УДК 004.042

Распределённая система защиты облачных вычислений от сетевых атак

А. А. Талалаев, И. П. Тищенко, В. П. Фраленко, В. М. Хачумов [1]

Рассматривается новая система защиты облачных вычислений от сетевых атак, основывающаяся на методах искусственного интеллекта. Описаны общая архитектура системы, входящие в её состав программные модули и взаимодействие компонентов. Представлен графический интерфейс администратора, показан процесс подготовки системы к работе.

Ключевые слова: облако, распределённая система, защита, сетевая атака, DDoS.

1. Введение

В связи с проникновением новых сетевых технологий в различные сферы деятельности человека становятся всё более актуальными вопросы защиты компьютерных сетей от постороннего вмешательства. Ведущими IT-компаниями мира в настоящее время разрабатываются различные принципы организации защиты облачных вычислений. Актуальность подобных исследований следует из современных возможностей подключения вычислительных ресурсов к сетям Интернет и предоставления пользователям облачных услуг, что, в свою очередь, создает реальную угрозу их безопасности. Распределённая система защиты облачных вычислений, реализованная в ИПС им. А.К. Айламазяна РАН в виде экспериментального образца (далее «ЭО ИПС»), обеспечивает информационную безопасность за счёт функций обнаружения и предотвращения как классических, так и распределённых сетевых атак. Система имеет высокую точность работы механизмов обнаружения вторжений за счёт применения методов искусственного интеллекта, автоматного моделирования сетевых атак на облачные вычисления, таблично-алгоритмического и нейросетевого распознавания. Различные аспекты функционирования «ЭО ИПС» были опубликованы в серии работ [1 – 8]. В настоящей статье даётся обобщение выполненных исследований.

2. Общая архитектура и взаимодействие компонентов

Общая архитектура системы представлена на рис. 1. В состав «ЭО ИПС» входят следующие программные компоненты:

1) сетевые сенсоры, осуществляющие сбор сетевых пакетов и преобразующие полученную информацию в набор признаков;

[1] Работа выполнена в рамках проекта 2.3 «Обнаружение и предотвращение распределённых сетевых атак на высокопроизводительные системы облачных вычислений на основе отечественных аппаратно-программных комплексов семейства «СКИФ» по Программе фундаментальных исследований ОНИТ РАН и НИР «Разработка программно-инструментальных средств обнаружения и распознавания сетевых атак для обеспечения информационной защиты вычислительных (компьютерных) систем (в том числе кластерных установок) для систем и средств управления космическими аппаратами НСЭН и создаваемой системы интегрального информационного обслуживания» (шифр «Телеприбор-ИСА»).

2) модули анализа, производящие нейросетевой и статистический анализ сетевого пакета на предмет наличия признаков известных типов;

3) модуль реакции, вырабатывающий реакцию на каждый определённый тип атаки;

4) модуль управления компонентами, осуществляющий взаимодействие остальных модулей, в том числе расположенных на других вычислительных узлах и виртуальных машинах;

5) модуль хранения, обеспечивающий чтение и запись настроек «ЭО ИПС», а также хранение в базе данных (БД) признаков сетевых пакетов;

6) графический интерфейс, основными задачами которого являются: отображение информации о текущем состоянии «ЭО ИПС» в целом; изменение настроек «ЭО ИПС»; управление процессом сбора признаков, получаемых из сетевых пакетов; обучение и дообучение модулей анализа; поддержка решений администратора.

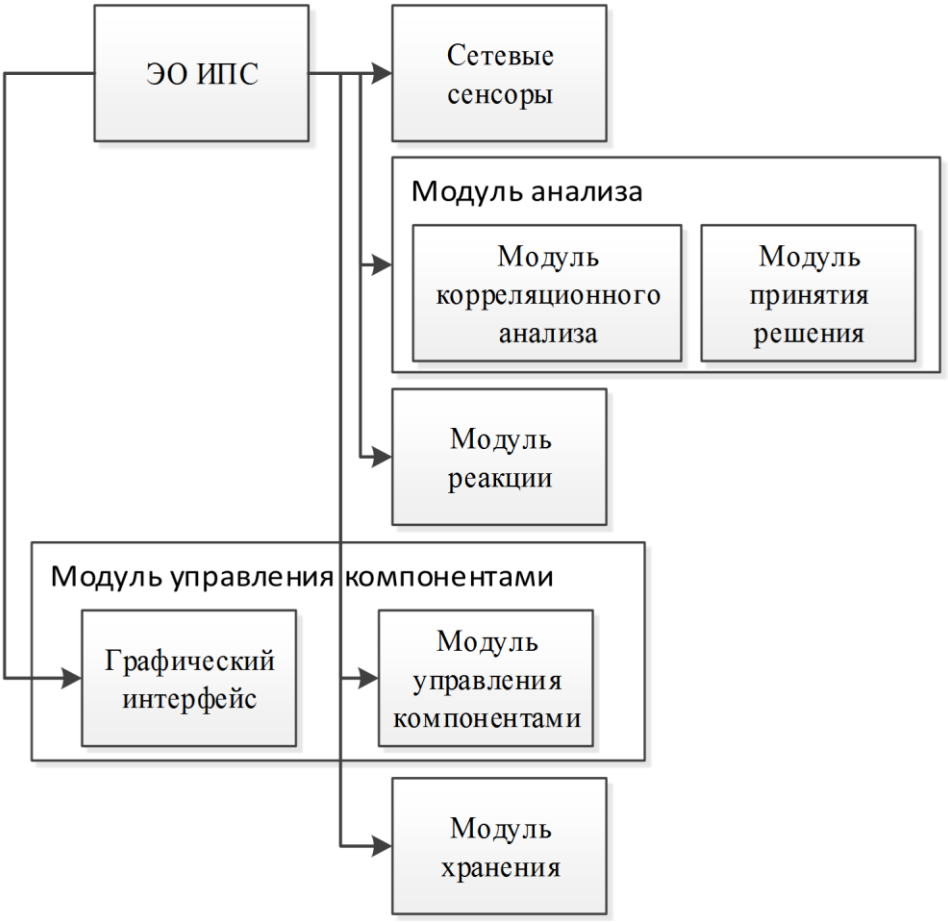

Рис. 1. Архитектура «ЭО ИПС»

Предполагается, что экземпляры «ЭО ИПС» размещаются на всех узлах распределённой вычислительной системы. Анализируя трафик и взаимодействуя друг с другом, они обеспечивают обнаружение атаки и определение её типа. Кроме того, экземпляры «ЭО ИПС» отправляют сообщения об атаках и обеспечивают визуализацию ситуации, в том числе в когнитивном графическом виде.

Взаимодействие между компонентами «ЭО ИПС» осуществляется по следующей схеме:

1) сетевые сенсоры производят захват пакетов из сети, получая при этом некоторые признаки, далее эти признаки передаются в модуль управления компонентами;

2) модуль управления компонентами производит контроль всех модулей системы: при поступлении новых признаков сетевых пакетов от сетевых сенсоров данные передаются модулю анализа, также при необходимости осуществляется запись признаков сетевых пакетов в БД;

3) модуль корреляционного анализа устанавливает значимость сетевых признаков, анализирует результаты, полученные модулями принятия решения на разных узлах распределённой системы облачных вычислений;

4) модуль принятия решения определяет принадлежность пакета к нормальному или вредоносному трафику;

5) модуль реакции в случае определения факта атаки осуществляет действия сдерживающего характера;

6) модуль хранения данных осуществляет хранение настроек системы и поступающих признаков сетевых атак (рис. 2).

