предложены методы, которые могут работать в онлайн моделях [59]. В таких методах предусмотрено не только обнаружение аномалий на конкретных данных, но и последовательное развитие имеющейся модели.

ЛИТЕРАТУРА

1. Лукацкий А.В. Обнаружение атак. — СПб: БХВ-Петербург, 2001. — 624 с.

2. Милославская Н.Г., Толстой А.И. Инстрасети: обнаружение вторжений: Учеб. пособие для вузов. — М: ЮНИТИ-ДАНА, 2001.

3. Девянин П.Н., Михальский О.О., Правиков Д.И., Щербаков А.Ю. Теоретические основы компьютерной безопасности: Учеб. пособие для вузов. — М.: Радио и связь, 2000.

4. Анонимный автор. Максимальная безопасность в Linux.: Пер. с англ. — К.: ДиаСофт, 2000.

5. Сайт системы обнаружения атак IDS «Snort». www.snort.org/

6. Потёмкин А. Общий обзор наиболее часто применяемых техник компьютерных атак и защиты от них // Системный администратор, 2003. № 1 (2).

7. Чирилло Дж. Обнаружение хакерских атак. — СПб: Питер, 2002. — 864 с.

8. Медведовский И.Д., Семьянов П.В., Леонов Д.Г., Лукацкий А.В. Атака из Internet. — М.: Солон-Р, 2002. — 368 с.

9. Фостер Дж., Лю В. Разработка средств безопасности и эксплойтов. Пер. с англ. — СПб.: Питер. Русская редакция, 2007. — 432 с.

10. Гамаюнов Д.Ю. Обнаружение компьютерных атак на основе анализа поведения сетевых объектов. Диссер. МГУ им. Ломоносова, 2007.

11. Eckmann S.T., Vigna G., Kemmerer R.A. STATL: An Attack Language for State-ased Intrusion Detection. — Dept. of Computer Science, University of California, Santa Barbara, 2000.

12. Ozturk A. OSSEC-HIDS Capabilities, Architecture and plans // Presentation at the 5th Linux and Free Software Festival, Ankara, Turkey, 2006.

13. Paxson V. Bro: A system for detecting network intruders in realime // Proceedings of the 7th USENIX Security Symposium, San Antonio, TX, USA, January 1988.

14. Paxson V. Bro: A System for Detecting Network Intruders in RealTime. // Computer Networks. 1999. 31 (23-24). P. 2435–2463.

15. Roesch M. Snort Users Manual, Snort Release: 2.4, 2007. — snort.org.

16. Trusted Computer System Evaluation Criteria, The Orange Book, Department of Defense, NCSC, National Computer Security Centre, DoD 5200.28-STD, December 1985.

17. Balasubramaniyan J., Garcia-Fernandez J.O., Spafford E.H., Zamboni D. An Architecture for Intrusion Detection using Autonomous Agents, Department of Computer Sciences, Purdue University; Coast TR 98-05; 1998.

18. Нестеренко В.А. Статистические методы обнаружения нарушений безопасности в сети // Информационные процессы. 2006. Т. 6, № 3. С. 208–217.

19. Проталинский О.М. Применение методов искусственного интеллекта при автоматизации технологических процессов. — Астрахань: Изд-во АГТУ, 2004.

20. Медведев С.Ю. Преобразование Фурье и классический цифровой спектральный анализ. www.vibration.ru/preobraz_fur.shtml

21. Шелухин О.И. Мультифракталы. Инфокоммуеникационные приложения. — М.: Горячая линия–Телеком, 2011. — 576 с.

22. Muzy J.F., Bacry E., Arneodo A. Wavelets and Multifractal Formalism for Singularity Signals: Application to Turbulence data // Physical Review Letters. 1999. V. 67, 25. P. 3515–3518.

23. Audit B., Bacry E., Muzy J.F., Arneodo A. Wavelet-based estimators of scaling behavior // IEEE Trans. Info. Theory. 2002. № 48. P. 2938–2954.

24. Security Scanner For Network Exploration and Security Audits. — www.insecure.org/nmap/

25. Sheluhin O.I., Atayero A.A., Garmashev A.B. Detection of Teletraffic Anomalies Using Multifractal Analysis // International Journal of Advancements in Computing Technology. 2011. Vol. 3, No 4. P. 174–182.

26. Mallat S. A wavelet tourof signal processing 3 ed. — TheSparseWay, 2005.

27. Шелухин О.И., Гармашев А.В. Обнаружение DoS и DDoS атак методом дискретного вейвлет анализа // Телекоммуникации и транспорт Спецвыпуск по информационной безопасности. 2011. С. 44–47.

28. Шелухин О.И., Гармашев А.В. Обнаружение аномальных выбросов телекоммуникационного трафика методами дискретного вейвлет-анализа // Электромагнитные волны и электронные системы. 2012. № 2. С. 15–26.

29. Sheluhin O.I., Atayero A.A. Integrated Model for Information Communication Systems and Metworks // Design and Development. IGI Global, USA. 2012. 462 p.

