СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

Р.Р. Фаткиева

Основы построения защищенных компьютерных сетей

Средства и методы интеллектуального обнаружения и предотвращения

СПбГЭТУ «ЛЭТИ», 2021 г.

1 СРЕДСТВА И МЕТОДЫ ИНТЕЛЛЕКТУАЛЬНОГО ОБНАРУЖЕНИЯ И ПРЕДОТВРАЩЕНИЯ

1.1 Архитектура систем обнаружения вторжения. Пассивные и активные системы обнаружения вторжения, системы мониторинга сети (IDS/IPS).

От эффективности функционирования современных компьютерных сетей во многом зависит успешность деятельности практически во всех сферах общества. С каждым годом происходит увеличение пространственно-временной конфигурации сценариев атак, при этом сценарии взаимодействия могут молниеносно переходить в противоборство. Это требует введения систем мониторинга за состоянием компьютерных сетей и прогнозирования их поведения. Однако трудность мониторинга и анализа внутреннего состояния обусловлена огромным многообразием вероятностных состояний. В этих условиях возникает противоречие как в выборе набора показателей для прогнозирования, так и в моделях функционирования компьютерных сетей. Существующие на рынке системы мониторинга и управления не позволяют учитывать динамику изменений стратегий при сетевом взаимодействии, что может привести к снижению устойчивости при отсутствии их согласованности. В качестве наиболее характерных особенностей использования нарушений безопасности можно привести:

-эксплуатацию уязвимостей программного обеспечения, при которых возможно отображение контента легитимного пользователя на стороне нарушителя («rickrolling»), обнаружение пароля устройства в прошивке производителя, создание ботнет -сети с целью проведения вредоносных атак, подключение к Интернету web-камер;

-эксплуатация уязвимостей протоколов передачи стека протокола TCP/IP и протоколов стандарта IEEE 802.11 поскольку технологии, используемые при передаче данных, осуществляются посредством сетей беспроводного доступа, что не только увеличивает коллизии и затрудняет управление за счет задержек передачи данных;

-использование межмашинного взаимодействия позволяет осуществить компрометацию сервера, с которого осуществляется администрирование всеми устройствами с возможностью дальнейшего управления со стороны нарушителя (потенциальное удаленное неавторизованное

переконфигурирование конечных устройств). Поскольку межмашинное взаимодействие осуществляется через канал передачи данных, возникает возможность перехвата или подмены данных (атаки на протокол взаимодействия (MiTM, DoS)). Использование модификации передаваемых пакетов, при которых принимаемое устройство не сможет идентифицировать устройство или выполняемую операцию может превратить производство в хаос. Отдельным видом стоит угроза внедрения вирусов модульного типа;

- диспетчеризация управляющих сообщений от оператора с возможностью изменения исполняемого кода или модификации собственной структуры требует обеспечения дополнительной безопасности.

При этом существуют проблемы обеспечения безопасности информационного обмена:

1. Отсутствие семантической и функциональной интероперабельности.
2. Чрезвычайно сложная техническая и программная инфраструктура автоматизированных производств и информационных систем.
3. Отставание разработки и внедрения средств защиты информации за интенсивным развитием IT-технологий.
4. Отсутствие комплексного подхода при решении задач обеспечения безопасности передаваемых данных из-за разнородности средств защиты, предоставленных разными производителями.
5. Слабая гармонизация нормативно-законодательной базы и отставание от практических задач особенно осуществимо в рамках международного законодательства.
6. Отсутствие единая методика оценки рисков в критических ситуациях.
7. Нелинейность процессов, происходящих в информационно-вычислительных системах и как следствие возникновение уязвимостей.

Поскольку сетевая безопасность является процессом, то необходимо в первую очередь проанализировать данные, полученные путем измерения процесса, который начинается с определения пространства сети и метрик, определить процессы, протекающие в ИВС и на основании процессов осуществить управления в условиях потери устойчивости состояния сетевой инфраструктуры. В этих условиях возникает необходимость разработки систем мониторинга и прогнозирования сетевого трафика.

Мониторинг— система наблюдения за явлениями и процессами, проходящими в окружающей среде, результаты которого служат для обоснования управленческих решений по обеспечению безопасности наблюдаемого объекта. Система мониторинга базируется на съеме и преобразовании информации с датчиков в систему показателей, для последующей обработки и принятия управленческих решений с формированием управляющих воздействий. Для выработки управляющих воздействий анализируется такая информация:

-структурированные данные предсказуемого формата (информация из БД);

-неструктурированные данные высокой степени случайности и вариативности;

-частично структурированные данные с некоторой степенью случайности вариативности.

В подобных условиях возникает необходимость построения систем обнаружения вторжений (СОВ). Особенностью подобных систем является наличие компонентов в виде сенсоров/датчиков, блока мониторинга ситуации, базы происходящих событий, модуля принятия решения. В некоторых системах предусмотрен блок моделирования и прогнозирования ситуации (рис. 7.1). Рассмотрим каждый блок отдельно.

Компонент съема информации с датчиков содержит следующие методы, которые могут быть реализованы как на контроллере датчика, либо в виде программного модуля в информационной системе:

- *предварительная обработка* – отбрасывание незначительных событий, извлечение свойств, сегментация, перевод данных в более подходящий формат;
- *анализ данных* – проверка данных; если они не соответствуют каким-то пограничным значениям, отправляется предупреждение.

Компонент монитринга и хранения (сбора информации о событиях) позволяет агрегировать потоки инфрмации и выявить отклонения от пороговых значений используя методы:

- *окна событий* – создается подвижный диапазон набора данных, который может быть ограничен по времени, длине. Эта функция

хорошо подходит для создания правил и событий, занимающихся подсчетом;

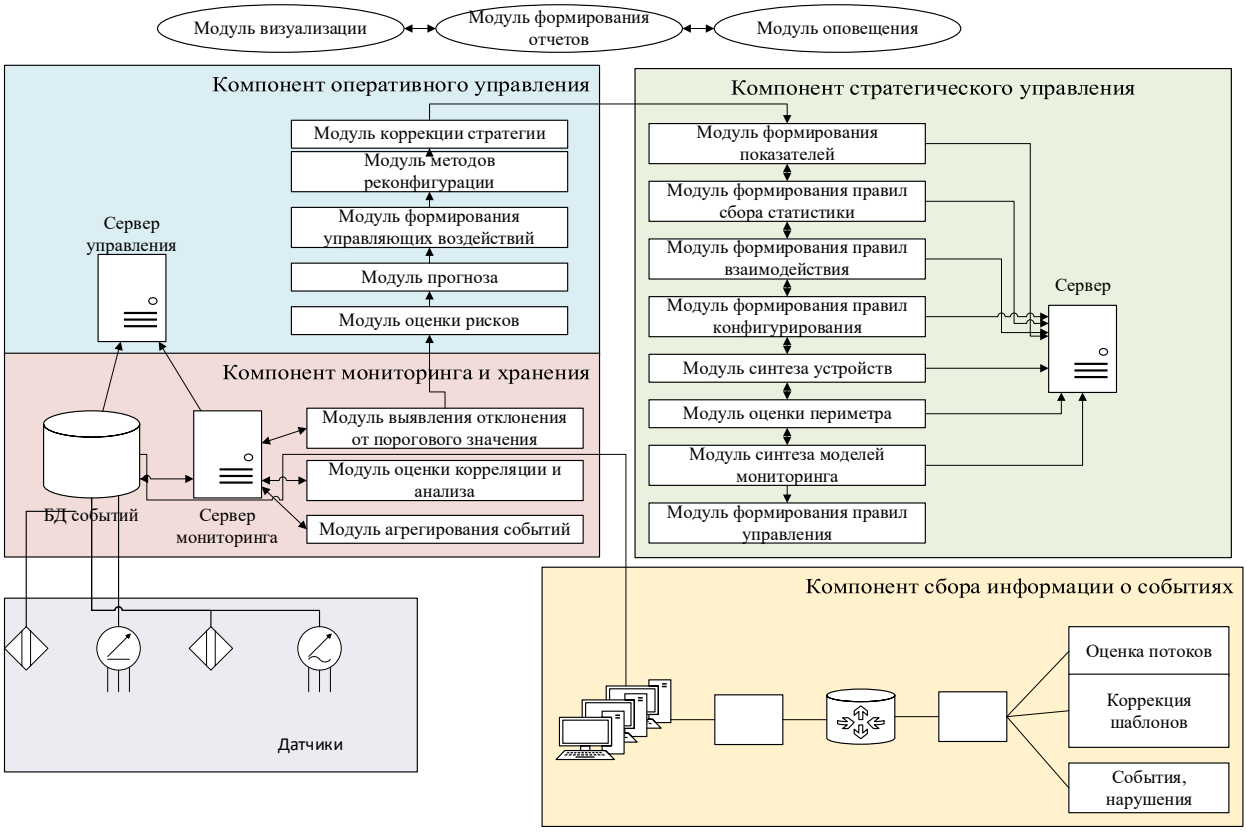


Рис. 7.1.

- *агрегация данных* – объединение несколько потоков данных в один для их дальнейшей обработки;
- *обработка результатов измерений*– оценка полноты и наличие ошибок при потере данных, их искажении или неправильным порядком следования, оценка корреляции, выявление отклонений от пороговых значений;
- *коррекция шаблонов*– позволят скорректировать набор показателей, необходимым для отслеживания тех или иных состояний наблюдаемого объекта.

Компоненты оперативного управления (консоль управления) позволяют формировать наборы управляющие воздействия и содержать методы:

-оценки рисков- необязательный компонент, позволяющий получить определение величины ущерба от возникшей рисковой ситуации и несвоевременного принятия мер по предотвращению риска.

-прогноза - необязательный компонент, позволяющий спрогнозировать набор показателей в заданном окне событий;

-управления -позволяют сформировать комплекс управляющих воздействий и/или мероприятий для достижения целевой функции;

-реконфигурации- изменение состава и способа взаимодействия программных и аппаратных средств объекта с целью исключения отказавших программных или аппаратных компонентов или протекающих процессов;

-коррекции стратегии -исправление операционного процесса достижения целевой функции.

Компоненты стратегического управления позволяют создавать наборы управляющих воздействий для достижения целевой функции и содержат методы формирования:

- *показателей*- наборов данных, характеризующих количественные, качественные или интегральные свойства объекта;
- *правил взаимодействия*-правил, по которым осуществляется совместный доступ к ресурсу;
- *правил конфигурирования* и синтеза устройств- правил соединения элементарных единиц, операционных процессов;
- *оценки периметра* и моделей мониторинга- нахождения границ, в пределах которых осуществляется мониторинг за объектами управления.
- *правил управления*- набора связей ответных реакций на входное воздействия, отраженное показателями, считанными с датчика.

По типам системы обнаружения вторжений подразделяются на:

- Сетевая СОВ используя доступ к сетевому трафику, подключаясь к шлюзу, настроенному на зеркалирование портов.
- Основанная на протоколе СОВ, представляет собой систему (либо агента), которая отслеживает и анализирует коммуникационные протоколы со связанными системами или пользователями.
- Узловая СОВ (Host-based IDS, HIDS) — система (или агент), расположенная на хосте, отслеживающая вторжения, используя анализ системных вызовов, логов приложений, модификаций файлов.

- Гибридная СОВ совмещает два и более подходов к разработке СОВ.

Отдельным пунктом, который не попадает в классификацию, являются системы honeypot.

1.2 Многоагентные системы обнаружения вторжений.

Для реализации СОВ на наблюдаемых устройствах с возможностью управления на отдельном сервере в частном случае, используют многоагентые технологии. В этом случае архитектура состоит из набора агентов различных типов, специализированных для решения подзадач обнаружения отклонений от заданных параметров функционирования сети, расположенных на сетевых хостах, с возможностью сбора и анализа статистики. Такую систему можно представить в виде графа G = (A,B), где A – множество агентов, B – множество ребер графа; ребро существует тогда и только тогда, когда пара агентов может связаться друг с другом, минуя других агентов.

В общем случае многоагентные системы включают в себя (Рис. 7.2):

- агентов сети, анализирующих сведения о пакетах, передаваемых по сети, которые могут размещаться на сетевых устройствах (например, анализирующих сведения о событиях маршрутизаторов)
- агентов серверов, анализирующих сведения о событиях, происходящих на серверах ИС
- агентов рабочих станций, анализирующих сведения о событиях, происходящих на рабочих станциях.

При применении указанной архитектуры необходимо учитывать, что зачастую возникают трудности при определении типа отклонений и степени ее опасности. Отклонения могут иметь различные причины и быть связаны как с деятельностью наблюдаемого объекта, так и с неисправностью аппаратуры и дефектами программного обеспечения. Отклонения могут быть видимыми и проявляться непосредственно в некорректной работе датчика, информационно-вычислительной системы, а могут не иметь видимых признаков, но привести к сбоям через длительное время.

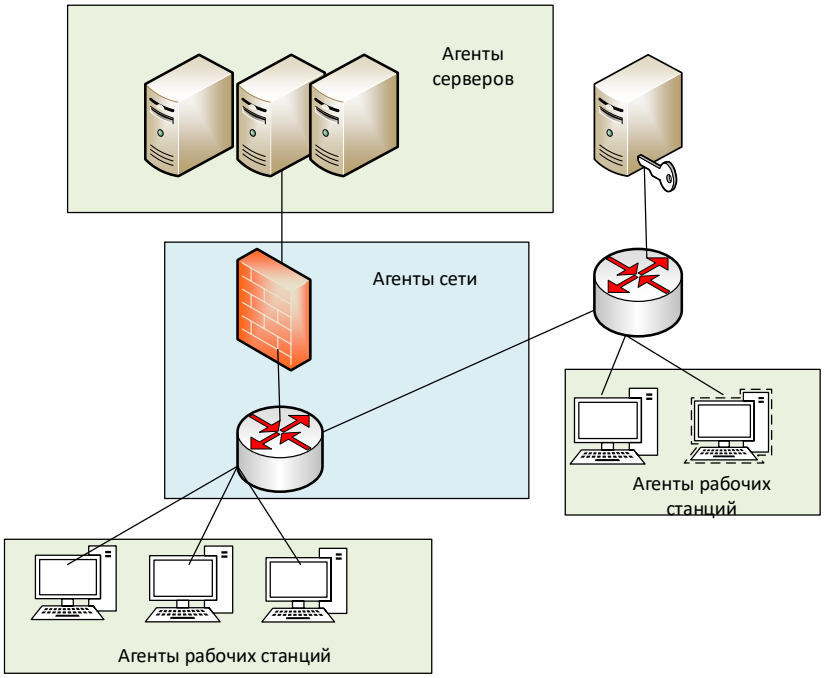


Рис. 7.2.

Существующие СОВ как правило не предусматривают контроль контента, содержащегося в сообщениях. Исторически сложилось, что методы обнаружения вторжений можно разделить на основные классы, которые могут пересекаться (Рис. 7.3):

1. *Статистические методы*, используемые, в том числе при машинном обучении предполагают количественный анализ трафика. Системы, реализующие этот метод, производят мониторинг поведения и контролируют значения характеристических величин сетевого трафика. Характеристики: низкая достоверность, не всегда высокая производительность, но существует способность к обнаружению новых атак.

2. *Сигнатурные методы* используют как качественный, так и количественный анализ трафика и позволяют обнаружить атаку по характерным для неё параметрам. Характеристики: обнаружение только известных атак, но с более высокой точностью.

Дальнейшие формализация, моделирование и исследование противоборства злоумышленников и систем защиты в сети Интернет привело к увеличению количества методов обнаружения, к которым относятся:

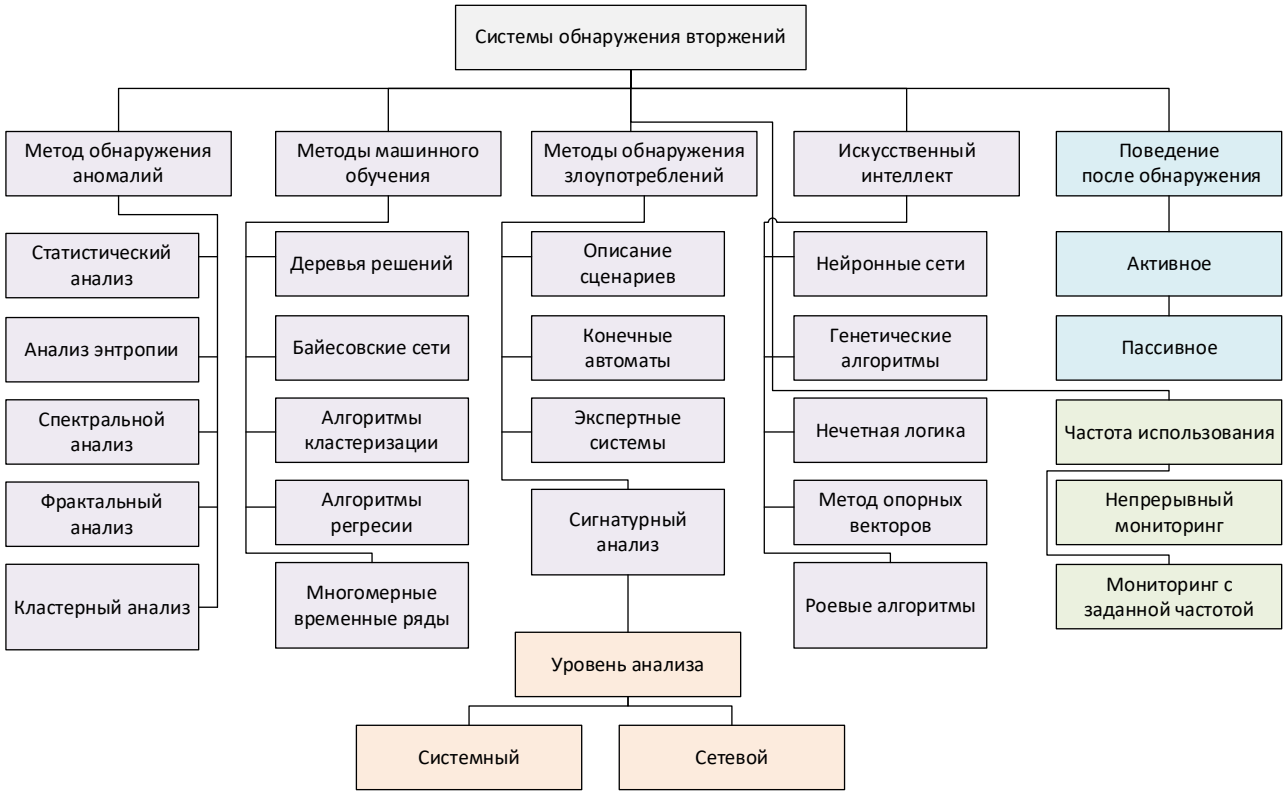


Рис. 7.3

3. Метод обнаружения аномалий, основанный на анализе поведения пользователей и мониторинге отклонения текущего поведения от нормального профиля, при котором превышение пороговой величины расценивается как аномалия. Метод позволяет обнаруживать известные и неизвестные ранее аномалии сетевого трафика, но обладает низкой достоверностью принимаемых решений. Это связано с нестационарностью процессов, протекающих в сети, и совершенствованием несанкционированных воздействий.