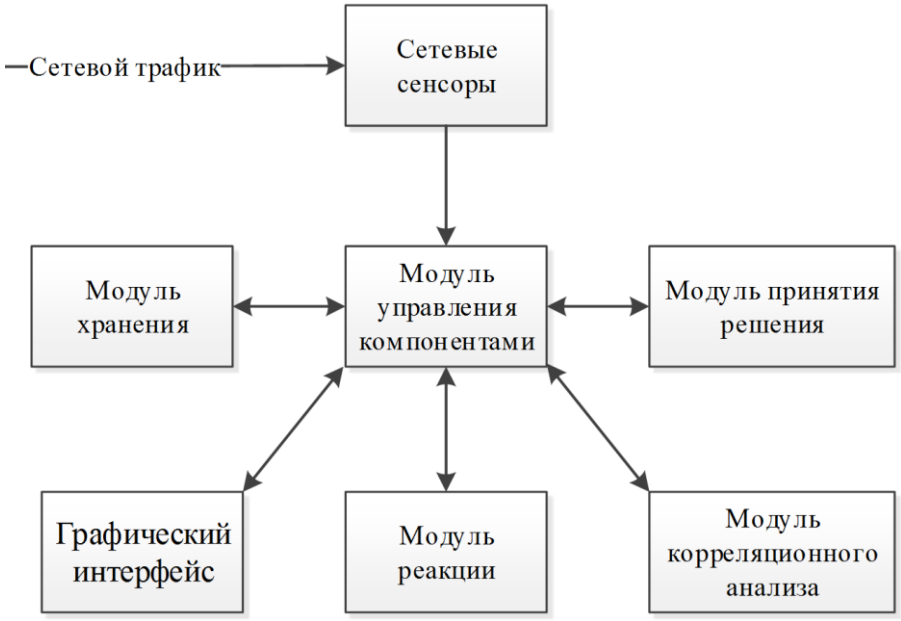

Рис. 2. Обобщённое взаимодействие компонент «ЭО ИПС»

Описанное взаимодействие модулей затрагивает работу только одного узла вычислительной системы. В действительности экземпляры «ЭО ИПС» для учёта специфики облачных вычислений располагаются на всех узлах (при этом почти все его компоненты дублируются) (рис. 3).

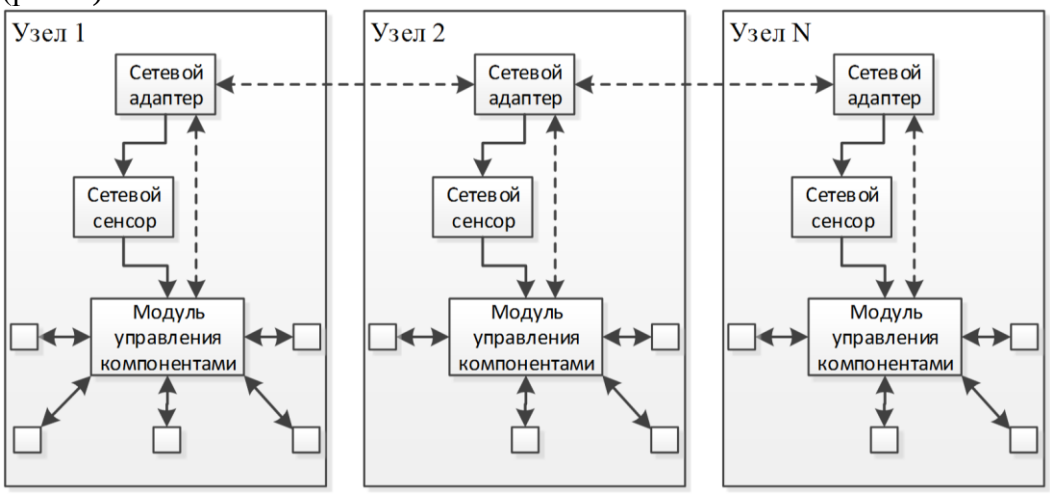

Рис. 3. Взаимодействие компонент между узлами

Пунктиром на рис. 3 отмечены управляющие связи между различными копиями (экземплярами) системы. Связь между компонентами, находящимися на разных узлах, должна осуществляться через модули управления компонентами. Общий алгоритм работы осуществляется согласно схеме, отображённой на рис. 4. Рассмотрим далее программные компоненты «ЭО ИПС» более подробно.

3. Сетевые сенсоры

Программные сетевые сенсоры – модули, которые устанавливаются по одному на каждую сетевую шину защищаемого объекта и предназначаются для выделения информативных признаков, характеризующих пакеты данных, циркулирующих в сети, и состояние защищаемой системы. Анализ выделяемых значений позволяет определить факт несанкционированного доступа к системе либо попытки нарушения её работоспособности.