30. www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html

31. Vapnik V.N. Statistical learning theory. — Wiley, New York, 1998.

32. Vapnik V.N. The Nature of Statistical Learning Theory. — Springer-Verlag, 1995.

33. Thorsten J. Making Large-Scale SVM Learning Practical // Lehrstuhl VIII, Kunstliche Intelligenz, Dortmund, 1998.

34. Колегов Д.Н. Проблемы синтеза и анализа графов атак. — www.securitylab.ru/contest/299868.php.

35. Котенко И.В., Уланов А.В. Кооперативная работа команд агентов при защите от сетевых атак нарушения доступности. — www.comsec.spb.ru.

36. Люгер Д.Ф. Искусственный интеллект, стратегии и методы решения сложных проблем. 4-е изд. — М.: Вильямс, 2003. — 864 с.

37. Рассел С., Норвиг П. Искусственный интеллект: современный подход. — М.: Вильямс, 2007. — 1408 с.

38. Gorodetski V.I., Kotenko I.V., Karsaev O. Multi-agent technologies for computer network security: Attack simulation, intrusion detection and intrusion detection learning // International Journal of Computer Systems Science & Engineering. 2003. № 4. С. 191–200.

39. Лукацкий А. В. Как работает сканер безопасности? — www.citforum.ru/security/internet/scaner.shtml

40. Дасгупта Д. Искусственные иммунные системы и их применение. Пер. с англ. под ред. А.А. Романюхи. — М.: ФИЗМАТЛИТ, 2006. — 344 с.

41. Комар М.П. Система анализа сетевого трафика для обнаружения компьютерных атак // Вестник Брестского государственного технического университета. Серия «Физика, математика и информатика». 2010. № 5. С. 14–16.

42. Комар М. Методы искусственных нейронных сетей для обнаружения сетевых вторжений // Сборник тезисов седьмой международной научно-технической конференции «Интернет – Образование – Наука» (ИОН-2010) — Винница: Винницкий национальный технический университет, 2010. — С. 410–413.

43. Комар М.П., Боднар Д.И., Саченко А.А. Интеллектуализированная информационная технология обнаружения компьютерных

атак // Измерительная и вычислительная техника в технологических процессах. 2010. № 2. С. 133–137.

44. Головко В.А. Нейрокомпьютеры и их применение. Учеб. пособие. — М., 2001. — 256 с.

45. Kohonen T. Sell-organised formation of topologically correct feature maps // Biological Cybernetics. 1982. № 43. P. 59–69.

46. KDD Cup 1999 Data. — kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

47. Безобразов С.В., Головко В.А. Алгоритмы искусственных иммунных систем и нейронных сетей для обнаружения вредоносных программ // Научная сессия НИЯУ МИФИ «Нейроинформатика». Материалы Всеросс. науч. конф., МИФИ, Москва, 25-29 янв. 2010. С. 273–287.

48. Безобразов С.В., Головко В.А. Применение нейросетевых детекторов в искусственных иммунных системах для обнаружения и классификации компьютерных вирусов // Нейрокомпьютеры. 2010. № 5. С. 17–31.

49. Wang H., Zhang D., Shin K.G. Detecting SYN flooding attacks // Proceedings of IEEE INFOCOM'2002, New York City, NY. 2002. P. 1530–1539.

50. Peng T., Leckie C., Ramamohanarao K. Detecting distributed denial of service attacks using source IP address monitoring // Proceedings of the Third International IFIP-TC6 Networking Conference (Networking 2004). P. 771–782.

51. Gonzalez R.C., Woods R.E., Digital Image Processing. — Prentice Hall, 1993.

52. Owens R. Computer Vision. IT412, Lecture 6, 1997.

53. Крюков Ю.А., Кубарский М.А., Чернягин Д.В. Метод сбора данных о текущих характеристиках в высокоскоростных каналах пакетной передачи данных // Электронный журнал «Системный анализ в науке и образовании». 2009. № 3.

54. Yaroshkin F. SnortNet — A Ditributed Intrusion Detection System // IVT-1/95, Kyrgyz Russian Slavic University, Bishkek, Kyrgystan, June 2000.

55. Barford P., Kline J., Plonka D., Ron A. A signal analysis of network traffic anomalies // Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment (IMW '02), Marseille, France, November 2002. — P. 71–82.

56. Кобзарь А.И. Прикладная математическая статистика. Для инженеров и научных работников. — М.: ФИЗМАТЛИТ, 2006.

57. Zimmermann J., Mohay G. Distributed intrusion detection in clusters based on non-interference // Proceedings of the Australasian Workshops on Grid Computing and E-Research (ACSW Frontiers). Australian Computer Society, Inc. 2006. P. 89–95.

58. Vaidya J., Clifton C. Privacy-preserving outlier detection // Proceedings of the 4th IEEE International Conferrence on Data Mining. 2004. P. 233–240.

59. Pokrajac D., Lazarevic A., Latecki L.J. Incremental local outlier detection for data streams // Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining. 2007.

Оглавление