4. Иерархические модели Маркова позволяют вычислить неизвестные характеристики моделей.

5. Обнаружение вторжений, основанное на использовании многоагентного моделирования механизмов защиты от атак, базируется на моделях команд агентов и их взаимодействия, особенностью которых является учет ключевых параметров исследуемых процессов – параметров сети и ее узлов, параметров команды атаки и реализации атаки, параметров команды защиты и механизмов защиты, параметров взаимодействия команд, что позволяет исследовать эффективность разнообразных механизмов защиты.

6. Применение нейросетевых методов показывает возможность повышения качества обнаружения сетевых атак в компьютерных системах, а также повышения быстродействия системы за счет сокращения анализируемых данных. Однако некоторые типы атак, такие как ipsweep, multihop, warezclient, perl и rootkit, недостаточно хорошо обнаруживаются. Поскольку нейросетевой подход достаточно терпим к приемлемому количеству ошибочных обучающих примеров, то имеет и слабые стороны: в частности, относительность выдаваемых ответов, высокая вычислительная стоимость обучения, что приводит к отсутствию гарантий приемлемости результата применения метода.

7. Методы, основанные на анализе поведения компонент ПО и ОС, осуществляют контроль поведения на уровне исполняемого кода и на уровне доступа к ресурсам. Изменение поведения процесса на уровне исполняемого кода приводит к выполнению инструкций, не предусмотренных логикой решаемой задачи, что влечет за собой изменение характера обращения к ресурсам вычислительной системы и может быть зафиксировано путем анализа процессов, протекающих в интерфейсах системы.

Все перечисленные методы используются как в пассивных СОВ (IDS), в которых информация о нарушении записывается в лог приложения, а сигналы опасности отправляются администратору системы по определенному каналу связи и в активных системах (IPS), в которых предусмотрены ответные действия на нарушение, сбрасывая соединение или перенастраивая межсетевой экран для блокирования трафика от злоумышленника (Рис. 7.4).

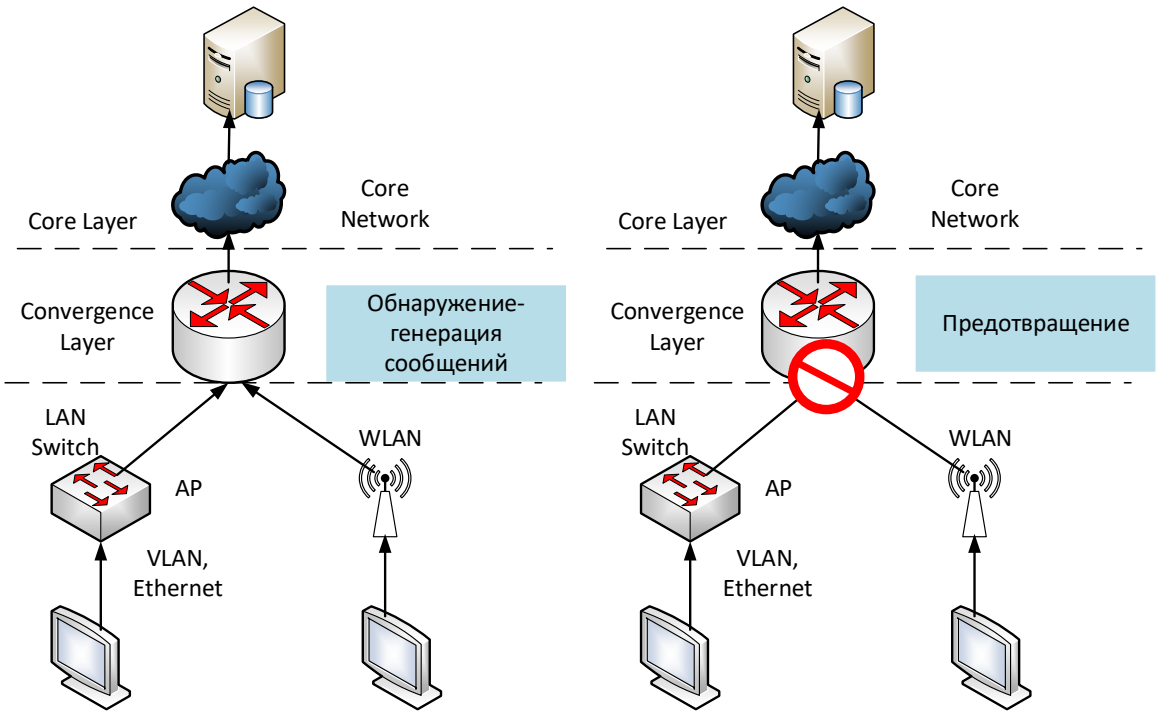


Рис. 7.3

Одним из подходов к разработке систем обнаружения нарушения безопасности является статистический анализ и построение прогноза. Из методов статистического анализа наиболее часто применяются модели сглаживания, автокорреляции, спектрального анализа, авторегрессии и др (Рис. 7.4).

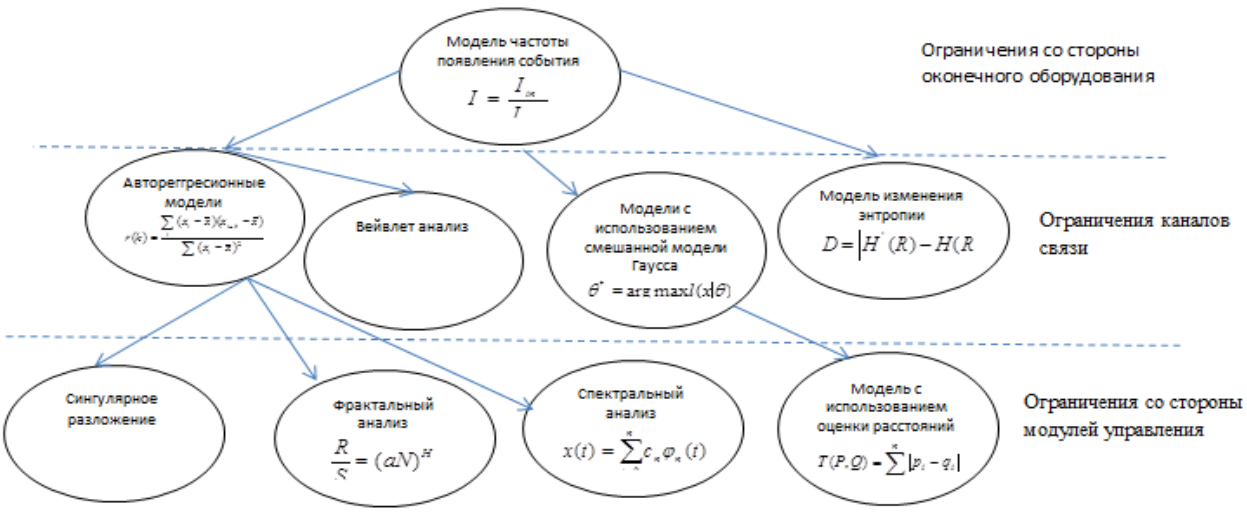


Рис. 7.4.

Построение процедуры сглаживания на основе метода скользящего среднего позволяет оценить тренд и построить упрощенную модель прогноза. Однако необходимо помнить, что данная модель линейна и не может

использоваться для критических условий. При подборе модели анализа статистическими методами возникает необходимость построения набора параметров для обнаружения идентификации возникновения аномалии. Это в свою очередь приводит к необходимости сопоставления тех или иных параметров и формирования требований к метрическим показателям, в частности:

- воспроизводимости;
- отсутствии субъективной оценки;
- простоте в сборе и снятии показателей с точек доступа;
- возможности выразить измеряемую характеристику числом;
- специфичностью и понятностью для администратора сети.

Проблема построения параметров функционирования сети напрямую связана с необходимостью описания классификации информационных воздействий, так как идентификация атаки требует поиска тех характеристик информационного потока, которые однозначно смогли бы обнаружить с заданной точностью тип воздействия или совокупность типов в случае множественности воздействий.

1.3 Методы машинного обучения для оценки сетевого трафика

Машинное обучение (machine learning, ML)— класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение за счёт применения решений множества сходных задач. Для построения таких методов используются средства математической статистики, численных методов, математического анализа, методов оптимизации, и др. Применение статистических методов для оценки сетевого трафика позволяет сформировать математическую модель следующим образом:

$$M = (T, X_r, Md, P),$$

где *T* – постоянный временной интервал, X_r – характеристики трафика, *P* – пороговое значение, *Md* – метод расчета.

Если задать за T некий постоянный временной интервал, то анализируемая статистика может быть как текущей (частной) при малых значениях T, так и долговременной (глобальной) при значении T от нескольких часов или суток. Кроме временной составляющей присутствует и

сам объект анализа. Его образует информация о сетевых взаимодействиях: источник; получатель; тип пакета; порт обращения; длина пакета; тип обслуживания; время жизни пакета; протокол передачи; размер окна; наличие соответствующих флагов и др. Это позволяет сформировать методы обнаружения сетевых атак, основанные на анализе статистических характеристик сетевого трафика, отражающих параметры штатного поведения системы. При регистрации отклонения характеристики от заданного значения фиксируется факт обнаружения атаки (Рис. 7.5). Примерами подобных моделей являются пороговая модель, модель среднего значения и среднеквадратичного отклонения и др.

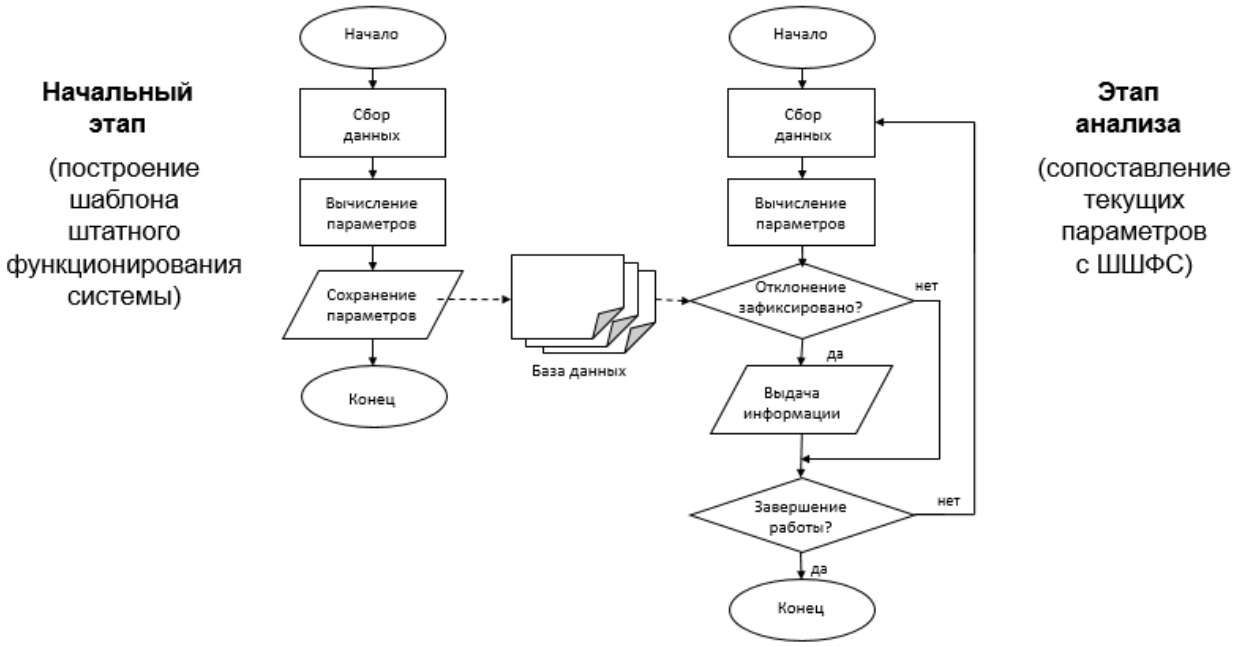


Рис. 7.5

Такой подход позволяет связать поведение параметров с типом атак, которым подвержена система. Пороги отклонений от штатного функционирования рассчитываются различными методами, в том числе основанными на:

- использовании оценки изменения распределений [1];
- основе вычисления разности энтропии [2];
- расчете появления частоты событий [3, 4];
- использовании спектрального анализа [5];
- использовании сингулярного спектрального анализа [6,7];
- использовании фрактального анализа [8].

Преимущество данных методов заключается в учете нелинейных факторов, присутствующих в сетевом трафике. Ряд методов достаточно удобен в реализации, однако не всегда отражает реальную динамику процесса, за счет присутствия нелинейной динамики и стремительности процесса. Ряд методов труднореализуем за счет сложности методов и реализации алгоритмов как программным, так и аппаратным способом. Традиционная модель сетевого взаимодействия, представлена на Рис. 7.6. Применение указанных методов на элементах сети приводит к избыточности средств защиты информации в одном случае, и не достаточности в другом. Все вышесказанное приводит к необходимости гармонизации средств защиты на элементах сетевого взаимодействия, что позволит уменьшить избыточность средств защиты, исключить наличие конфликтов и уменьшит нагрузку на сеть.

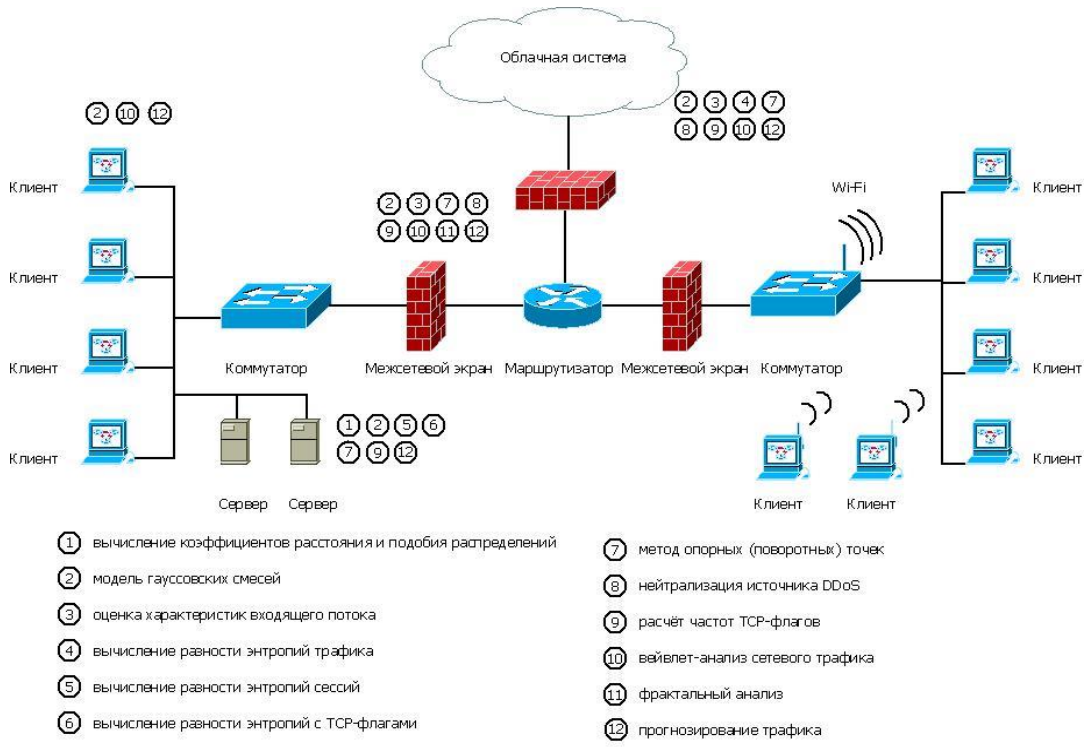


Рис. 7.6

Использование метода гауссовских смеси. Использование методов гауссовских смесей основан на том, что подверженная DDoS-атаке система теряет способность отвечать на запросы. Таким образом, в системе-жертве при наличии атаки входящий трафик за некоторый промежуток времени Δt возрастает, а исходящий трафик – снижается или возрастает в меньшей

степени. С учётом этого авторы принимают в качестве статистической характеристики величину:

$$IP_{traffic} = \frac{IPtraffic_{in}}{IPtraffic_{out}}$$

где $IP_{traffic_{in}}$ объем, входящего трафика, $IP_{traffic_{out}}$ объем исходящего трафика.