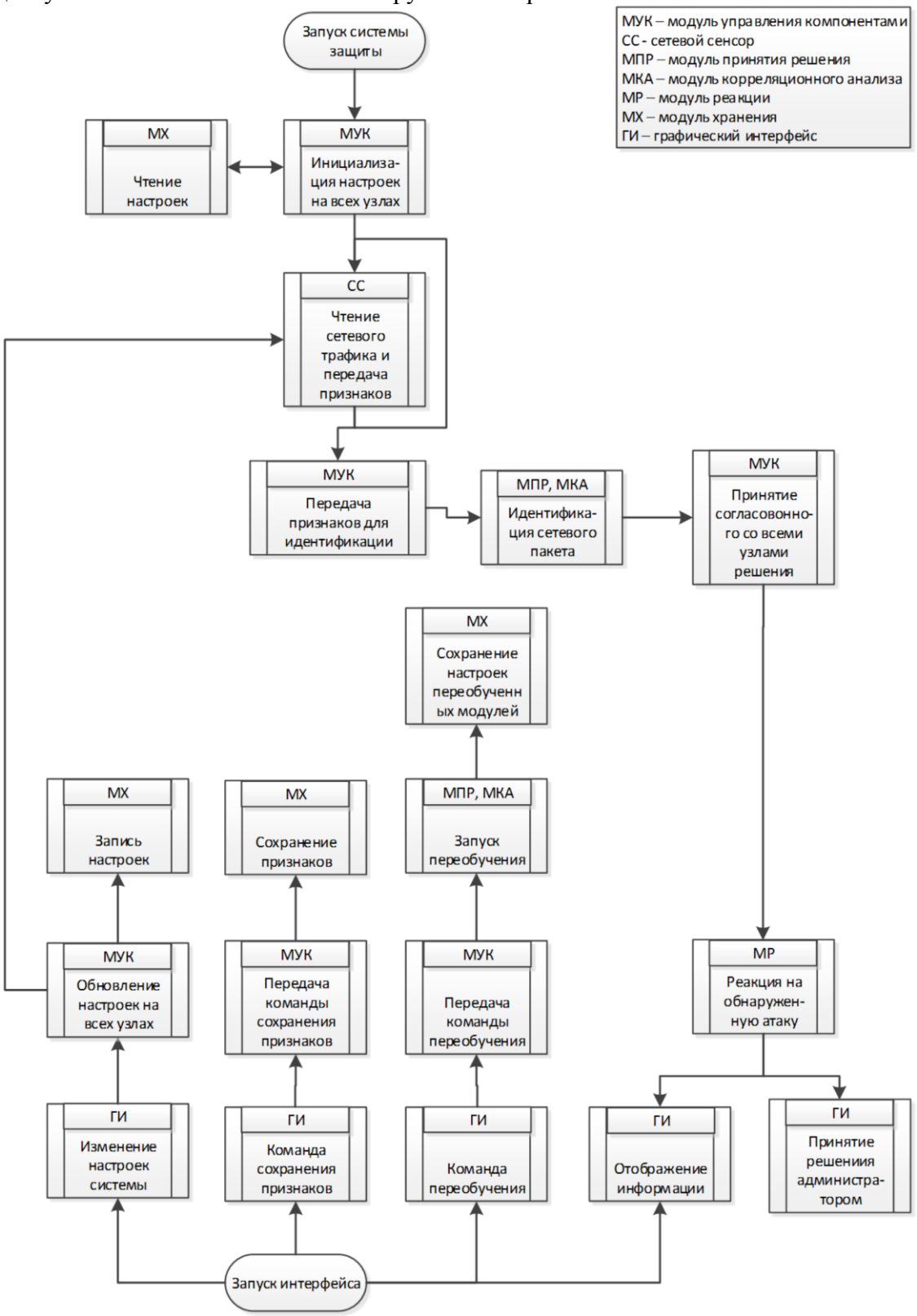

Рис. 4. Алгоритм взаимодействия компонент

Программная реализация модуля выделения информативных признаков представляет собой набор классов, функционирующих на основе библиотеки PCAP [9]. Объектная структура модуля представлена на рис. 5.

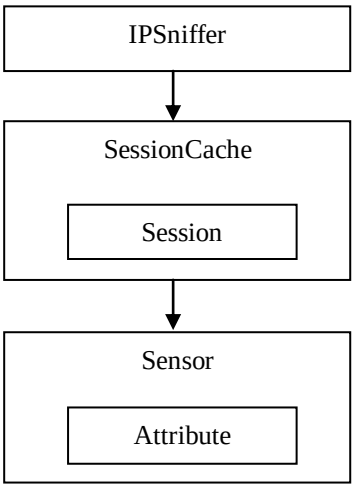

Рис. 5. Объектная структура модуля выделения информативных признаков

Класс *IPSniffer* инкапсулирует функции работы с потоком сетевых пакетов посредством вызова функции библиотеки PCAP, организуя их перехват с указанного сетевого интерфейса. При этом определяется корректность структуры полученного сетевого пакета и выделяются его заголовки, после чего перехваченный пакет передаётся на дальнейшую обработку. Производится перехват сетевых пакетов TCP- и UDP-протоколов.

Экземпляр класса *SessionCache* хранит информацию об установленных сессиях сетевого обмена и предоставляет сервисные функции для определения сессии, к которой относится новый полученный пакет сетевого трафика. Данный класс оперирует объектами типа *Session* (сессия). Каждая сессия определяется четвёркой <IP отправителя, Порт отправителя, IP получателя, Порт получателя>; кроме того, в полях данной структуры хранится информация, необходимая для выделения значений некоторых из признаков.

Класс *Sensor* реализует вспомогательный управляющий механизм для унификации подхода к обработке потока сетевого трафика. Оперируя выделенными на предыдущих этапах структурами сессии и сетевого пакета, производит вызов функций выделения значений информативных признаков, определённых в классе *Attribute*, с использованием механизма полиморфизма. Каждый из классов-наследников *Attribute* реализует поддержку программного интерфейса, инкапсулируя в себе программную логику выделения значений одного или группы признаков, характеризующих сетевой поток. С использованием интерфейса данного класса производится получение текущего набора значений информативных признаков, передаваемых другим модулям системы.

Множество выделяемых информативных признаков определяется структурой базы сетевых атак KDD-99 [10] и двумя дополнительными группами признаков, выделяемых на основе анализа структуры запросов, передаваемых по протоколу HTTP [11], и анализа SNMP-пакетов [12]. Таким образом, все выделяемые информативные признаки могут быть разделены на шесть групп:

1) признаки, информация о которых может быть выделена из заголовка сетевого пакета (1 – 9);

2) признаки, выделяемые из информационной части пакета на основании экспертных знаний либо как информация о состоянии контролируемого узла (10 – 22);

3) признаки, значения которых вычисляются как статистика за прошедшие две секунды сетевой активности, для протокола TCP (23 – 31);

4) признаки, значения которых вычисляются как статистика за последние 100 соединений (32 – 41);

5) признаки, выделяемые на основе анализа HTTP-запросов (42 – 55);

6) признаки, значения которых вычисляются как статистика за прошедшие две секунды сетевой активности, для протокола SNMP (56 – 60).

4. Модули анализа

Модуль анализа «ЭО ИПС» включает модуль корреляционного анализа и модуль принятия решения, построенные на основе методов искусственного интеллекта. Модуль анализа, предназначенный для обработки собранных сенсорами данных с целью обнаружения атак и вторжений, в своей работе использует информацию программных сетевых сенсоров и базы знаний системы.

Модуль корреляционного анализа осуществляет отбор независимых признаков по алгоритму, основанному на ранжировании переменных (табл. 1), исходя из значений соответствующих коэффициентов парной корреляции между отдельными переменными.

Таблица 1. Ранжирование переменных по значению величины коэффициента парной корреляции $R_{i,j}$

Значение коэффициента парной корреляции	Ранг
$R_{i,j} > 0.90$	1.25
$0.90 < R_{i,j} < 0.85$	1.00
$0.85 < R_{i,j} < 0.80$	0.75
$0.80 < R_{i,j} < 0.75$	0.50
$0.75 < R_{i,j} < 0.70$	0.25

Суммарный ранг переменной x_i определяется в соответствии с выражением $\mathrm{RANG}_i = \sum_{j=1}^{n-1} R_{i,j}$. Количество переменных n, допускаемых для идентификации, определяется составом перечисленных шести групп признаков. При этом из эксперимента исключаются переменные с $R_{i,j} > 0.7$. Для известных классов атак были выбраны номера независимых признаков и затем отсортированы по информативности с помощью алгоритмов Add и Dell [13] и расстояния Евклида – Махаланобиса [14].

Принятие решения о выявлении атаки или вторжения осуществляется на основе применения ряда методов к данным, полученным от программных сетевых сенсоров, в том числе:

а) метода выявления вторжений, основанного на методах искусственного интеллекта и использовании базы знаний, выполняющего обнаружение распределённых атак;

б) метода обнаружения аномалий, выполняющего выявление отклонений в работе системы, основывающегося на профилировании пользователей;

в) метода обнаружения атак отказа в обслуживании.

Модуль принятия решения имеет два уровня классификации для выявления аномальной сетевой активности. На первом уровне находится классификатор на основе генетически обучаемого конечного автомата, разделяющего сетевые записи на два класса: «норму» и «подозрение на атаку» [15].

На втором уровне сетевые записи, отнесённые ко второму классу, далее обрабатываются комитетом классификаторов на основе метода опорных векторов [16] и нейронной сети прямого распространения, обучаемой по методу Левенберга – Марквардта [17].

Каждый классификатор обучается распознавать не только «норму», но и классы атак. Необходимость в распознавании «нормы» на втором уровне возникает из-за недопустимости ложных срабатываний защитных механизмов на этом виде сетевой активности. Комитетом производится подсчёт количества голосов, поданных за тот или иной вид сетевой активности. Побеждает класс, набравший большее количество голосов. В случае если количество голосов за «норму» и какой-нибудь из классов атак совпадает, то сетевая запись относится к «норме». Если количество голосов совпадает для разных классов атак, то сетевая запись относится к классу неизвестных/смешанных атак.

Расширенная схема взаимодействия, включающая «Модуль корреляционного анализа» и «Модуль принятия решения», представлена на рис. 6.

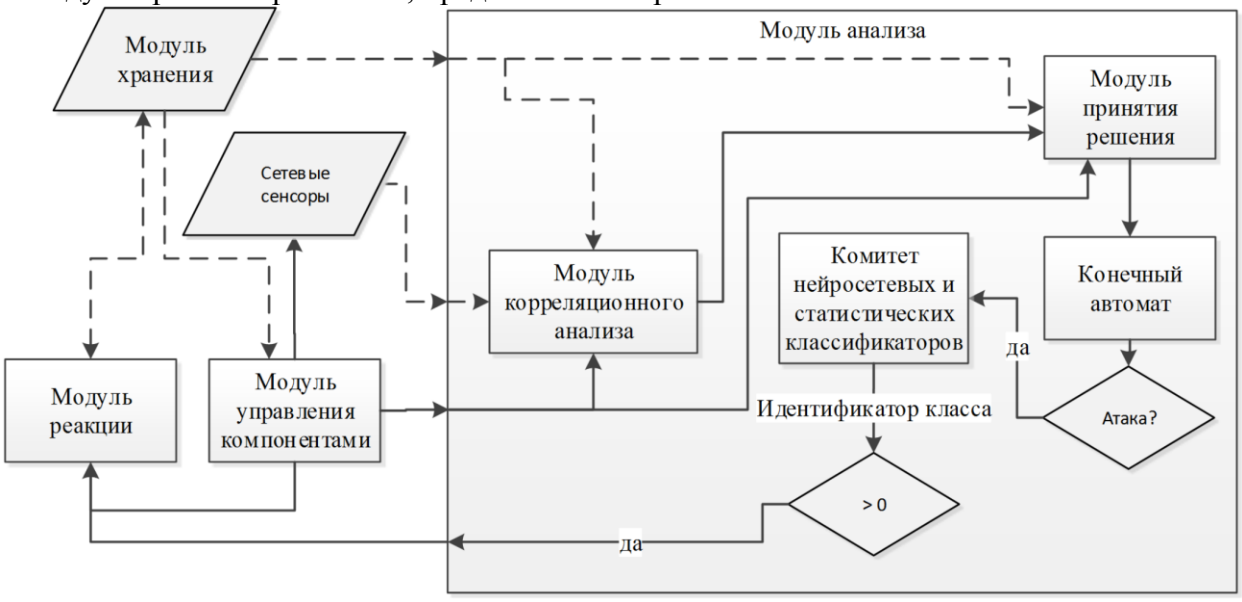

Рис. 6. Схема взаимодействия модуля корреляционного анализа и модуля принятия решения

5. Модуль реакции

Модуль реакции состоит из класса *Reaction*, содержащего следующий набор методов:

- *CreateSet (set_name)* – создает множество *ipset* типа *ipset:hash* с именем *set_name*;
- *DeleteSet (set_name)* – удаление списка с именем *set_name*;
- *ClearSet (set_name)* – очистка списка с именем *set_name*;
- *DeleteIPFromSet (set_name, ip)* – удаляет из множества с именем *set_name* адрес *ip*;
- *DeleteReaction (react_name)* – удаляет реакцию с именем *react_name*;
- *HttpFlood* – реакция на атаку типа HTTP Flood;
- *SynFlood* – реакция на атаку типа SYN Flood;
- *UdpFlood* – реакция на атаку типа UDP flood;
- *IcmpFlood* – реакция на атаку типа ICMP flood;
- *TcpFlood* – реакция на атаку типа TCP flood;
- *NMap* – реакция на атаку типа NMap (сканирование портов);
- *BanIP (set_name, ip)* – добавляет IP-адрес в множество с именем *set_name*;
- *LimitPackets* – функция ограничения обрабатываемых пакетов в единицу времени;
- *LimitConnections* – функция ограничения новых соединений в единицу времени.

В «ЭО ИПС» задействованы перечисленные в табл. 2 реакции на сетевые атаки.

Таблица 2. Команды iptables для борьбы с различными типами атак

Тип атаки	Реакция
HTTP Flood	ipset -A blacklist xx.xx.xx.xx [timeout XX*60]
SYN Flood	echo "1" > /proc/sys/net/ipv4/tcp_syncookies
ICMP Flood	iptables -A INPUT -p icmp -m limit --limit 1/s --limit-burst 1 –j ACCEPT iptables -A INPUT -p icmp -j DROP iptables -A OUTPUT -p icmp -j ACCEPT iptables -A INPUT -p icmp -m conntrack –ctstate ESTABLISHED,RELATED -j ACCEPT iptables -A INPUT -p icmp -m conntrack --ctstate NEW -m limit --limit 32/sec --limit-burst 32 -j ACCEPT iptables -P INPUT DROP
UDP Flood	iptables -N thyl-udp-flood iptables -A INPUT -p udp -j thyl-udp-flood iptables -A thyl-udp-flood -m limit --limit 10/s --limit-burst 20 -m comment --comment "Limit UDP rate" -j RETURN iptables -A INPUT -p udp -m conntrack –ctstate ESTABLISHED,RELATED -j ACCEPT iptables -A INPUT -p udp -m conntrack --ctstate NEW -m limit --limit 32/sec --limit-burst 32 -j ACCEPT
TCP Flood	ipset --add blacklist xx.xx.xx.xx [timeout XX*60]
Nmap	iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 1/s -j ACCEPT

6. Модуль управления компонентами

Система защиты предназначена для распределённых систем, поэтому модуль управления компонентами учитывает это обстоятельство в работе и обеспечивает централизованное управление. Синхронизация и поддержка нескольких точек «входа» необходимы на случай выхода части узлов облачной системы из строя. Выход из строя одного из компонент облака, не влияющего на функциональность облака в целом, не должен повлиять на работоспособность системы защиты. Для реализации модуля используется клиент-серверная технология. Модуль управления компонентами отвечает за запуск и правильное взаимодействие модулей системы. Рассмотрим его основные задачи.

Задача 1: Инициализация

После осуществления запуска системы защиты на конкретном узле модуль производит ряд действий для обеспечения правильной дальнейшей работы. Проверяется наличие необходимых конфигурационных файлов. Производится загрузка настроек из конфигурационных файлов, происходит инициализация модулей базы данных, сенсоров, модулей корреляции и распознавания.

Задача 2: Обеспечение взаимодействия модулей

Модуль определяет порядок и характер взаимодействий остальных модулей в различных ситуациях, таких как обнаружение атаки или переопределение параметров системы администратором.

***Задача 3: Обеспечение сетевых обменов между экземплярами* «ЭО ИПС»**

Модуль обеспечивает передачу данных и команд между модулями, находящимися на одном узле или на разных.

Задача 4: Запуск/остановка работы

Модуль обеспечивает запуск и остановку системы, загрузку/выгрузку других модулей, закрытие установленных программой соединений.