Увеличение характеристики $IP_{traffic}$ свидетельствует о том, что система подвержена DDoS-атаке. Рисунки 7.7 (Динамика изменений значения $IP_{traffic}$ без атаки), 7.8 (Динамика изменений значения $IP_{traffic}$ с атакой) демонстрируют динамику $IP_{traffic}$ в штатном режиме и в режиме вторжения.

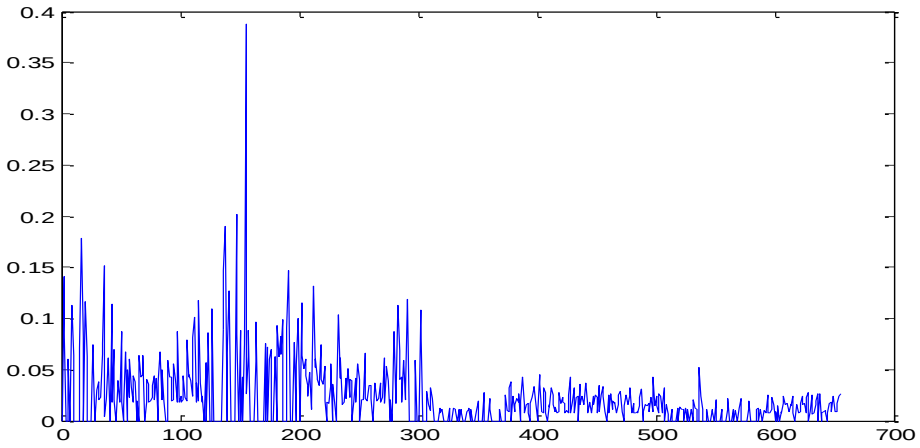


Рис. 7.7

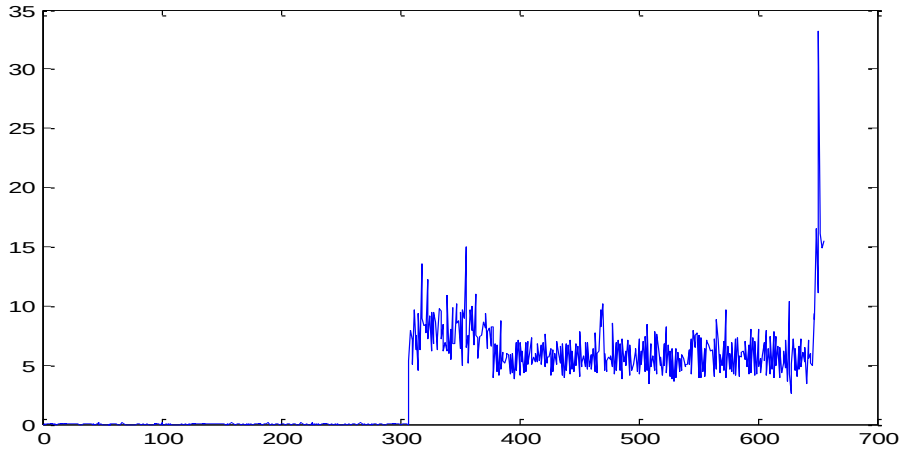


Рис. 7.8

Использование данного подхода в сети показали простоту реализации и возможность адаптации при необходимости изменения пороговых значений

детектирования атаки. К недостаткам можно отнести трудность идентификации класса атак.

Определение отклонений с использованием смешанной модели Гаусса. Для этой используют смешанную модель Гаусса и ЕМ-алгоритм. Случайная величина X имеет смешанное нормальное распределение, если функция плотности распределения:

$$p(x) = \sum_{i=1}^{k} \alpha_i f_i(X; \mu_i ; \sigma_i), \quad (2)$$

где f_i- плотность нормального распределения, α_i - весовые коэффициенты.

Для оценки параметров этой модели (множество$\theta = \{\alpha_i, \mu_i, \tau_i\} i = 1, \ldots, k$) применяется ЕМ-алгоритм, максимизирующий функцию правдоподобия: $l(x|\theta) = p(x_1, x_2, \ldots, x_N|\theta)$, т. е. вероятность появления набора данных при множестве параметров θ. Решение имеет вид:

$$\theta^* = argmax\, l\,(x|\theta) \quad (3)$$

Алгоритм представляет собой последовательность действий.

1. Начальная инициализация параметров θ^0 выполняется произвольно.

2. Вычисляются условные вероятности $p(i|x_n)$для каждого компонента смеси $i \quad i(i = 1, \ldots, k)$ и каждого образца данных $x_n (n = 1, \ldots, N)$

$$p(i|x_n) = \frac{\alpha_{iN(x_n, \mu_i, \sigma_i)}}{\sum_{i=1}^{k} \alpha_{iN(x_n, \mu_i, \sigma_i)}} \quad (4)$$

3. М-шаг. Используя эти вероятности, оцениваются новые, более точные, параметры смеси:

$$\alpha_i^{'} = \frac{1}{N} \sum_{n=1}^{N} p(i|x_n), \quad (5)$$

$$\mu_i^{'} = \sum_{n=1}^{N} \frac{p(i|x_n)}{\sum_{n=1}^{N} p(i|x_n)} x_n, \quad (6)$$

$$\sigma_i^{'} = \sum_{n=1}^{N} \frac{p(i|x_n)}{\sum_{n=1}^{N} p(i|x_n)} (x_n - \mu_i^{'})^2 \quad (7)$$

4. Переход к шагу 2 до достижения сходимости.

Выявление отклонений выполняется с помощью вычисленных в алгоритме условных вероятностей $p(i|x_n)$ Они отражают степень согласования между образцом данных x_n и i -м компонентом гауссовой смеси. Элементы данных можно распределить по компонентам смеси:

Таблица 1

Образцы данных	Компоненты

	N_1	N_2	N_3	N_4
Набор данных 1	1	0	0	0
Набор данных 2	0	1	0	0
Набор данных 3	p_1	p_2	p_3	p_4
Набор данных 3	0	0	0	0

Некоторые наборы данных не согласуются ни с одним компонентом смеси (т. е. их условные вероятности для всех *i* близки к *0*). В данном случае это элемент $data_5$, он является подозрительным.

К достоинствам метода можно отнести простоту реализации и низкую вычислительную сложность, которая может увеличиваться с ростом трафика. К недостаткам, также можно отнести трудность адаптации алгоритма при изменении коэффициентов процессе эксплуатации.

Оценка появления частоты событий. К часто используемым статистически характеристикам, применяемым для оценки частоты появления события в сетевом трафике можно отнести:

1. количество подключений и их статусы (например, установленное, закрыт-ожидание, время-ожидание);
2. усредненное количество посланных и полученных пакетов;
3. продолжительность соединения;
4. тип соединения;
5. используемый порт и протокол.

Метод, основанный на анализе частоты протоколов и флагов

Предлагаемый метод использует для выявления факта атаки характеристики, представляющие собой частоты появления протоколов транспортного уровня и определённых флагов в заголовках TCP-пакетов:

$$R_t[Pi] = \frac{\text{число пакетов с протоколом } P}{\text{общее число пакетов}} \text{ (входящих)},$$

$$R_t[Po] = \frac{\text{число пакетов с протоколом } P}{\text{общее число пакетов}} \text{ (исходящих)},$$

$$R_t[Fi] = \frac{\text{число флагов типа } F \text{ в } TCP-\text{пакетах}}{\text{общее число } TCP-\text{пакетов}} \text{ (входящих)},$$

$$R_t[Fo] = \frac{\text{число флагов типа } F \text{ в } TCP-\text{пакетах}}{\text{общее число } TCP-\text{пакетов}} \text{ (исходящих)}$$

здесь t – период сбора данных, F представляет собой один из 6 флагов *SYN*, *FIN*, *ACK*, *PSH*, *RST* и *URG*, обозначаемых своей первой буквой. P обозначает тип протокола (*TCP, UDP, ICMP*).

Для определения правил принятия решений авторы используют алгоритмы машинного обучения, такие как C4.5, CN2 и Байесовский классификатор. Алгоритм C4.5 показывает, что при $R_t\{S_i\} > 0{,}4$ высока вероятность SYN-flood.

К достоинствам можно отнести простоту реализации. К недостаткам можно отнести то, что описан только способ обнаружения SYN-flood, для обнаружения остальных атак потребуются дополнительные исследования.

Метод с использованием отклонения от порогов. На основании использования характеристик из базового перечня формируется оценочная шкала для фиксирования события T_r, представляющего собой прибытие пакета. Формируется ряд X, состоящий из длин полученных пакетов, и ряд Y, представляющий собой шаблон штатного функционирования сети. Пороговая методика применяется в два этапа. На первом выбирается коэффициент чувствительности k и определяются нижний и верхний пороги значений величины X:

$$(1-k)Y_i < X_i < (1+k)Y_i$$

На втором этапе необходимо определить краевые значения допустимых интервалов. Для этого определяется выборочное среднее множества:

$$Average(X) = \frac{1}{n}\sum_{i=1}^{n} X_n$$

и строится допустимый диапазон для X:

$$\frac{1}{2}Average(X) < X_i < \frac{2}{3}Average(X)$$

Многократное выявление отклонений от заданного интервала определённо свидетельствует об изменении в процессе функционирования сети, что может быть как легальным изменением нагрузки на сеть, так и следствием несанкционированными действиями.

Другим показателем, применяемым в пороговой методике, является среднеквадратическое отклонение потока TCP/IP как от среднего значения

(7.10), так и от данных, заложенных в шаблон штатного функционирования сети (Рис. 7.9):

$$\sigma_1 = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(X_i - Average(X))^2}.$$

$$\sigma_2 = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(X_i - Y_i)^2}.$$

Стоит отметить, что эффективность применения данного метода зависит от временного периода анализа.

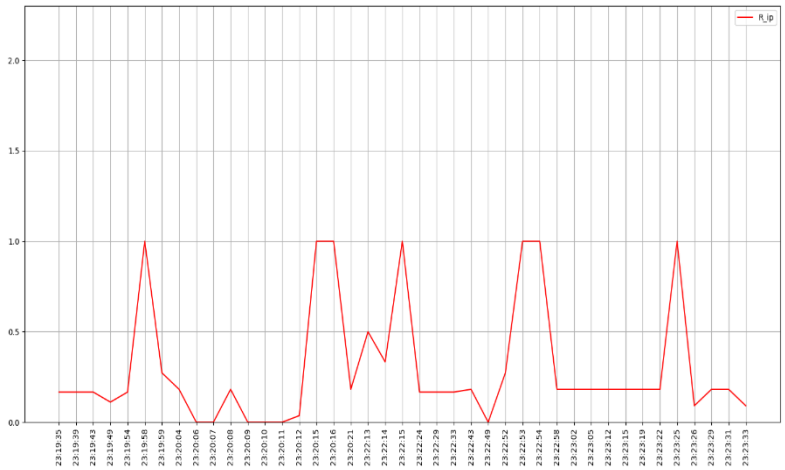


Рис. 7.9

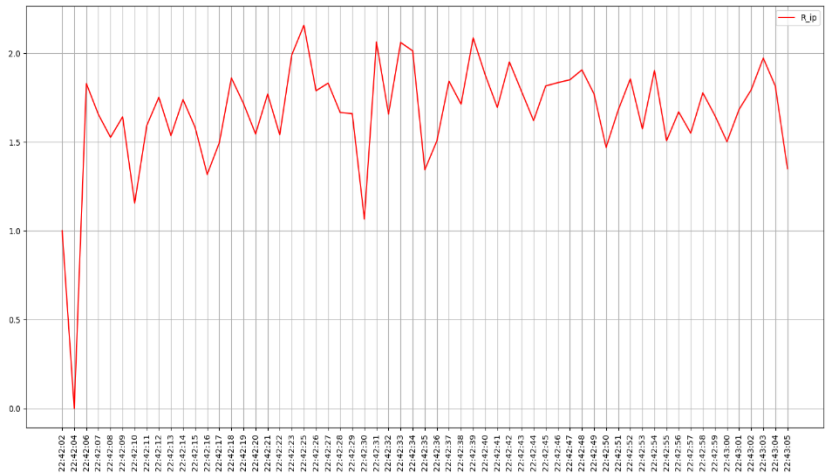


Рис. 7.10

Основываясь на состояние сетевого потока, временной интервал варьируется в пределах от 1 до 30 минут на итерацию анализа. Немаловажным также является уменьшение вероятности появления ошибок первого рода,

характерных для статистических методов, в совокупности с которыми рекомендуется применять компоненты независимых методов анализа.

Метод спектрального анализа сетевого трафика. В характеристиках сетевого трафика существует наличие «колебательного» поведения. Это связано как с особенностями протоколов передачи данных, работы сетевого оборудования, так и активностью пользователей.

Мощным инструментом обработки данных, определенных дискретной зависимостью $y(x_i)$ или непрерывной функцией $f(x)$ (полученной, например посредством интерполяции), является спектральный анализ, имеющий в своей основе различные интегральные преобразования. Спектральный анализ используется как в целях подавления шума, так и для решения других проблем обработки данных, в том числе прогнозирования перегрузок и обнаружения сетевых атак.

Случайную реализацию $x(t)$ разложим по детерминированным ортогональным функциям:

$$x(t) = \sum_{i=0}^{n} c_n \phi_n(t),$$

где коэффициенты разложения c_n - случайные величины.

Определим коэффициенты гармонического разложения из формулы:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \overline{X}(\varpi) e^{-2\pi i} \, d\varpi,$$

и

$$\overline{X}(t) = \int_{-\infty}^{\infty} x(t) e^{-2\pi i} \, dt.$$

Для случайного процесса $x(t)$ спектральная плотность мощности, связана с автокорреляционной функцией преобразованием Фурье (соотношение Винера - Хинчина):

$$K_k(r) = \frac{1}{2\pi} \int_{-\infty}^{\infty} W_k(\varpi) e^{-2\pi i} \, d\varpi,$$

$$W_k(\varpi) = \int_{-\infty}^{\infty} K_k(r) e^{-2\pi i} \, dr.$$

Определим из последнего соотношения по функции корреляции спектральную плотность мощности для эргодического процесса:

$$K_k(r) = \lim_{T \to \infty} \frac{1}{T} \int_{\frac{-r}{2}}^{\frac{r}{2}} x(t) x(t+r) dt$$

Получим формулу при нулевом среднем значении:

$$\overline{x^2}(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} W_k(\varpi)d\varpi,$$

При уменьшении энергетического спектра случайного процесса, уменьшается изменения $x(t)$, а время корреляции увеличивается.

Как было сказано выше, применение спектрального анализа затруднительно в программно-аппаратных комплексах, поскольку метод имеет большую вычислительную сложность, а также избыточность полученных результатов для анализа. Это приводит к трудностям адаптации метода при применении в условиях использования многопользовательской сети.

Модель описания сетевого трафика на фрактальном анализе. Сетевой трафик обладает свойством самоподобности вследствие:

1) объединения множества отдельных изменчивых ON/OFF источников;

2) синдрома бесконечной дисперсии;

3) клиент-серверной архитектуры.

Одним из свойств, приводящих к самоподобности сетевого трафика, является ограниченность ресурсов, накладываемых клиент-серверной архитектурой. Процессы, порождаемые пользователями в такой среде, в борьбе за ресурс вычислительной системе провоцируют нелинейное поведение, которое приводит к возникновению долговременной зависимости.

Генерация трафика может быть вызвана деятельностью пользователя по передаче и поиску информации, при этом распределение пользовательских запросов обладает свойствами флуктуаций во временном диапазоне и цикличностью. Имеет место корреляция между распределениями времени поступлением пакета и размера пакета. Одной из причин возникновения долговременной зависимости является также свойство аппаратной реализации средств передачи данных, позволяющее объединять пакеты из разных источников. Это проявляется в неустойчивости при операциях:

- объединения;
- построения очереди;
- управления трафиком;
-изменения емкости буфера.

При наложении однородных и разнородных источников, обладающих разными корреляционными характеристиками, объединенный трафик сохраняет долговременную зависимость, независимо от источника и интенсивности трафика. Анализ чередования участков с различной фрактальной размерностью и того, как на систему воздействуют внешние и внутренние факторы, позволяет предсказывать поведение системы. Самоподобные модели могут проявлять свойство долговременной зависимости, что означает проявление зависимости между событиями через достаточно большие промежутки времени. Для количественной оценки наличия сетевой аномальности и возможности статистического анализа, для последующего реагирования на инцидент используют такие параметрами как: показатель Херста, R/S статистика, оценка Виттла, вейвлет-анализ, анализ индекса дисперсии и др. Оценки Херст- параметра H основываются на идее измерения наклона линейного приближения на графике log- log, если имеется дискретная реализация наблюдений $x_1, x_2, \ldots x_n$ в соответствующие моменты времени$t_1, t_2, \ldots t_N$, где N - объем выборки отсчетов.