Задача 5: Выполнение задач синхронизации

Модуль обеспечивает работу процедур обновления базы данных, конфигурационных файлов, базы автоматов, их синхронизацию между узлами.

Задача 6: Предоставление графического интерфейса администратора

Модуль отвечает за предоставление графического интерфейса администратора, предназначенного для корректирования конфигурации системы защиты, контроля и управления элементами распределённой системы. Интерфейс предоставляет возможность редактирования конфигурационных файлов и базы данных системы, включает в себя функцию запуска управляемой синхронизации, включает механизмы управления ведением логов и дампов баз данных и дампов информации с сенсоров.

7. Модуль хранения

Модуль хранения, основанный на системе управления базами данных, обеспечивает:

1) хранение настроек системы;

2) хранение событий в системе;

3) организацию базы знаний системы.

Для организации распределённой работы с несколькими БД и разрешения конфликта версий записей используется схема «сеть» [18], как это показано на рис. 7. С помощью такой схемы синхронизируются состояния двух БД, далее изменения распространяются по всей «сети».

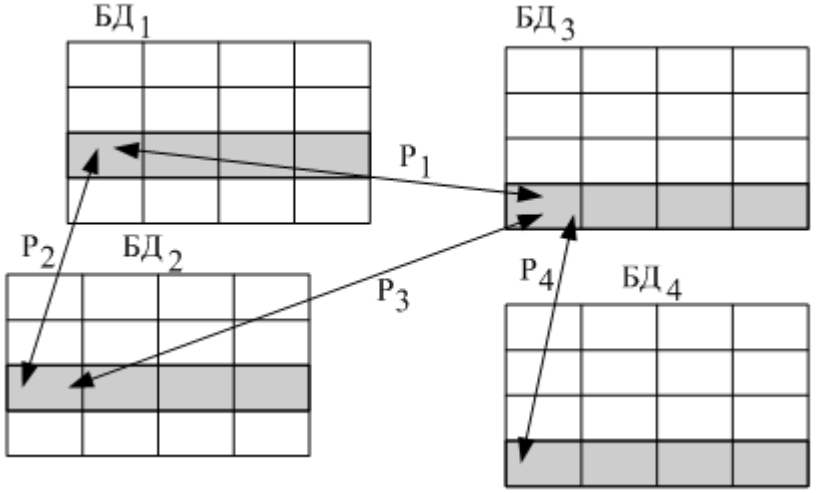

Рис. 7. Взаимодействие нескольких БД в схеме «сеть»

Для реализации механизма репликации между базами таблицы, кроме неслужебных данных, не участвующих в процессе репликации, модуль использует служебную информацию:

1) об идентификаторе БД;

2) об идентификаторе записи (0 – с записью ничего не произошло, 1 – запись внесена, 2 – запись изменена, 3 – запись помечена на удаление);

3) о времени внесения записи в таблицу.

8. Графический интерфейс

Графический интерфейс системы служит для визуализации обнаруженных атак и вторжений, для управления функционированием интеллектуальной программной системы обнаружения и предотвращения распределённых сетевых атак.

Сразу после старта графического интерфейса появляется форма входа на сервер (рис. 8). Процесс входа состоит из двух этапов. На первом этапе происходит соединение с сервером.

Имя сервера и порт вводятся в полях с соответствующими названиями в верхней части формы. После установки соединения становятся активными поля ввода логина и пароля в нижней части формы. Для подключения к другому серверу необходимо нажать кнопку «Отмена», при этом произойдёт отключение от текущего сервера и поля ввода информации о сервере вновь станут активными. Для входа на сервер необходимо нажать кнопку «Войти», после чего произойдёт проверка введённых данных, и в случае успеха форма входа на сервер исчезнет.

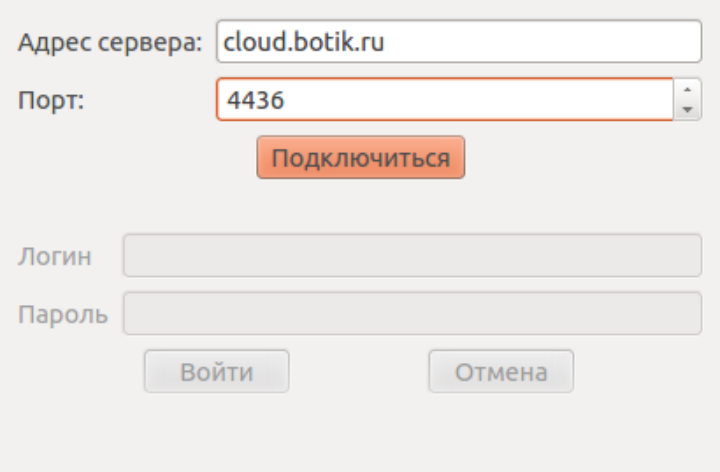

Рис. 8. Форма входа на сервер

На рис. 9 изображён общий вид интерфейса администратора. В левой части интерфейса (область А) визуализирована структура облачной системы. Кнопка «Обновить» служит для получения актуальной структуры. Область Б содержит набор разделов, каждый из которых соответствует определённому модулю системы. Структурная схема разделов показана на рис. 10. Некоторые из разделов содержат подразделы (область В). Область Г представляет собой рабочую область интерфейса. В нижней части интерфейса расположена строка статуса (область Д), в которой отображаются информационные сообщения интерфейса.

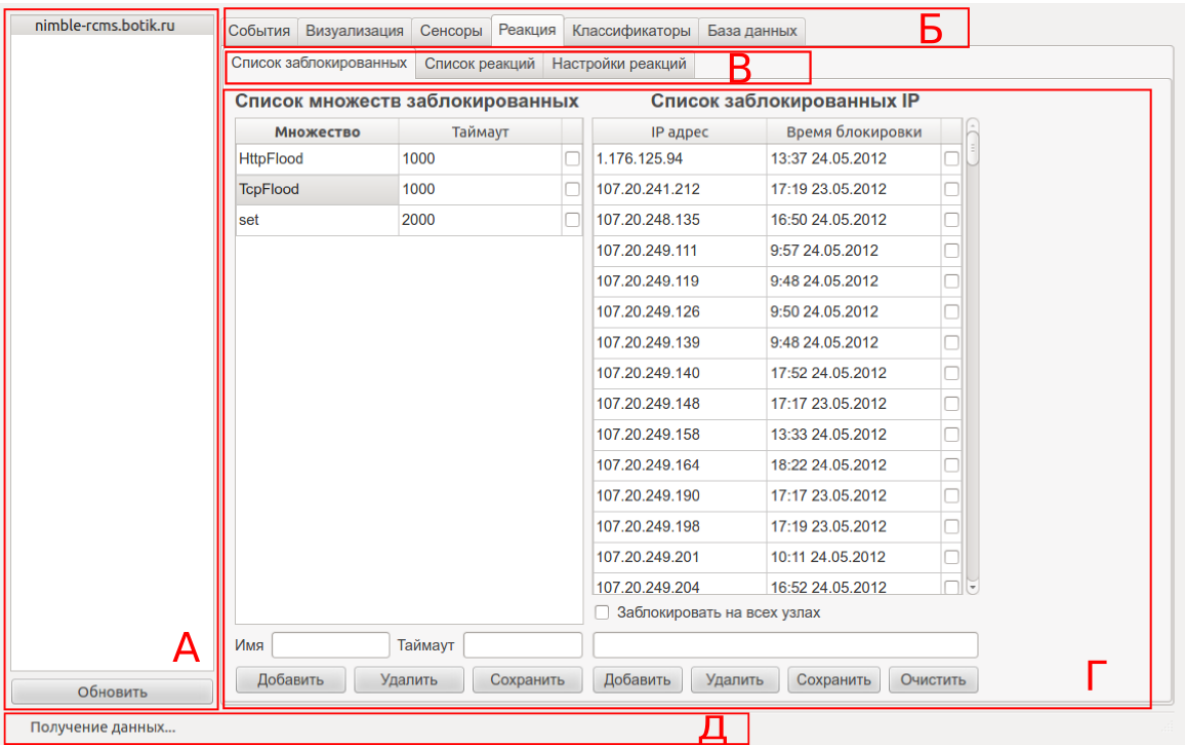

Рис. 9. Общий вид интерфейса системы

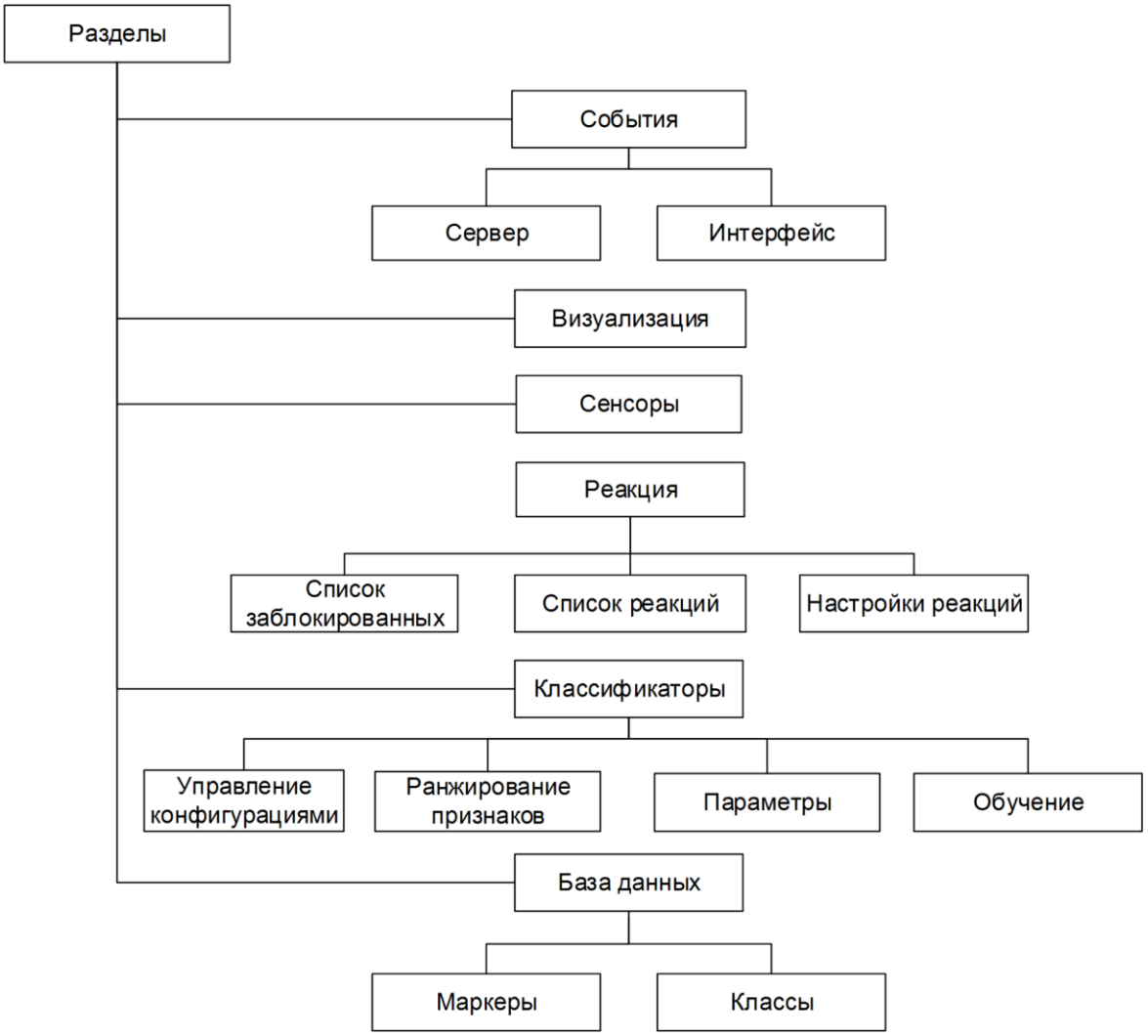

Рис. 10. Структурная схема разделов интерфейса системы

Некоторые из форм интерфейса показаны в следующем разделе.

9. Подготовка «ЭО ИПС» к работе

Рассмотрим пример обучения «Модуля анализа» на заранее созданной конфигурации *“new_configuration”*. Обучение состоит из двух шагов: ранжирования признаков по информативности и обучения комитета классификаторов.

Для ранжирования признаков по информативности необходимо выполнить следующие шаги:

а) перейти в подраздел «Ранжирование признаков»,

б) выбрать классы сетевой активности, на основании обучающих выборок которых и будет выбран список наиболее информативных признаков, перечисленных в порядке убывания значения этого показателя, нажать кнопку «Ранжировать» (рис. 11).