Обозначим через $g_j(t)$ накопленное отклонение процесса $x_i(t)$ от среднего $\bar{x}$ к моменту времени t_j:

$$g_j = \sum_{i=1}^{j}(x_i - \bar{x}).$$

Размах накопленного отклонения R определяется как разность между максимальными и минимальными накопленными отклонениями $g_j(t)$ по формуле:

$$R = \max_{1 \le j \le N}\{g_j\} - \min_{1 \le j \le N}\{g_j\}.$$

Тогда параметр Херста определяется из соотношения:

$$\frac{R}{S} = (aN)^H,$$

где R -размах отклонения, S -стандартное отклонение, N -число членов временного ряда, α -константа.

Диапазон изменений параметра Херста для сетевого трафика находится в интервале (0.5, 1). На качественном уровне такой самоподобный трафик имеет постоянный «взрывной» характер и обладает высокой пачечностью на многих масштабах временной оси.

Самоподобие можно расценивать как фундаментальное статистическое свойство сетевого трафика, которое необходимо учитывать на практике.

Проведенный анализ трафика при атаке и без нее, подтвердил результаты работ и показал присутсвие эффекта самоподобия, при котором возникает уменьшение автокорреляции, свидетельствующее о нарушении стационарности процесса обработки информации. Однако при происходящем увеличении дисперсии характеристики интенсивности сетевого трафика фрактальный анализ поведения показал увеличение показателя Херста (Таблица 2). Возможно, это связано с увеличением количества «однотипных пакетов» в информационной среде. При этом оценка показателя Херста показала, что для каждого вида атак показатель Херста меняется в зависимости от интенсивности атаки и от вида атаки.

Таблица 2

Вид трафика	Среднеквадратичное отклонение	m	N	Dmax	Показатель Херста
Нормальный	0,314	25,71	250	4.79	0,53
SYN-flood	0,531	164,06	250	4.79	0,95
TCP-flood	0,738	200,02	250	4.795	0,79

Следует заметить, что наличие больших всплесков на фоне относительно низкого среднего уровня трафика при функционировании в штатном режиме можно принять за атаку. Во избежание ошибок второго рода (ложное срабатывание систем защиты) при формировании реакции на атаку необходимо учитывать в том числе самоподобие сетевого трафика.

Методы, основанные на автокорреляционной функции. В общем случае автокорреляционная функция (АФК) характеризует внутреннюю зависимость между временным рядом и тем же рядом, но сдвинутым на некоторый промежуток (сдвиг) времени. Вычисления АКФ проводятся по формуле:

$$r(k) = \frac{\sum_i (x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_i (x_i - \bar{x})^2},$$

где $\bar{x}$ - выборочное среднее, $k = 1,2,\ldots$ - шаг сдвиг, x_i-*i*- элемент ряда

Понятие медленно убывающей зависимости (автоковариации) имеет ключевое значение в теории самоподобных процессов и фактически описывает свойство трафика - продолжительную память. В процессе обнаружения аномального поведения функционирования сети на основе среднеквадратичного отклонения анализ экспериментальных данных показал возможность выявления несанкционированной активности также и по отклонениям от автокорреляционной функции, представленным на Рис. 7.11 (Расчет автокорреляционной функции в штатном режиме функционирования), 7.12 (расчет автокорреляционной функции трафика с атакой SYN flood).

Анализ автокорреляционной функции позволяет не только определить наличие несанкционированной активности, но и сделать предположение о виде атаки.

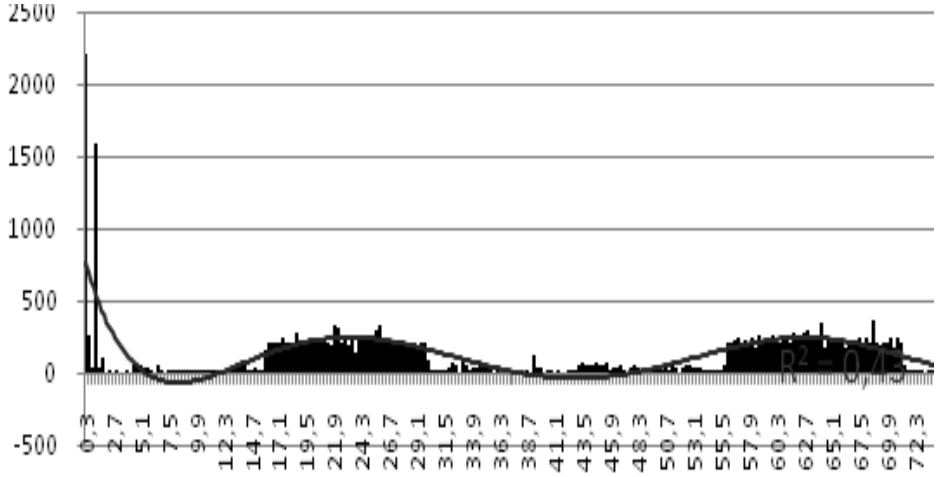


Рис. 7.11

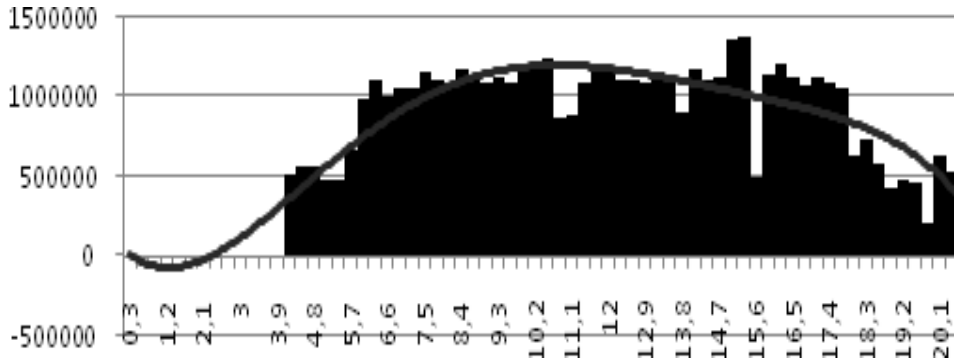


Рис.7.12

Модель, основанная на вейвлет преобразовании входных и выходных потоков. Идея многомасштабного представления сигналов, с вычислительной точки зрения реализации процедур прямого и

обратного вейвлет-преобразования, базируется на каскадном, или пирамидальном, способе представления сигналов и вытекающих из него методах вейвлет-фильтрации. Использование подходов каскадного преобразования сигналов позволяет получать как приближение исходных данных, имеющих локальные особенности с необходимой точностью, так и последовательное огрубление первоначальной информации при решении задач оперативного обнаружения сетевых аномалий.

В качестве критерия выбора оптимального базиса вейвлет-пакетного преобразования возможно использование критерия минимума энтропии, характеризующей уровень усреднения и определяющей количество существенных коэффициентов модели трафика. Дополнительными ограничениями являются ресурсные затраты. Использование вейвлет-пакетного преобразования (ВПП) позволяет повысить адекватность описания модели сетевого трафика за счет устранения влияния флуктуаций. Однако предложенная модель не обеспечивает управление базой правил, заложенных в фильтрующие устройства, в режиме реального времени в силу необходимости пересчета нормального состояния трафика сети $f_{d^э}(t)$ на каждом интервале мониторинга сети.

В традиционных моделях ВПП трафик рассматривается как одномерный массив данных в виде числового ряда$f(t_i)$, разложенного по функциям тренда q_t , циклических компонент q_s , локальных аномалий ε_a , и флюктуаций значений более высокого порядка ε_f (шумов). Т. е. модель трафика имеет вид:

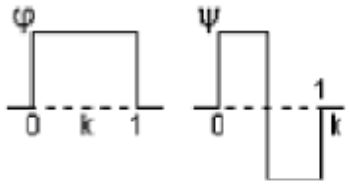
$$f(t_t) = q_m(t_i) + q_s(t_i) + \varepsilon_a(t_i) + \varepsilon_f(t_i).$$

Вейвлет-анализ предполагает представление трафика в различных масштабах. Это позволяет обнаружить характерные детали, которые могут оставаться незамеченными при одном разрешении. В таком виде вейвлет-модель имеет вид:

$$f(t_i) = \sum_{k=-\infty}^{\infty} c_{m,k}\phi_{m,k}(t_i) + \sum_{m=m'}^{\infty} \sum_{k=-\infty}^{\infty} d_{m,k}\psi_{m,k}(t_i),$$

где ϕ – масштабирующая функция, с помощью которой выполняется аппроксимация сетевого трафика (Рис. 7.13, а), ψ – вейвлет-функция, выделяющая детали трафика и его локальные особенности (рис. 7.13 ,б) $c_{m,k}$-

аппроксимирующие коэффициенты, $d_{m,k}$ детализирующие коэффициенты, вычисляемые с помощью алгоритма Маллата, $m, k \in I$ - параметры масштаба и сдвига. Первая сумма характеризует тренд и циклические компоненты, вторая – аномальность сетевого трафика.



а *б*

Рис. 7.13

Построение модели в режиме обучения позволяет сформировать эталонный ряд $f_э(t_i)$. Разница между ним и рядом $f_p(t_i)$, построенным в режиме анализа, определяет текущий уровень аномальности:

$$f_a(t_i) = \sum_{k=-\infty}^{\infty} \sum_{m=m'}^{\infty} d^p{}_{m,k}\psi_{m,k}(t) - \sum_{m=m'}^{\infty}\sum_{k=-\infty}^{\infty} d^э{}_{m,k}\psi_{m,k}(t).$$

Как упоминалось выше, вейвлет-анализ состоит из двух этапов. Сначала находится разложение функции $f(t)$= $\{x_n\}$ (дискретный сигнал), т.е. нахождение наборов коэффициентов разложения и, собственно, анализ этих коэффициентов.

Для анализа возможно использование ортонормированного вейвлета Хаара. В этом случае имеем две функции: масштабирующую функцию $\phi(x)$ и вейвлет-функцию $\psi(x)$. Для каждого уровня разложения $j(j \in Z)$ система функций $\phi_{j,n}(x) = \sqrt{2^j}\ \phi(2^j x - n)$, где $n \in Z$ является ортонормированным базисом пространства $V_j \subset L^2(R)$.

Обозначим символом cA и cD наборы коэффициентов разложения и рассчитаем с помощью операции свёртки исходную функцию с функцией $\phi(x)$ (для получения cA) и с функцией $\psi(x)$ (для получения cD):

$$a_{j-N,k} = (f, \phi_{j-N,k}) = \int_R f(x)\overline{\phi_{j-N,k}(x)}dx.$$

$$d_{j-m,k} = (f, \psi_{j-m,k}) = \int_R f(x)\overline{\psi_{j-m,k}(x)}dx.$$

При сложении всех коэффициентов на одной глубине разложения получаем сглаживающую функцию этого уровня:

$$P_j(f) = \sum_{k=1}^{Z} a_{j-N,k}\phi_{j-n,k}(x) + \sum_{j=1}^{Z}\sum_{k=1}^{Z} d_{j-m,k}\psi_{j-m,k}(x).$$

Полученный набор коэффициентов разложения несет качественную информацию о сигнале:

- $cA_N = \{a_{j-N,n}\}$ - описывает более грубое (сглаженное) приближение функции $f(x)$ в пространстве V_{j-N}- аппроксимирующие коэффициенты;
- $cD_m = \{d_{j-m,n}\}$ - характеризует отклонения приближения функции $P_j(f)$ относительно $P_{j-N-1}(f)$ - детализирующие коэффициенты.

Повторяя процедуру N раз, получаем вейвлет-разложение аппроксимации j - ого уровня разрешения $P_j(f)$ в виде серии коэффициентов (рис. 7.14):

$$P_j(f) = \{cA_N, cD_N, cD_{N-1}, \dots cD_1\}.$$

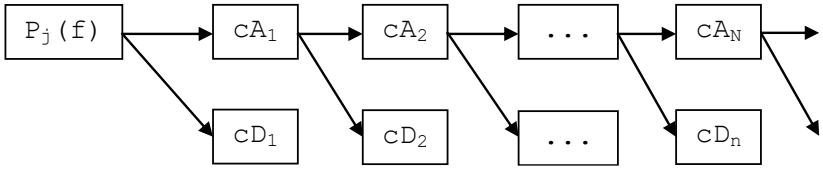


Рис. 7.14

Следует учитывать, что при высоком уровне разрешения j носители функций $\phi_{j,k}(x)$ и $\psi_{j,k}(x)$ становятся малыми (порядка $1/2^j$), что затрудняет вычисление интеграла с необходимой точностью. Для решения этой проблемы используется быстрое вейвлет-преобразование (БВП).

Быстрое вейвлет-преобразование (БВП), предложенное Стефаном Малла, даёт возможность вычислять коэффициенты вейвлет-разложения без интегрирования, используя алгебраические операции на основе свёртки.

Если начальные коэффициенты h_n и g_n фильтров вейвлетов $\phi_{j,k}(x)$ и $\psi_{j,k}(x)$ известны, то общий вид итерационных формул будет иметь следующий вид:

$$a_{j-1,k} = (f, \phi_{j-1,k}) = (f \sum_n h_n \phi_{j,n+2k}) = \sum_n h_n a_{j,n+2k},$$
$$d_{j-1,k} = (f, \psi_{j-1,k}) = (f \sum_n g_n \psi_{j,n+2k}) = \sum_n g_n a_{j,n+2k},$$

Равенства обеспечивают быстрые алгоритмы вычисления вейвлет-коэффициентов. Это объясняется тем, что в них используются более простые алгебраические процедуры. При этом число новых коэффициентов остаётся

прежним. Процедуру разложения можно повторить и таким образом найти все коэффициенты вейвлет-разложения глубины N.

Анализ изменений характеристик вейвлет-разложения, в частности, изменений детализирующих коэффициентов разложения, показал возможность выявлений аномалий в сетевом трафике. При уменьшении коэффициента отношения исходных данных и увеличении коэффициента отношения детализирующих данных, полученных в ходе применения вейвлет-анализа, можно предположить об аномальном поведении трафика и возможном проведении различных атак.

Сингулярный спектральный анализ сетевого трафика. Применение анализа сингулярного спектра (SSA – Singular Spectrum Analysis), называемого также «Гусеницей», позволяет преобразовать одномерный временной ряд в многомерный и исследовать полученные составляющие методом главных компонент. Преимущества метода заключаются в том, что не требуется стационарности ряда, знания математической модели тренда и периодических составляющих. Как правило, при этом метод позволяет выделить характерные слагаемые исследуемого ряда – тренд (медленно меняющаяся величина), периодические составляющие разных частот, случайные отклонения. Часто при применении метода можно выявить особенности составляющих ряда, не являющиеся на первый взгляд очевидными. Для нахождения периодических составляющих чрезвычайно большую визуальную информацию дает изучение двумерных графиков, аналогичных фигурам Лиссажу, когда по осям *x* и *y* откладываются различные пары собственных векторов или главных компонент. Оценка главных компонент вычислительного процесса методом «Гусеница», позволяет разбить набор главных компонент на две группы, первая из которых отвечает за формирование основной составляющей сетевого трафика, а вторая интерпретирована как шум.

Модель сетевого трафика хорошо восстанавливает исходные ряды, остатки имеют распределение, близкое к нормальному, автокорреляционная функция остатков сильно убывает. Это позволяет выделить характеристики сетевого трафика, которые могут идентифицировать тип атаки, например характеристики (Рис. 7.14) можно разбить на группы:

1) реагируют на атаку типа HTTP flood

2) реагируют на атаку типа UDP flood

3)все характеристики реагируют на SYN flood

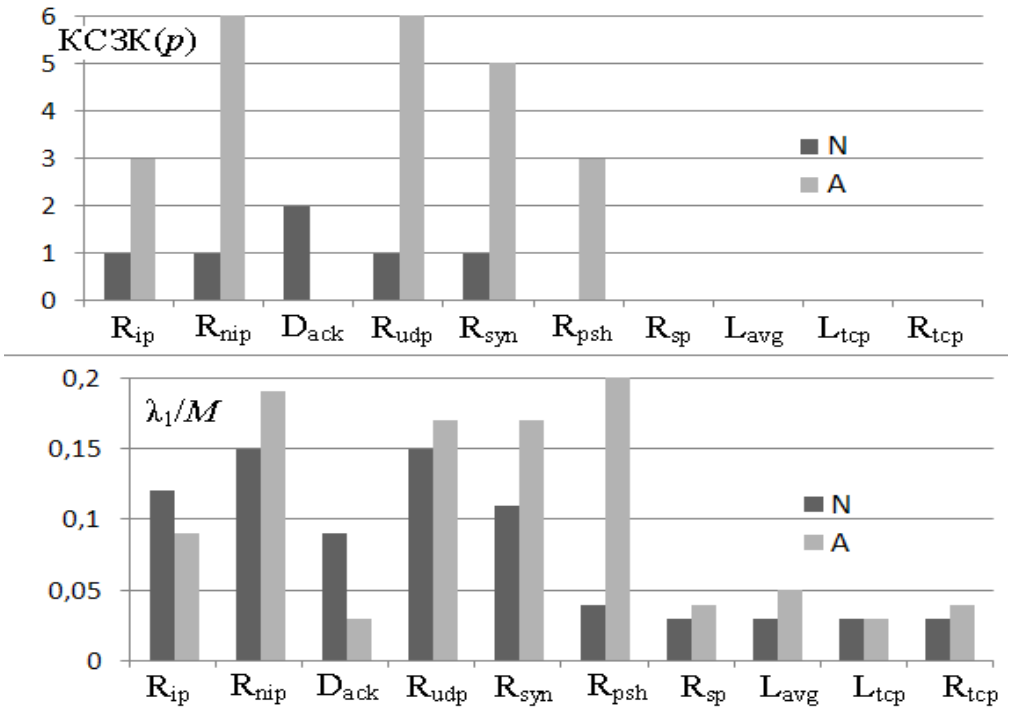


Рис. 7.14

Кластерный анализ. Идентифицирует характеристики, позволяющие осуществить классификацию сетевого трафика за счет группировки сходные входные данные, например с использованием сетей Кохонена, в которых производится кластеризация объектов, описываемых количественными характеристиками. В этом случае можно рассмотреть множество характеристик сетевого трафика как $S = \{s_1, s_2, \dots, s_l\}$, а поведение сетевого устройства вектором $X = \{x_1, x_2, \dots, x_n\}$.

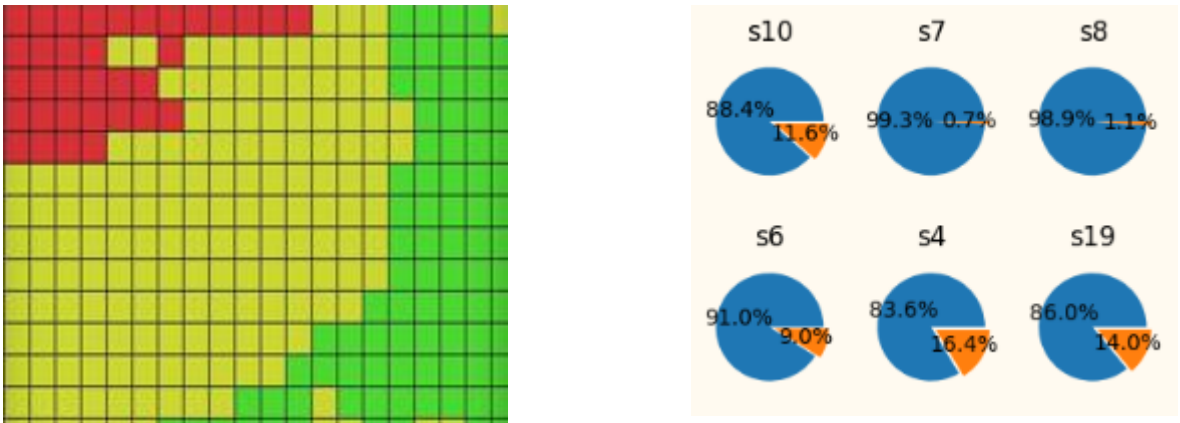
Возникает необходимость построить множество кластеров C и отображение F множества S на множество C, то есть $F: S \to C$.

Задача кластеризации по типу сетевого трафика состоит в построении множества $C = \{c_1, c_2, \dots, c_k, \dots c_M\}$, где c_k,— кластер, содержащий схожие по характеристикам объекты из множества S:

$$c_k = \{s_i, s_j | s_i \in S \; and \; d(s_i, s_j) < h\},$$

где h — величина, определяющая меру близости для включения объектов в один кластер, d_{ij}— мера близости между объектами, называемая расстоянием. Если расстояние d_{ij} меньше некоторого значения h, то характеристики считаются близкими и помещаются в один кластер. В

противном случае считается, что характеристики отличны друг от друга и их помещают в разные кластеры. На Рис 7.15 а изменение распределения кластеров в зависимости от характеристик сетевого трафика. Цвет отображает области, характеризующих уровень опасности (низкий риск, умеренный риск и высокий риск), который превышает нормативные показатели.



а) б)

Рис. 7.15

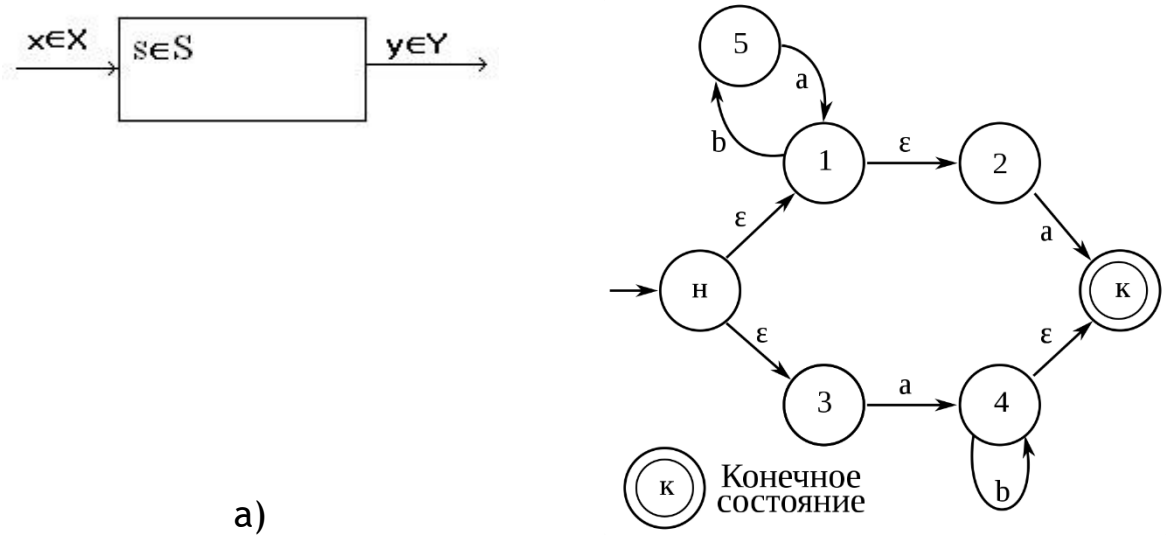
Использование кластеризации позволяет осуществить детальный мониторинг хостов, однако для оценки заданных характеристик целесообразно дополнять метод, используя статистические методы (Рис 7.15 б) обнаружения аномалий.

1.4 Сигнатурные методы обнаружения

Для выявления сетевой аномалии осуществляют сравнение с шаблоном поведения. Классификация сигнатурных методов обнаружения представлена на Рис 7.16. В общем виде каждый из этих методов можно представить в виде конечного автомата-модели дискретного устройства, имеющего один вход, один выход и в каждый момент времени находящегося в одном из состоянии из множества возможных (Рис. 7.16 а).

При анализе функционирования сетевого устройства полагают, что оно начинает работу в некотором начальном состоянии и последовательно

получает по одной команде, которое переводит устройство в новое состояние в соответствии с функцией переходов (Рис. 7.16 б)



а)

б)

Рис. 7.16

Это позволяет сформировать матрицу переходов, отклонение от которой характеризует нарушение безопасности. Применение аппарата конечных автоматов за счет оценки переходов из состояния в состояние позволяет осуществить:

- вычисление возможный сценариев/трасс атак;
- оценку корреляции последовательности событий;
- определение показателей защищенности;
- оценку рисков;
- определение мероприятий по защите.

В качестве общего примера можно рассмотреть процесс переходов из состояния в состояние на примере шаблона поведения злоумышленника (Рис. 7.17). Для формирования атаки необходимо осуществить подготовительные мероприятия, представленные набором последовательной из входного воздействия конечного автомата. Это позволяет сформировать набор действий для изменения внутреннего состояния атакуемого объекта и/или в зависимости от успешности скорректировать сигнатуры атак.

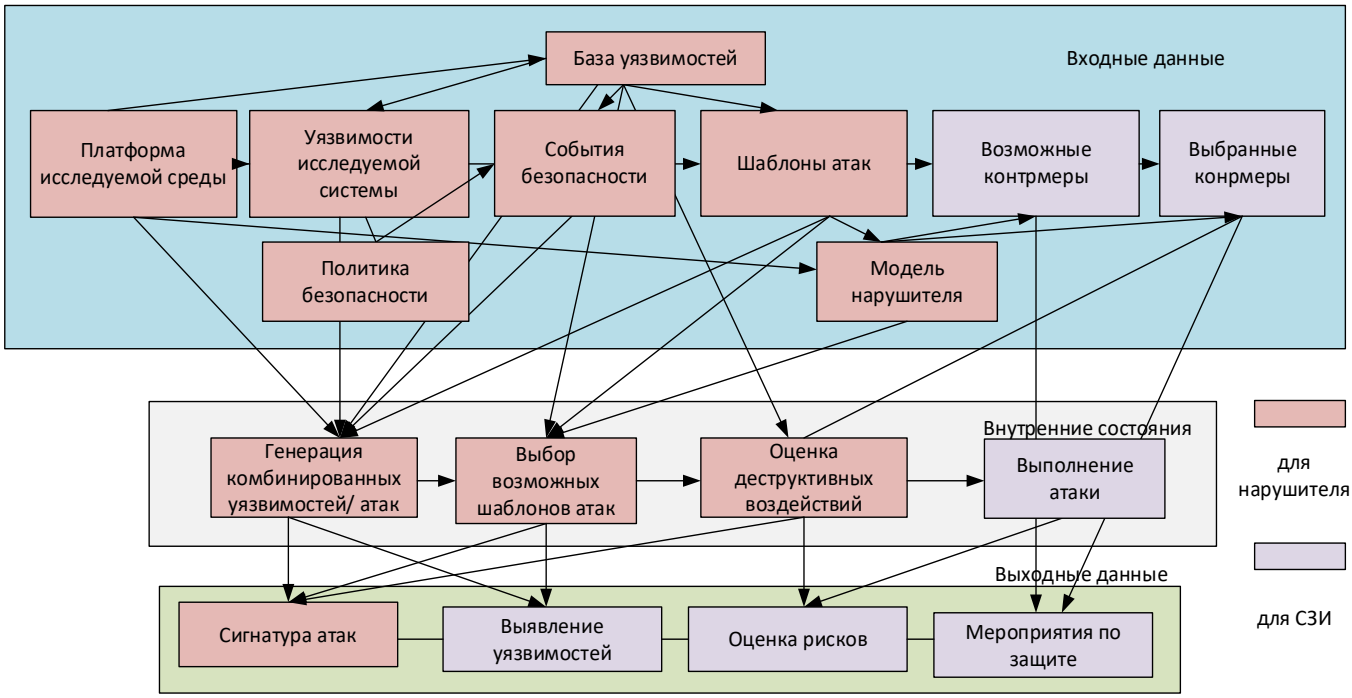


Рис. 7.17

Использование функции переходов позволяет сформировать граф атак (Рис 7.18). Граф атак – это граф, представляющий всевозможные последовательности действий нарушителя для достижения угроз (целей). Такие последовательности действий называются трассами (путями) атак. Например, на Рис 7.18 представлено две трассы атак при достижении цели из пункта 1 в пункт 8. На граф-схеме показан пример, который позволяет осуществлять скрытый поиск подключенных в сеть хостов, с возможностью поиска открытых портов, что дает преимущества при обнаружении единичного агента, поскольку в целом распределенный атакующий комплекс продолжит достижение целевой функции, характеризующиеся следующими состояниями: 1. выбор объекта и портов для атаки; 2. отправка корректных запросов; 3. отправка некорректных запросов; 4. принятие корректных данных; 5. принятие отказа в данных; 6. отсутствие ответа; 7. анализ результатов; 8. коррекция проведения атаки.

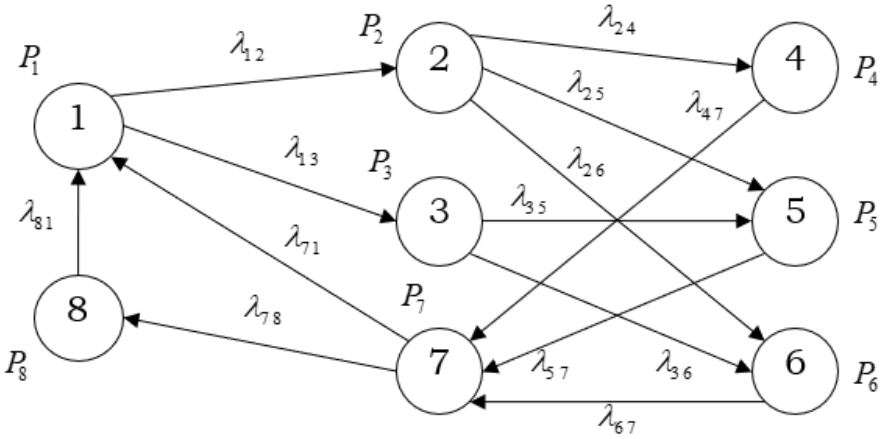
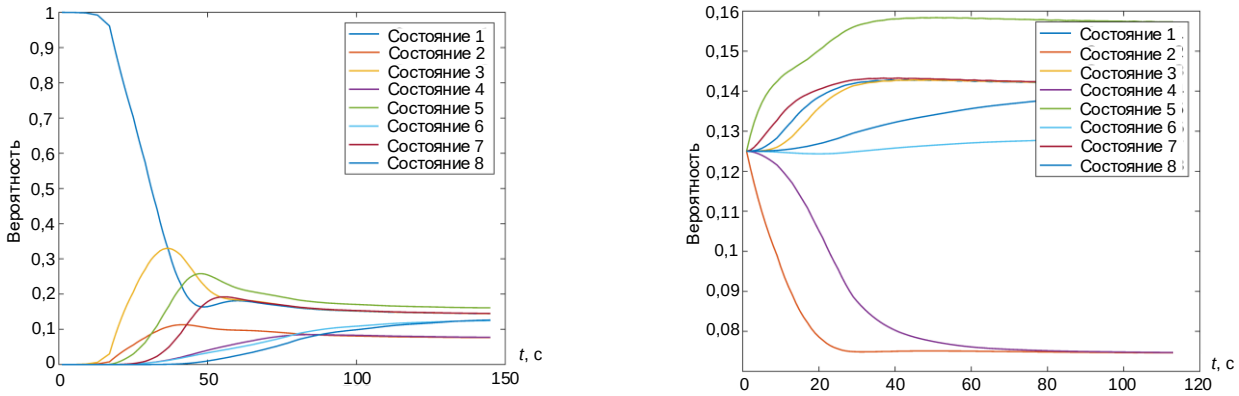


Рис. 7.18

Эту граф-схему можно описать системой дифференциальных уравнений, на основании которых осуществить оценку вероятностей перехода из состояния в состояние. Для нахождения оптимальной конфигурации многоагентной распределенной сети, позволяющей осуществить параллельность проведения атаки, было рассмотрено два варианта системы с различными начальными состояниями для распределенных агентов сети (Рис. 7.18 а, б). В первом варианте распределенная атакующая система находится в частично спящем состоянии, когда всю работу выполняет лишь один агент. Второй вариант описывает поведение распределенной атакующей системы, находящейся в состоянии, когда функционирует множество агентов. На рисунке 7.18 а выбраны начальные состояния [1; 0; 0; 0; 0; 0; 0; 0] – начало атаки с первого состояния. Траектория развития атаки показала, что переходный процесс в целом завершен к 40-й секунде. На рисунке 7.18 б выбраны начальные состояния [0,125; 0,125; 0,125; 0,125; 0,125; 0,125; 0,125; 0,125] – равномерное распределение начальных состояний. Траектория развития атаки показала, что переходный процесс в целом завершен к 40-й секунде. В обоих случаях наибольшую вероятность имеет 5-е состояние, наименьшую – 2-е и 4-е. Оценка переходного процесса, согласно графикам, позволяет предположить, что второй вариант более предпочтителен для реализации многоагентной распределенной сети, поскольку система приходит в устойчивое состояние раньше и соответственно вероятность обнаружения атаки ниже для второго случая. Оценка поведения кривых на рисунках показывает, что с момента завершения переходного процесса

кривые имеют одинаковую динамику и вероятности практически сравниваются по значению.



а) б)

Рис. 7.18

Анализ представленных результатов показал, что выявление распределенных атакующих элементов является достаточно сложной задачей. При этом в этих условиях первоочередной задачей является выявление координационного центра, осуществляющего атаку, поскольку его обнаружение позволит в значительной мере нейтрализовать деструктивные воздействия на сеть.

Дальнейший анализ функционирования сети целесообразно осуществлять в рамках поиска всего комплекса показателей, характеризующих ее работу. Это позволит не только осуществить поиск распределенных атакующих элементов сети, но сформировать комплекс мероприятий по защите как элементов, расположенных в сети, так и информации, циркулирующей в ней.

Минимизация последствий выявления атаки возможна через математическое ожидание выявления агента защищаемыми элементами сети:

$$M_0 = \sum_{i=0}^{N} \gamma_i P_i(j),$$

где $P_j(j)$ – вероятность выявления системами безопасности на каждом элементе, при j-ом методе защиты; $i = \overline{1,N}$ – количество представленных в системе элементов; γ_i – значение ущерба от нарушения на -ом элементе.

1.5 Нейронные сети

Нейросетевые методы позволяют удовлетворить высокие требования к производительности за счет автоматического нахождения сложных взаимосвязей в необработанных данных, это повышает точность

распознавания в сравнении с сигнатурными методами за счет устранения проблемы постоянного обновления актуальных баз данных сетевых угроз. В общем виде метод распознавания атак может быть представлен в виде:

1. Формирование наборов данных, описывающих период сетевой активности для организации обучения нейронной сети и валидации результатов обучения. Если входящие и исходящие информационные потоки представить в виде многомерного временного ряда, то отклонения от статистических закономерностей, характеризующих ряд, позволяет предположить сетевую атаку. На каждом шаге временного ряда необходимо выставить метку, сигнализирующую об опасности или безопасности сетевого трафика, с распознаванием соответствующего типа угроз в случае наступления инцидента. Формально, модель представления данных можно представить себе следующим образом:

$$\begin{matrix} y_1 & \dots & y_n \\ \uparrow & \uparrow & \uparrow \\ x_{1,1} & \dots & x_{n,1} \\ \dots & \dots & \dots \\ x_{1,k} & \dots & x_{n,k} \end{matrix}$$

где n-длина исследуемого временного ряда, а k - количество признаков, y_i- числовые метки, характеризующие тип атаки, $i \in \overline{1,n}$ а $x_{i,j}$ - значение j призака сетевого трафика, $j \in \overline{1,k}$ на временном шаге i.