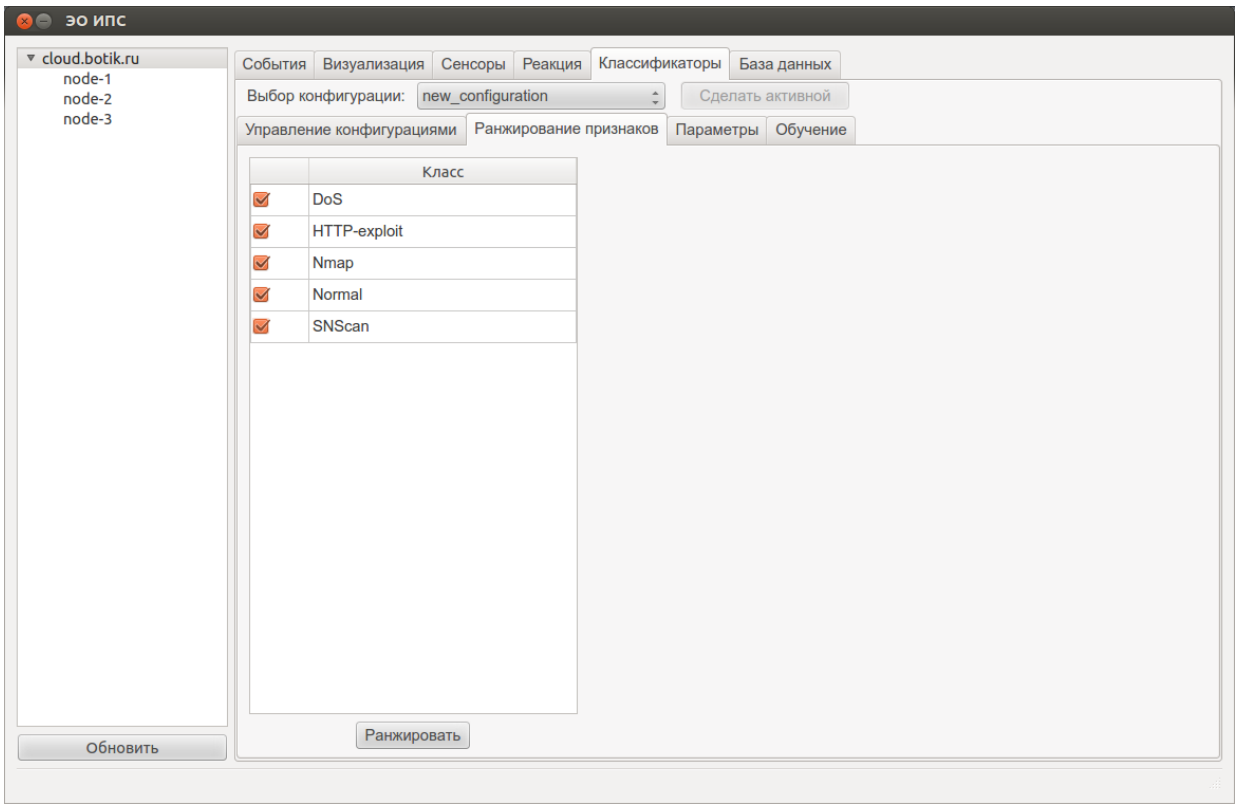

Рис. 11. Ранжирование признаков

По окончании описанной операции следует перейти в подраздел «Параметры» для указания желаемых настроек классификаторов, входящих в комитет (рис. 12).

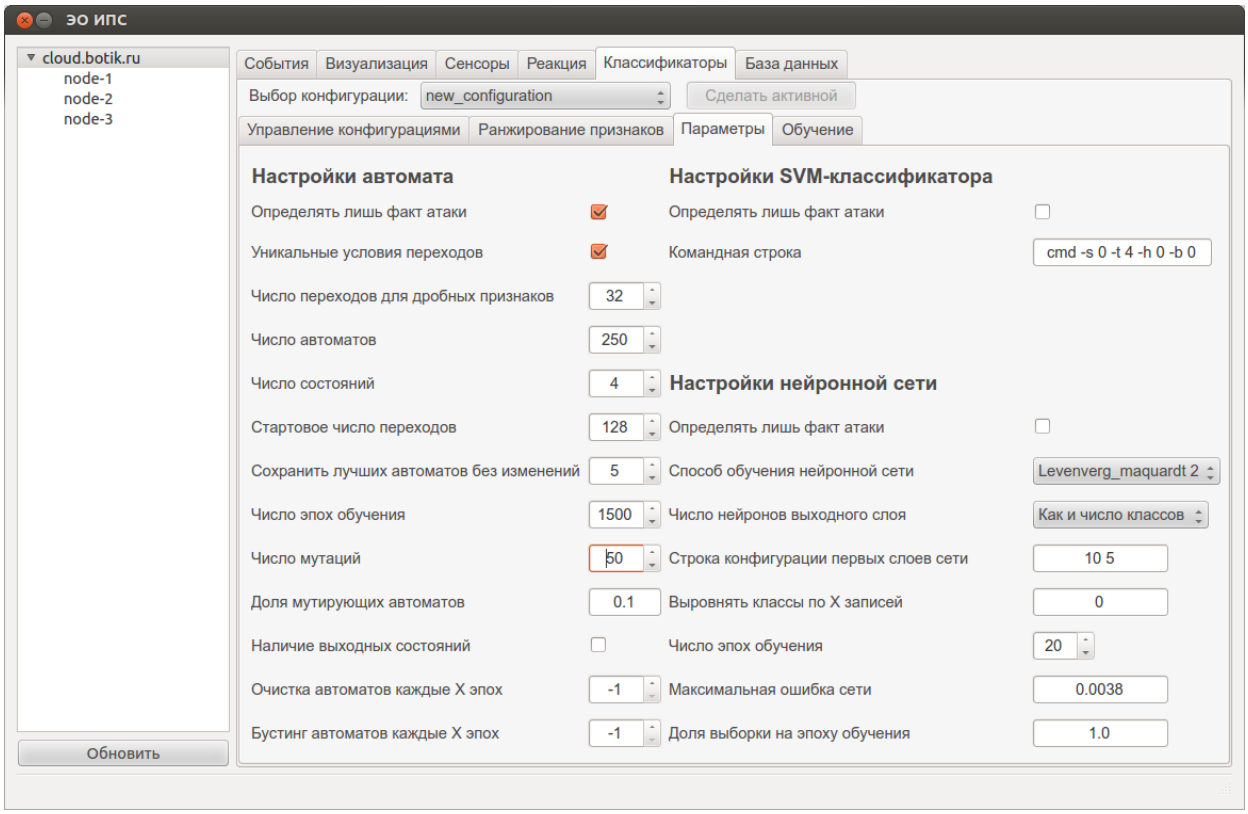

Рис. 12. Установка настроек классификаторов

Далее необходимо обучить все классификаторы, входящие в комитет. В частности, классификаторы «Автомат», «SVM-классификатор» и «Нейронная сеть». Соответствующие инструменты доступны в подразделе «Обучение» (рис. 13).