Тогда, для распознавания типа сетевых атак в модели необходимо для каждого i-ого шага предусмотреть метки целевого класса, соответствующего определённому типу угроз, а также класс, сигнализирующий о том, что трафик безопасный. Однако, для наиболее полного анализа состояния сети на i-ом шаге необходимо учитывать не только признаковое описание текущего шага, но и предыдущих. Наиболее подходящим решением в подобной ситуации будет учет не всей предыдущей последовательности, а только промежутка (окна) определённой длины, что позволяет учитывать временной контекст, с ограничением потребляемых ресурсов. В этом случае, на вход нейронной сети подается k последних шагов временного ряда, а предсказание метки целевого класса осуществляется на последнем шаге.

2. Подготовка данных для обучения нейронной сети (Рис. 7.19) Предобработка данных увеличивает точность распознавания и

предотвращает сбои в формате входных данных. Разделение данных на тренировочную (для построения системы распознавания угроз) и тестовую (для оценки и получения численных характеристик качества обученной модели) части.

Предобработка исходного набора данных, для получения информации о признаковых описаниях сетевого трафика на каждом временном шаге и отбор конечных признаков;

Избавление от пропусков и возможных неожиданных значений признаков, как например отрицательные, если признак характеризует количество каких-либо единиц;

Нормализация данных. Эта процедура необходима как во время обучения нейронной сети, так и во время работы модели в режиме реального времени для того, чтобы распределения данных, на которых модель обучалась были близки к тем, которые приходиться обрабатывать;

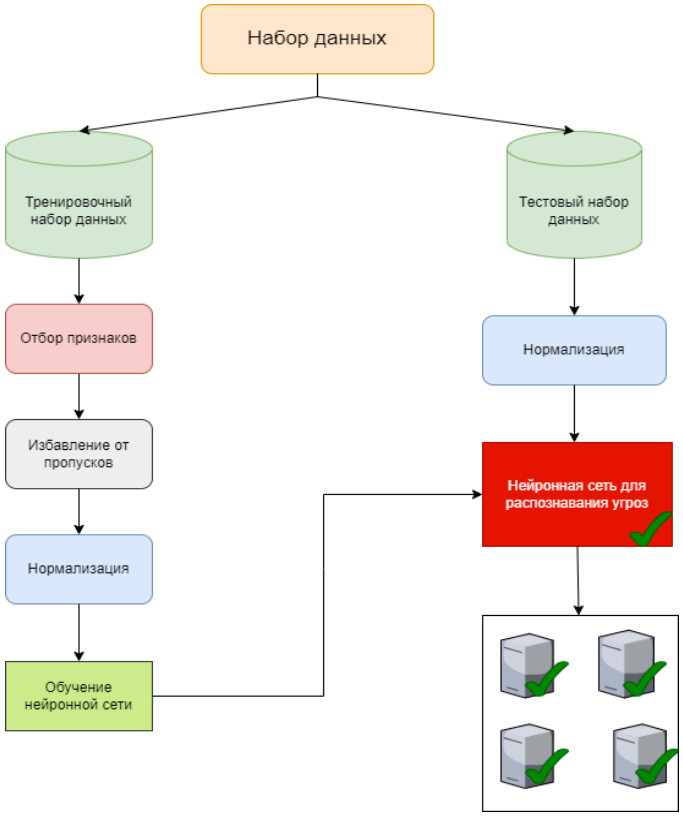


Рис. 7.19

3. Обучение нейронной сети. На данном этапе происходит последовательная подача на нейронную сеть элементов обучающей выборки, считается значение выбранной функции ошибки и делаются

шаги обратного распространения для подстройки обучаемых параметров для её минимизации.

4. Оценка точности распознавания сетевых атак обученной моделью и интерпретация результата. Поскольку в работе стоит задача многоклассовой классификации, то для оценки качества используются такие метрики как: точность, полнота (Precision, Recall), F-мера и матрица ошибок. Все перечисленные метрики, кроме матрицы ошибок, считаются отдельно для каждого целевого класса. В матрице ошибок по строкам описываются истинные значение классов, а по столбцам – предсказанные, так на пересечении какой-либо строки и столбца будет количество элементов, для которых было предсказан класс, указанный в столбце, а верным будет класс, указанный в строке. В случае, если модель достоверно распознаёт тип атаки матрица ошибок будет диагональной.

Пример практической реализации программного модуля распознавания информационных угроз. Для исследования были взяты данные, представленные в открытый доступ Канадским университетом Нью-Брунсуика CIC-DDoS-2019, которые позволяют извлечь около 80 описательных признаков из текущего сетевого трафика за короткий промежуток времени (длина шага, количество пакетов, количество входящих и исходящих байт). Для распознавания были взяты следующие типы сетевых атак: DdoS MSSQL; DdoS LDAP; SYN flood; UDP flood; DdoS NetBIOS.

Шаг 1. Реализуемая модель будет обучаться на распознавание 6 классов, 5 из которых – это различные DdoS атаки, и специальный класс для трафика с нормальной активностью.

Шаг 2. В сформированных тренировочных и тестовых данных отобраны только те признаки, которые могут идентифицировать атаку (Таблица 2). Отброшены неинформативные признаки, представляющие собой константное значение, а также признаки, в которых содержится большое количество пропусков. Также были удалены признаки, характеризующие конкретные соединения, такие как номер порта и IP-адрес.

Таблица 2

Имя вещественного признака	*Тип данных признака*	*Описание*
ACK Flag Count	Бинарный	Присутствие ACK сообщений
Average Packet Size	Вещественный	Средний размер пакета в входящем и исходящем трафике
Avg Bwd Segment Size	Вещественный	Средний размер сегмента, на который разделены входящие пакеты для передачи по сети
Avg Fwd Segment Size	Вещественный	Средний размер сегмента, на который разделены исходящие пакеты для передачи по сети
Bwd IAT Min	Целочисленный	Минимальное время, которое приходит после получения пакета до прибытия следующего, мс
Down/Up Ratio	Вещественный	Отношение входящего трафика к исходящему

Шаг 3. Подготовка данных для передачи на нейронную сеть. Для каждого из 6 распознаваемых классов (5 для различных атак и 1 для безопасного трафика) было взято по 150 000 последовательностей векторов, каждая из которых имеет длину в 30 элементов, в которых необходимо будет предсказать класс последнего элемента. Такой размер был выбран ввиду достаточной скорости обучения и обработки таких данных в реальных условиях: время обработки одной такой подпоследовательности составляет 0.007 секунд на процессоре Intel Core i5-10310U с тактовой частотой 1.7 ГГц, что не превышает время между двумя временными шагами, получаемыми с помощью CICFlowMeter. Равное количество элементов обучения для каждого класса взяты для того, чтобы избежать проблемы несбалансированности классов, которая может привести к тому, что нейронная сеть во время обучения хуже подстроится под закономерности, характерные для малочисленных классов, что приведёт к уменьшению точности распознавания.

Для валидационной и тестовой выборки аналогичным образом было взято по 5 000 и 10 000 элементов. Таким образом размер выборок после разбиения представлен в Таблице 3:

Таблица 3

Название выборки	*Количество элементов*
Тренировочная	800 000

Валидационная	30 000
Тестовая	60 000

Для снижения некорректной оценки финальных результатов распознавания, элементы тестовой выборки взяты не из тех же файлов, которые использовались для тренировки, а из файлов, записанных в другие дни.

Шаг 3.1. Обработка пропущенных значений, которые составили около 5% от всех данных и удалены из обучающей выборки ввиду предположения, что пропущенные временные шаги не приведут к занчительному падению точности модели.

Шаг 3.2. Нормализация количественных признаков проводилась на тестовых данных и во время работы нейросети в реальных условиях. Для вычисления нормализованных значений использовались медиана и межквартильных размах, которые использовались вычислялись на тренировочной выборке.

Шаг 4. Реализация модели и процесс обучения проводилась с использованием класса torch.nn.Module, который является базовым для создания собственных нейронных сетей. Для описания архитектуры модели также использовался модуль torch.nn, предоставляющий возможность переиспользования многих реализованных слоёв нейронной сети. Сеть представлена на рекуррентной архитектуре GRU (Gated Recurrent Unit), которая является одной из модификаций архитектуры LSTM (long short-term memory). Модель позволяет уйти от проблем долгосрочных зависимостей и затухающих градиентов, присутствующих у стандартных рекуррентных блоков, за счет уменьшения количества матриц обучаемых весов, состоящих из обучаемых параметров. Конечная схема обработки данных в нейронной сети представлена на Рис.7.20

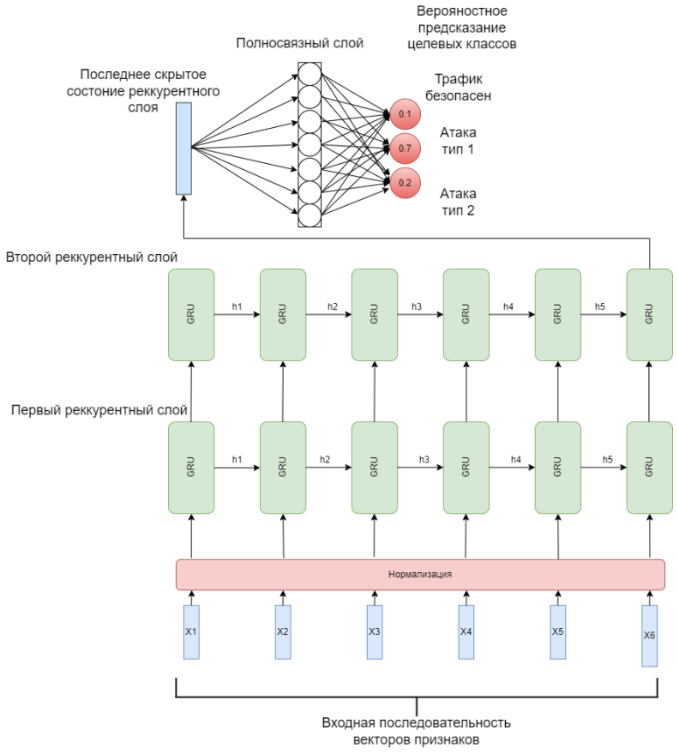


Рис. 7.20

1) На вход модели подается последовательность векторов фиксированной длины с характеристиками сетевого трафика. Каждый элемент последовательности представлен в виде вектора с значениями после нормализации данных;

2) Полученная последовательность векторов характеристик описания сетевого трафика обрабатывается с использованием двух последовательных реккурентных блоков GRU. Скрытые состояния h_t на каждом временном шаге первого реккурентного блока являются входными векторами второго блока GRU. В результате этого этапа не происходит изменений длины последовательности, обрабатывается только содержимое признаков и извлекается информация из последовательности.

3) Последнее скрытое состояние второго слоя GRU описывает в латентном представлении всю извлеченную информацию из последовательности, которую необходимо преобразовать в метку целевого класса, идентифицирующего атаку. Для этого последнее скрытое состояние пропускается через полносвязный слой, размерность выхода которого совпадает с количеством целевых классов, с функцией активации.

4) На этапе обучения после каждого обработанного окна вычисляется значение функции ошибки модели, которое должно быть минимизировано следующим образом:

$$L(\theta, x, y) = -\sum_{k=1}^{K} y_k \cdot f(\theta, x)_k \rightarrow \min_{\theta}$$

где f - обучаемая нейронная сеть, K – количество целевых классов, y_k – истинная бинарная метка 0 или 1 для определенного целевого класса, θ – обучаемые параметры модели, x – входные данные $f(\theta, x)_k$ – предсказанное нейронной сетью значение двоичного логарифма вероятности определенного целевого класса. Гиперпараметры для обучения представлены в таблице 3.

Таблица 4.

Наименование параметра	Значение параметра
Размер мини-батча	16
Размер окна последовательности	30
Алгоритм стохастического градиентного спуска	Adam
Скорость обучения	0.001
Размерность скрытого состояния первого реккурентного слоя GRU	64
Размерность скрытого состояния второго реккурентного слоя GRU	64
Размерность входного вектора	65

Процесс обучения нейронной сети продолжался в течение 7 эпох, после каждой эпохи было посчитано среднее значение функции ошибки на валидационной выбрке, данные предаставлены на Рис. 7.21

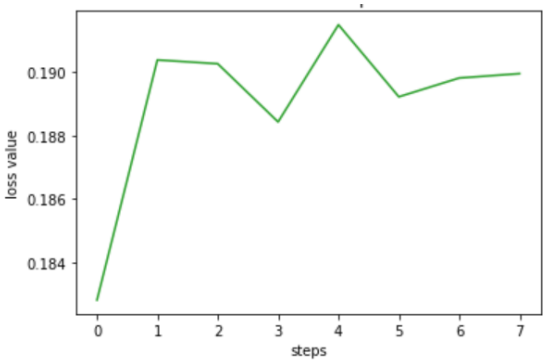


Рис. 7.21

Из рисунка видно, что наименьшее значение функции ошибки на валидационных данных приходится на эпоху с индексом 3, после чего происходит увеличение данного значения в связи с эффектом переобучения, описанным выше. Наибольшее значение ошибок приходится на распознавание

атаки DdoS NetBIOS, остальные атаки распознаются верно в более чем 90% случаев. Матрицы ошибок модели представлены на Рис. 7.22.

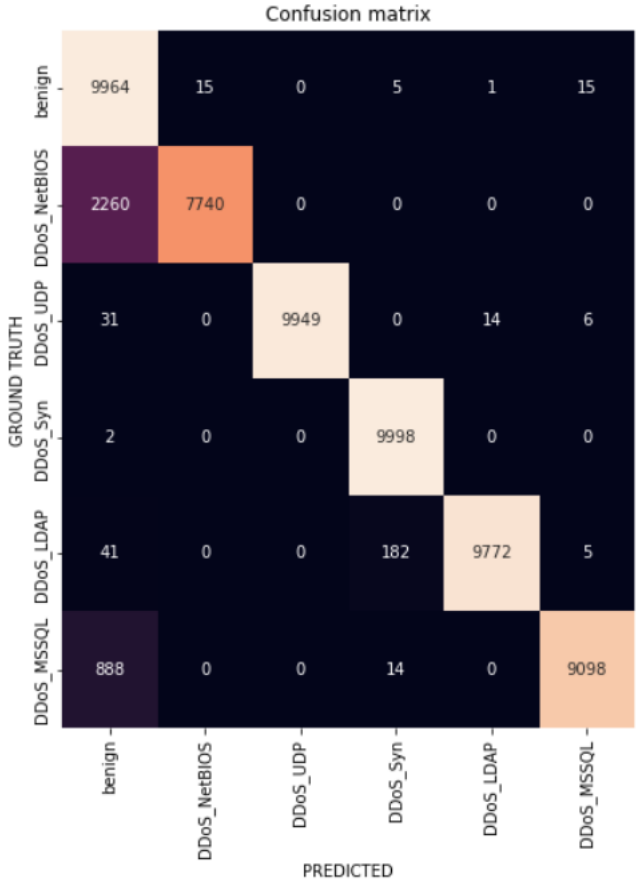


Рис. 7.22

Наиболее частая ошибка модели, как видно из матриц – это в 22,6 % атака DdoS NetBIOS и в 8.9% атака DdoS MSSQL принимаются за безопасный трафик. Однако при продолжительной DdoS атаке данных типов их удастся распознать, так как значительная часть трафика этих классов распознаётся без ошибок. Также только на 0.36% безопасного трафика была обнаружена угроза, что говорит о том, что разрабатываемый модуль не будет подвержен проблеме частых ложных срабатываний.

1.6 Методы прогнозирования сетевого трафика

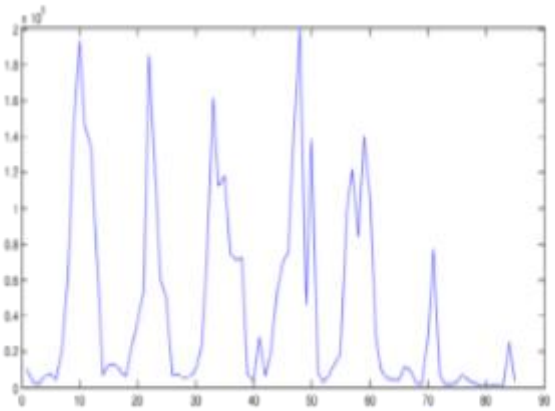
Прогнозирование функционирования сети в течение длительного времени при изменяющихся динамически параметрах не позволяет строить долгосрочных прогнозов. Альтернативным способом является использование методов краткосрочного прогнозирования с учетом времени полураспада и старения информации о параметрах измеряемых величин. При прогнозировании необходимо рассматривать изменение характеристик функционирования, как на системном, так и на сетевом уровне, а также

структуру самого информационного воздействия в виде сетевого потока, который может нарушить функционирование сети. При этом целесообразно оценить параметры регулярности, так как величина того или иного цикла непосредственно влияет на выбор метода прогнозирования и модель прогноза. Экспериментально было показано, что регулярность характеристик сетевого трафика можно оценить, рассчитав долю *k* ненулевых значений во время периода проведения измерений (Таблица 5):

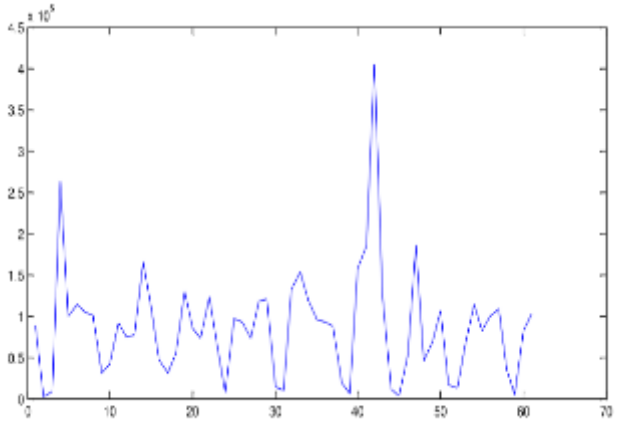
Таблица 5

Величина	k, %	Величина	k, %	Величина	k, %	Величина	k, %
$V_{DNS,i}$	69	$N_{ARP,rep}$	13	$N_{NO-TCP/IP}$	13	N_{D+UF}	48
$V_{DNS,o}$	35	$N_{ARP,req}$	13	$N_{RST,i}$	81	N_{UF}	1
$V_{TCP,i}$	100	$N_{DNS,i}$	92	$N_{RST,o}$	80	$N_{TCP/IP,i}$	100
$V_{TCP,o}$	99	$N_{DNS,o}$	79	$N_{SYN,i}$	100	$N_{TCP/IP,o}$	100
$V_{UDP,i}$	100	$N_{FIN,i}$	99	$N_{SYN,o}$	99	N	100
$V_{UDP,o}$	100	$N_{FIN,o}$	96	$N_{SYN+ACK}$	99	$N_{UDP,i}$	100
$V_{HTTP,i}$	88	N_{ICMP}	95	N_{TCP}	100	$N_{UDP,o}$	100
$V_{HTTP,o}$	77	$N_{ICMP,i}$	87	$N_{TCP,i}$	100	$N_{UDP,s}$	97
$N_{ACK,i}$	100	$N_{ICMP,o}$	57	$N_{TCP,o}$	100	$N_{HTTP,i}$	97
$N_{ACK,o}$	100	$N_{ICMP,s}$	5	$N_{TCP,s}$	15	$N_{HTTP,o}$	90

Как видно из таблицы, наиболее стабильная загрузка сети по протоколу TCP/IP обеспечивается протоколами TCP, UDP, HTTP. Для рассматриваемых временных рядов характерна ярко выраженная суточная и недельная периодичность. На примере $N_{TCP/IP,i}$ (рисунки 7.23 а (суточная периодичность в течение недели. Один отсчёт соответствует 2 часам трафик),б (Недельная периодичность в течение 2 месяцев. Один отсчёт соответствует суткам)):



а)

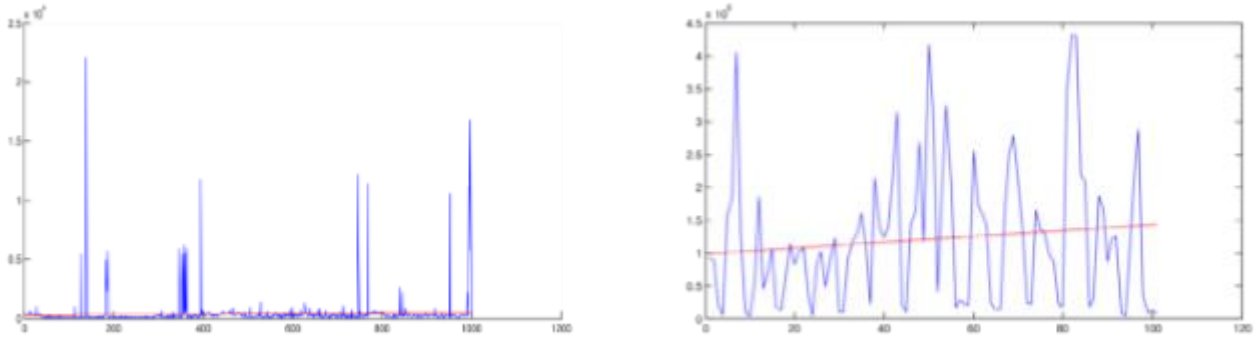


б)

Рис. 7.23

Стационарность рассматриваемых рядов, т. е. изменчивость во времени функций распределения случайных величин, генерирующих ряд. Это еще раз подтверждает, что не все методы прогнозирования эффективно работают с такими рядами.

Тренд, напротив, не имеет столь яркой выраженности. Тем не менее, для большинства измеряемых величин характерно постепенное возрастание (Рис. 7.24 а (анализ тренда в трафике. Отсчёт равен 10 минутам), б (анализ тренда в трафике. Отсчёт равен 1 суткам)).



а) б)

Рис. 7.24

Формирование требований к методам прогнозирования. В качестве общих требований к методам решения поставленных задач можно сформулировать наличие минимального значения ошибки прогнозирования и, как частный случай этого критерия, адекватность прогноза, т. е., соответствие диапазона прогнозируемых значений диапазону реальных данных. Помимо этого, структура трафика накладывает следующие ограничения на методы прогнозирования:

1. учет периодичности сетевого трафика;
2. учет наличия тренда;
3. выполнение сглаживания резких скачков и случайных отклонений в трафике (обусловленных, например, праздничными днями или выходом оборудования из строя). Отклонения не должны существенно изменять картину прогноза;
4. возможность накопления истории сетевой активности;
5. возможность работы с нестационарным временным рядом.

При выборе метода прогнозирования информационных потоков следует учитывать следующие ограничения:

1. Следует использовать методы краткосрочного прогнозирования, так как атака достигает максимальной мощности в скором времени после её возникновения.

2. Информационный поток является изменчивым и не имеет постоянного распределения. Таким образом, метод должен оперировать нестационарными временными рядами.

При построении краткосрочного прогноза основная проблема сводится к оценке детерминированной и случайной составляющих прогноза $Y(t) = T(t) + e(t)$, длины предыстории (общее количество точек ряда, используемых при построении) и горизонта прогноза. Ограничение перечня статистических методов, применяемых при обработке данных и характеризующих интенсивность потока, приводит к тому, что наиболее часто используемые в прогнозировании аналитические модели потока в основном строятся в предположении его стационарности. При этом наличие необходимого количества статистических данных не является решением проблемы прогнозирования возникновения сетевых аномалий и поведения сети под воздействием данных аномалий, поскольку с периодическим обновлением технологий передачи данных, в том числе и изменением протоколов передачи данных, происходит изменение статистических характеристик. Прогнозирование поведения сети требует построения специальных моделей прогноза. Часто используются модели, основанные на авторегрессии – AR, ARMA, ARIMA, FARIMA. Метод прогнозирования на основе моделей авторегрессии дает лучший результат для процессов со слабо выраженными или отсутствующими фрактальными свойствами. При повышении степени фрактальности прогноз осуществляется на свойствах самоподобных процессов, т.к. он лучше использует корреляционные связи, которые возрастают со степенью фрактальных свойств. Анализ прогнозирования сетевой активности моделями скользящего среднего, взвешенного скользящего среднего, экспоненциального сглаживания, двухпараметрической моделью Ч. Хольта, полиномиальной моделью второго порядка и моделью ARIMA показал, что наименьшей средней абсолютной

процентной ошибкой (MAPE = 13%) прогноза обладает модель ARIMA (4,1,3), но прогнозирование с помощью данной модели более чем на 2 шага значительно ухудшает показатель MAPE и приводит к увеличению стандартной ошибки прогноза (RMSE) до 1,59×107 бит/с. Значение параметра MAPE для оставшихся моделей лежат в пределах 20%, что соответствует хорошему прогнозу, а наименьшую стандартную ошибку прогноза показала модель экспоненциального сглаживания (RMSE = 1,08×107 бит/с). Таким образом, модель экспоненциального сглаживания имеет более высокую точность прогноза.

1.7 Системы обнаружения вторжений SIEM

Security information and event management (сокращенно SIEM) - это программное средство, осуществляющее анализ функционирования сети в реальном времени. SIEM используется также для журналирования данных и генерации отчетов в целях совместимости с прочими бизнес-данными. Как правило, SIEM системы представлены на базе архитектуры IDS и включают комплексные методы обнаружения сетевых аномалий. В общем виде SIEM системы включают технологии SIM и SEM (Рис. 7.25).

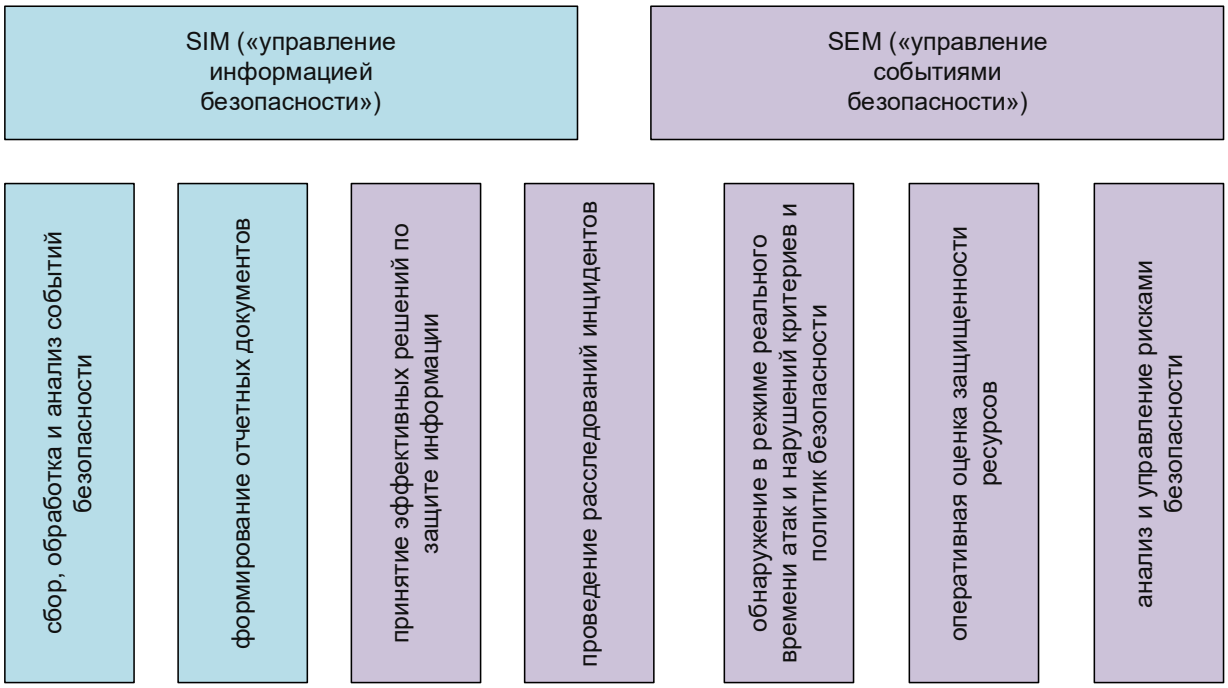


Рис. 7.25

При этом основное отличие SIEM- систем от IDS заключается в возможности осуществлять оценку рисков с выработкой мероприятий, по предотвращений вторжений (Рис 7.26).

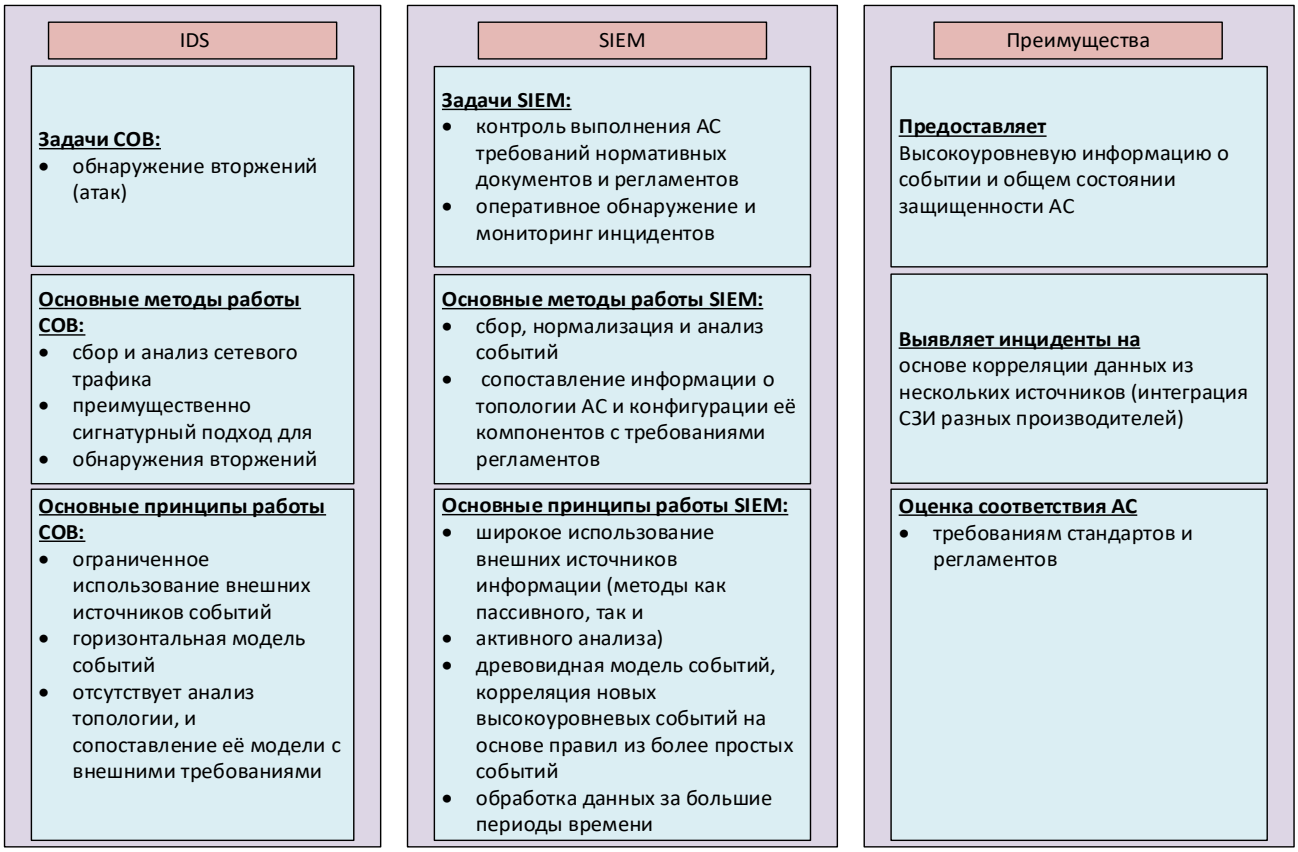


Рис. 7.25

Архитектура SIEM систем также идентична архитектуре IDS за исключение того, что содержит консоль управления и экспертного анализа с возможностью поиска по множеству журналов на различных узлах (результатом являются отчеты в предопределенной и произвольной форме, оперативная корреляция данных о событиях, а также выдаваемые предупреждения). При этом система может быть представлена в виде распределенной архитектуре, что предполагает большую производительность и лучшие возможности по масштабированию, а также позволяет развернуть SIEM-решение на инфраструктурах с несколькими управляющими устройствами. В этом случае SIEM представляют системы использующие многоагентный технологии. Агенты выполняют первоначальную обработку и фильтрацию, а также сбор событий безопасности. Передача информации от источников данных может осуществляться несколькими способами:

-источник сам инициирует передачу событий (например, отправляет по syslog-протоколу);

-события с источника забираются пассивно.

Это позволяет осуществлять сбор информации используя основные протоколы сбора информации (SNMPv2 and SNMPv3, Opsec, стек TCP/IP и др).

Протокол SNMP

SNMP (англ. Simple Network Management Protocol — простой протокол сетевого управления) — стандартный интернет-протокол для управления устройствами в IP-сетях на основе архитектур TCP/UDP.

К задачам SNMP относят:

- управление конфигурацией (Configuration Management);
- обработку ошибок (Fault Management);
- анализ производительности и надёжности (Performance management);
- управление безопасностью (Security Management);
- учёт работы (Accounting Management).

Решение указанных задач в протоколе построено с использованием многоагентной технологии, архитектура которого включает (Рис. 7.26):

- ***управляемое устройство*** — элемент сети, который разрешает однонаправленный (только для чтения) или двунаправленный доступ к конкретной информации об элементе.
- ***агент***- программный модуль сетевого управления, располагающийся на управляемом устройстве, либо на устройстве, подключенном к интерфейсу управления управляемого устройства. Агент обслуживает базу управляющей информации и отвечает на запросы менеджера SNMP.
- ***менеджер*** — это программное обеспечение, установленное на рабочей станции управления, наблюдающее за сетевыми устройствами и управляющее ими. В любой управляемой сети может быть один и более менеджеров.
- ***база управляющей информации*** — это совокупность иерархически организованной информации, доступ к которой осуществляется посредством протокола управления сетью.

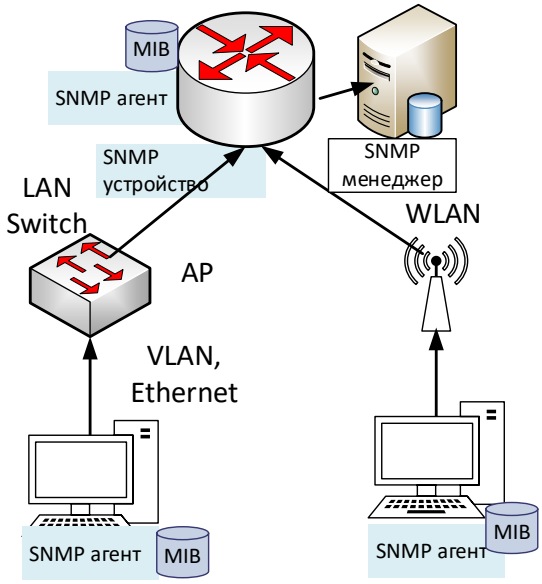


Рис. 7.26

Для идентификации агента используется база управляющей информации, которая описывает структуру управляющей информации устройств и состоят из управляемых объектов (переменных). Идентификаторы ID верхних уровней отданы организациям, контролирующим стандартизацию, а идентификаторы нижних уровней определяются самими организациями. Каждая ветвь дерева нумеруется целыми числами слева направо, начиная с единицы разделенных точкой, записанных слева направо и включает полный путь от корня до управляемого объекта. Обращение к управляемым объектам MIB происходит посредством идентификаторов объекта (Object IDentifier, OID) по уникальному идентификатору в пространстве имен OID, отвечающих за определенную ветку дерева OID и контролируемой агентством IANA (Рис.7.26). Например, 1.3.6.1.2.1.1- эквивалентное имя: iso.org.dod.internet.mgmt.mib-2.system.

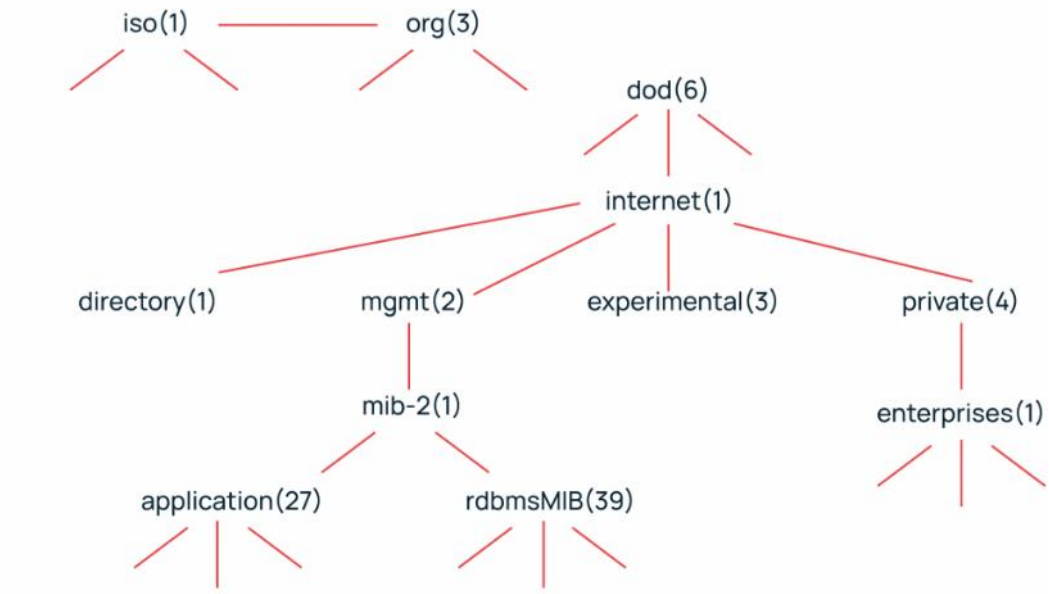


Рис. 7.27

Протокол SNMP предусматривает передачу между менеджером и агентом по UDP, с помощью сообщений (Рис. 7.28). К типичным сообщениям можно отнести: сведения о входящем/исходящем трафике на интерфейсе; ошибках при загрузке процессора; времени работы; температуре и другие OIDы, поддерживаемые производителем оборудования.

Протокол SNMP поддерживает сообщение типа «запрос — ответ», которыми обмениваются менеджер и агент:

- Get-Request — запрос значений одного или нескольких объектов;
- Get-Next-Request — запрос от менеджера к объекту для обнаружения доступных переменных и их значений;
- Set-Request — запрос на изменение значения одного или нескольких объектов
- Get(Set)-Reply — получение ответа от агента;
- Trap (ловушка) используется для асинхронного сообщения о событии, происходящем на управляемом сетевом устройстве;
- GetBulk- позволяет менеджеру получить несколько переменных за один запрос.

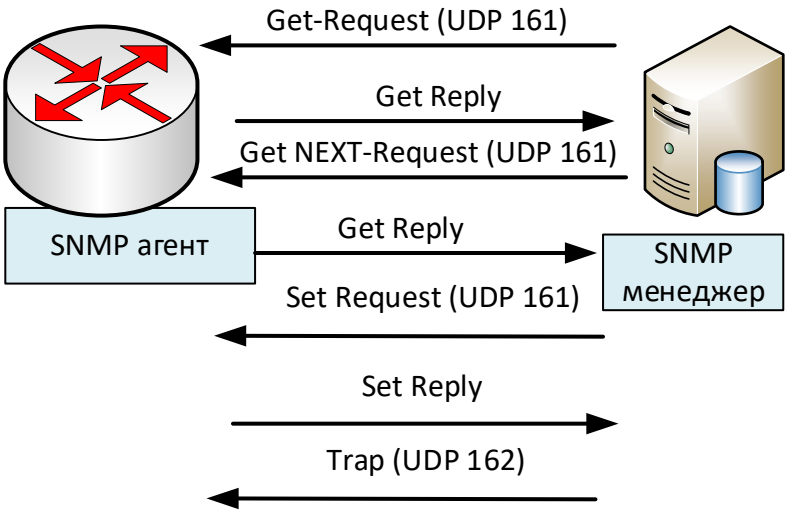


Рис. 7.28

Наибольший интерес представляют сообщения типа Trap. В SNMP есть два типа ловушек: Trap и Inform. Отличия между ними в том, что после получения Inform менеджер подтверждает получение ловушки. Рассмотрим их более подробно:

0 – **coldStart** – холодный запуск устройства;

1 – **warmStart** – горячий запуск устройства;

2 – **linkDown** – отключение интерфейса;

3 – **linkUp** – включение интерфейса;

4 – **authenticationFailure** –сообщение с неверной строкой сообщества;

5 – **egpNeighborLoss** – потеря связи по EGP;

6 – **entrpriseSpecific** –событие, характерное для производителя данного устройства.

Мониторинг указанных сообщений позволяет не только динамически хранить информацию о функционировании сетевых устройств, но осуществлять автоматический анализ доступности активов и наличия ресурсов, с возможностью оперативного контроля и анализ рисков, на основании которых осуществлять анализ и оценка соответствия требованиям нормативно-правовых документов (Рис 7.29).

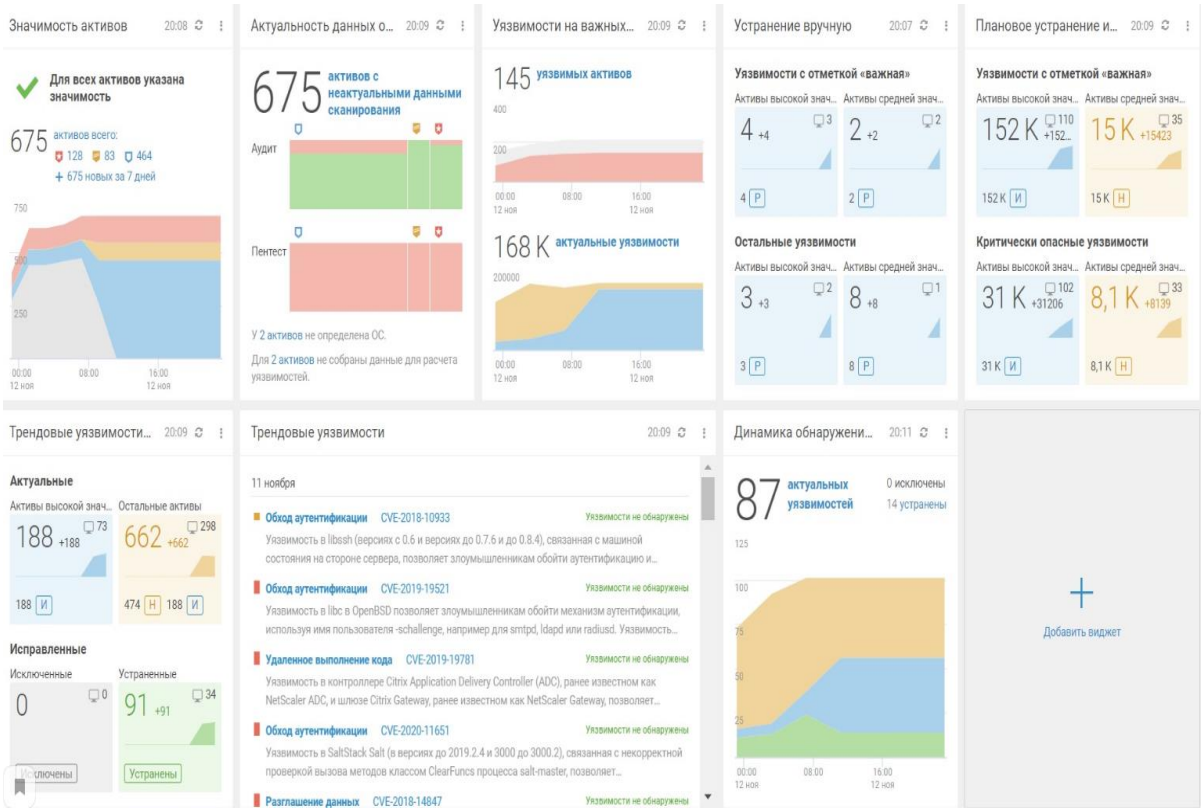


Рис. 7.29

1.8 Программно-конфигурируемые сети

Программно-конфигурируемые сети (Software Defined Networking/SDN) – это разделение *контура передачи данных* и *контура управления данными*, позволяющее осуществлять программное управление контуром передачи данных, которое может быть физически или логически отделено от аппаратных коммутаторов и маршрутизаторов.

Архитектура программно- конфигурируемой сети состоит из уровней (Рис 7.30):

- *инфраструктурный уровень*, на котором функционируют сетевые коммутаторы и каналы передачи данных

-*уровень управления* — набор программных средств, физически отделённых от инфраструктурного уровня, обеспечивающий реализацию механизмов управления устройствами инфраструктурного уровня

-*уровень сетевых приложений* — набор SDN-приложений, взаимодействующих с SDN-контроллером через программный протокол (API) для сбора, анализа, развёртывания и управления сетевой инфраструктурой на уровне приложений.

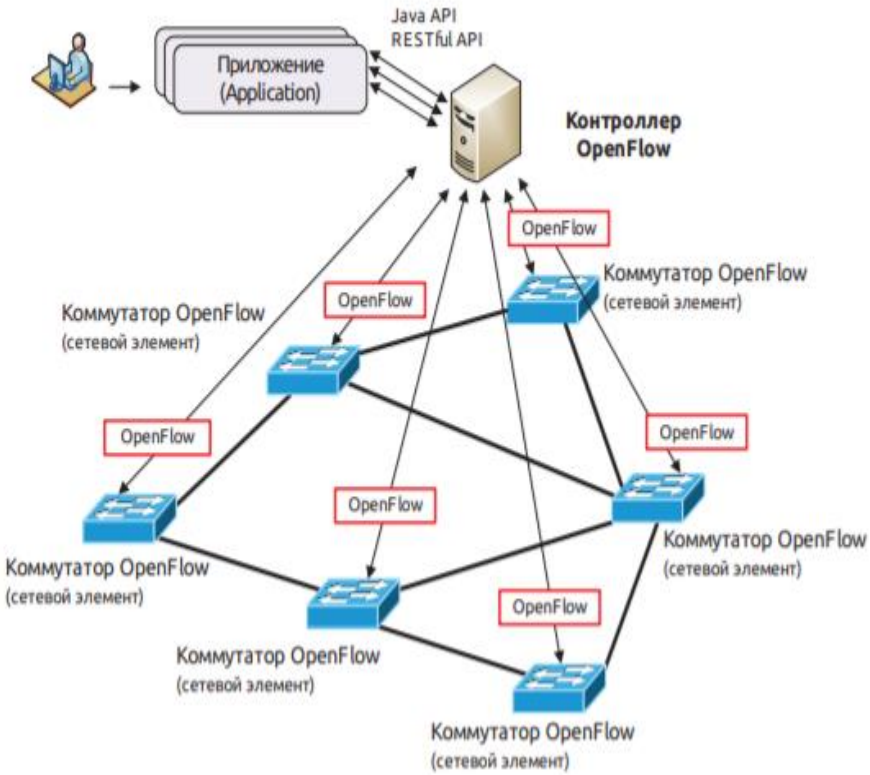


Рис. 7.30

Существуют базовые подходы к построению программно-конфигурируемых сетей:

- Open SDN реализуется на базе протокола OpenFlow, в которой предусмотрено полное удаление плоскости контроля (control plane) с сетевого оборудования. Плоскость контроля переносится на центральный SDN-контроллер (Рис. 7.30). Трафик на сетевом оборудовании передается на основании правил таблиц потоков, которые загружаются с SDN-контроллера, поскольку он обладает полным знанием о топологии сети и на основании за данных политик от SDN-приложений. Взаимодействие контроллера SDN с сетевыми элементами обеспечивается через «южный» интерфейс по стандартизированному открытому протоколу Openflow. Сетевое оборудование для работы в OpenSDN именуется OpenFlow-коммутаторами/маршрутизаторами. К преимуществам подхода OpenSDN на базе OpenFlow можно отнести возможность прямого доступа и изменения плоскости передачи на сетевых устройствах. Например неизвестный пакет отправляется на контроллер. Контроллер вычисляет лучший маршрут через всю сеть (с наименьшей стоимостью и удовлетворяющий политикам

маршрутизации). Соответствующие правила OpenFlow устанавливаются на коммутаторы с формированием обратного маршрут. К другому преимуществу можно отнести возможность динамической переконфигурацию в случае ошибки сети.

- SDN via API основное отличие заключается в сохранении плоскости контроля на сетевых элементах. Формирование сетевой топологии реализуется через управление конфигурацией оборудования, с влиянием на таблицы маршрутизации. Для этого SDN-контроллер взаимодействует с сетевым элементом либо через существующие протоколы (CLI, SNMP, PCEP и т. д.), либо через новые или усовершенствованные протоколы (Netconf/Yang, I2RS). Концепция программируемости контроллера через приложения через «северные» интерфейсы в данном подходе сохраняется. К преимуществам данного подхода можно отнести использование существующего оборудования и возможность реализации концепции SDN на существующих традиционных сетях. К минусам -отсутствие абстракции, поскольку возникает необходимость синхронизации между плоскостью контроля контроллера и распределенными плоскостями контроля каждого сетевого элемента.
- SDN via Overlay нашло свое применение в дата центрах, использующих на своей серверной инфраструктуре технологии виртуализации (Virtual Machine VM). Физическая сеть реализуется на базе классических коммутаторов/маршрутизаторов, поверх физической сети строятся виртуальные каналы типа «точка-точка» (туннелями). В качестве туннельного механизма в основном используется MAC-in-IP-туннелирование. Для организации передачи между контроллером и виртуальными коммутаторами может быть использован «южный» интерфейс с использованием протокола OpenFlow; Подход позволяет быстро, динамично, по запросу создавать произвольные топологии без внесения изменений в сетевое оборудование, прозрачные для сетевого оборудования. К недостаткам относят ручное конфигурирование, неэффективное использование физической инфраструктуры.

ПКС подход имеет преимуществ в области безопасности, особенно в части физической безопасности сетевого оборудования за счет разделение

плоскости данных и плоскости управления, а также гибкости конфигурирования политик коммутации. Использование протокола OpenFlow позволяет не только конфигурировать пересылку конкретных типов трафика через определенные точки сети, но также проверять, все ли сетевые пакеты проходят через эти конкретные точки. Применение криптографической защиты с использованием технологии SDN via Overlay позволяет построить меть с заданным уровнем защиты.

Список литературы

1. Li K., Zhou W. and others. Distinguishing DDoS attacks from flash crowds using probability metrics, in NSS 2009: Proceedings of the third international Conference on Network and System Security, IEEE, Piscataway, N. J., pp. 9-17.
2. Lu W., Traore I. An unsupervised approach for detecting DDoS attacks based on traffic-based metrics. Department of Electrical and Computer Engineering, University of Victoria.
3. Siaterlis C., Maglaris B. Detecting DDoS attacks with passive measurement based heuristics. // National Technical University of Athens, Greece, 2005.
4. Фаткиева Р.Р. Разработка метрик обнаружения атак на основе анализа сетевого трафика. Вестник Бурятского государственного университета. 2013. № 9. С. 81-86.
5. Гальцев Алексей Анатольевич. Системный анализ трафика для выявления аномальных состояний сети / Автореферат диссертации на соискание ученой степени кандидата технических наук. 05.13.01 – Системный анализ, управление и обработка информации (технические системы и связь). Самара – 2013
6. Левоневский Д.К., Пичугин Ю.А., Фаткиева Р.Р. Исследование компьютерных атак методом сингулярного спектрального разложения сетевого трафика. Труды СПИИРАН. 2013. № 3 (26). С. 101-114.
7. Фаткиева Р.Р., Левоневский Д.К. Детектирование компьютерных атак методом сингулярного спектрального разложения. Труды СПИИРАН. 2013. № 2 (25). С. 135-147.

8. Фаткиева Р.Р. Корреляционный анализ аномального сетевого трафика. Труды СПИИРАН. 2012. № 4 (23). С. 93-99.