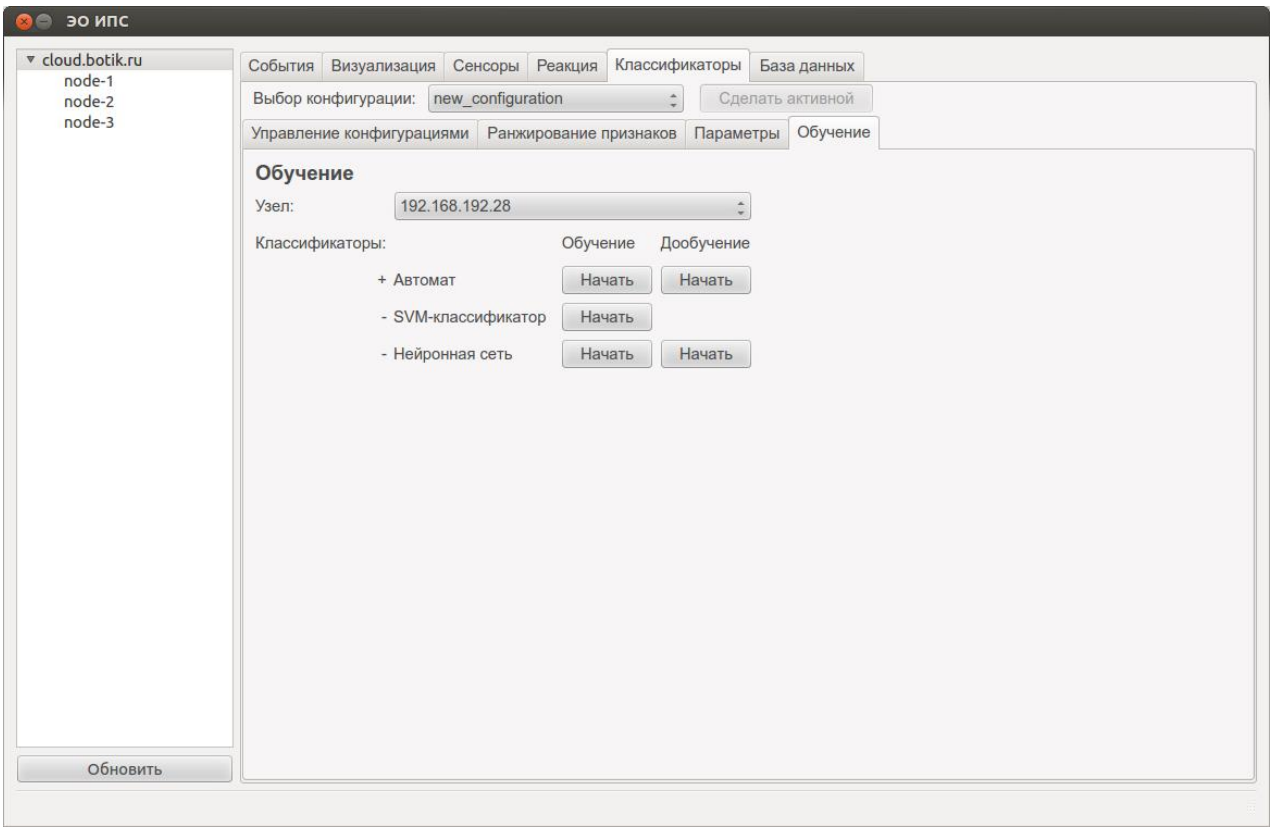

Рис. 13. Подраздел «Обучение»

Предварительно следует выбрать узел, на котором мы собираемся произвести обучение. Рекомендуется выбрать наиболее быстрый и менее загруженный из доступных узлов стенда. После этого необходимо произвести обучение. Рассмотрим этот процесс на примере «SVM-классификатора». После нажатия кнопки «Начать» (напротив строчки «- SVM-классификатор», в колонке «Обучение») запустится процесс обучения указанного классификатора, что вызовет соответствующие изменения в интерфейсе – см. рис. 14 (текст на кнопке сменился с «Начать» на «Остановить»).

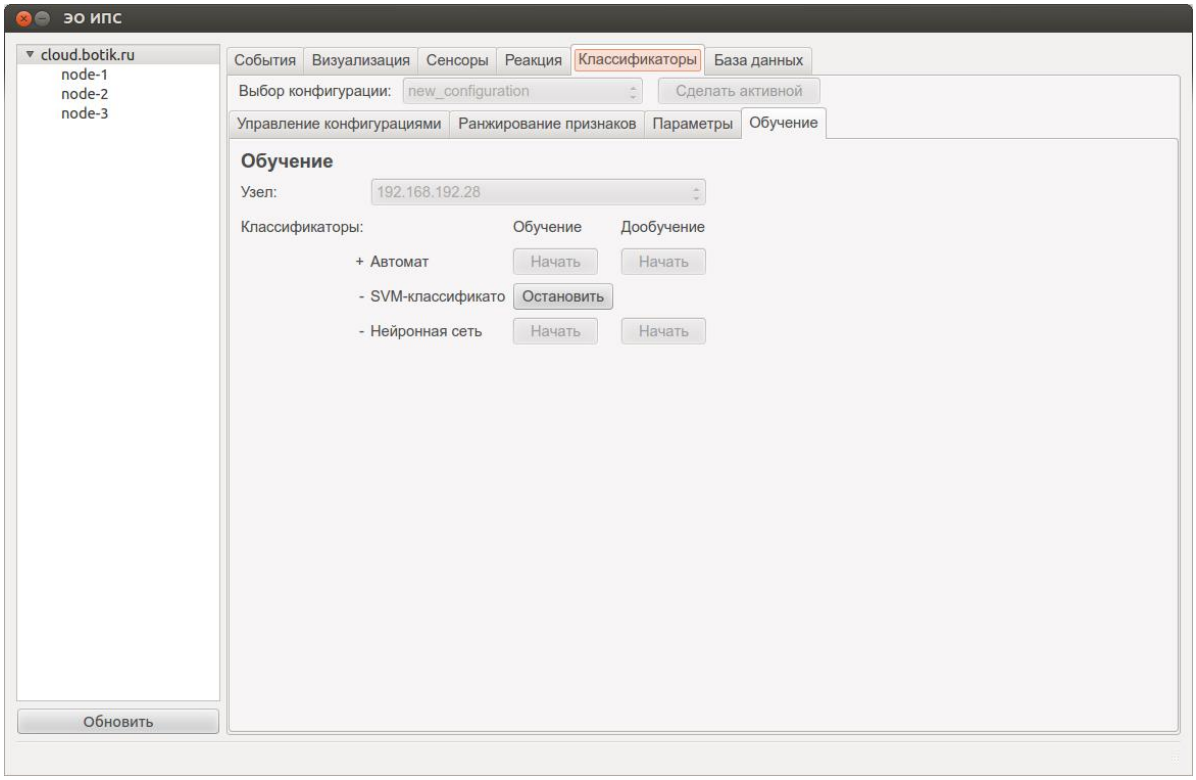

Рис. 14. Подраздел «Обучение» после запуска обучения SVM-классификатора

После завершения процесса обучения форма изменит свой вид на начальный (рис. 13).

Для классификаторов «Автомат» и «Нейронная сеть» доступны кнопки дообучения (кнопки с текстом «Начать» в колонке «Дообучить»). Алгоритм их использования аналогичен алгоритму обучения. SVM-классификатор, в отличие от остальных, не обладает такой функцией.

10. Экспериментальные исследования

Механизм обнаружения сетевых атак при помощи комитета классификаторов характеризуется более высокими показателями распознавания по сравнению с отдельными нейросетевыми классификаторами (табл. 3). Используемые классификаторы предоставляют возможность дообучения, что позволяет корректировать их работу в случае появления новых типов атак и существенно сокращает время подготовки классификатора. В табл. 3 представлены скоростные характеристики отдельных классификаторов в составе «ЭО ИПС».

Таблица 3. Примерные скорости распознавания различными классификаторами

Тип классификатора	Распознавание, прец./сек.
Автоматный классификатор	700000
Сеть прямого распространения, Левенберг – Марквардт	53000
Полиномиальное расстояние Евклида – Махаланобиса	192000
Метод опорных векторов	59000

Качество распознавания каждого класса сетевой активности X оценивалось по двум параметрам:

а) точность – число правильных определений записей класса X делится на число фактов указания на класс X;

б) полнота – число правильных определений записей класса X делится на реальное число записей класса X.

Экспериментальные исследования с использованием лабораторного стенда показали высокий уровень обнаружения атак на основе предложенных классификаторов (свыше 99% по полноте и точности обнаружения факта атаки распознающим автоматом, близкое к нулю число ошибок первого и второго рода у комитета классификаторов) (табл. 4).

Таблица 4. Сравнение результатов распознавания различными классификаторами

	Результаты распознавания атак, точность/полнота (%)				
	Норма	DoS	Nmap	HTTP-exploit	SNScan
ЭО ИПС	100/100	100/100	100/99.8	100/100	100/99.4
IDS Snort	100/99.77	100/0.002	100/0.075	100/5	100/85.93
IDS Bro	100/99.97	100/40.68	100/0.006	94.91/100	100/0.05

Система защиты за счёт использования нейросетевого классификатора показала способности к распознаванию модифицированных сетевых атак и обнаружению закономерностей и аномалий в потоках данных.

11. Заключение

В результате выполненных теоретических исследований и испытаний экспериментального образца системы обнаружения и предупреждения сетевых атак на системы облачных вычислений «ЭО ИПС» были получены следующие результаты:

а) разработан новый класс алгоритмов обнаружения сетевых атак с помощью генетически обучаемых конечных автоматов, которые характеризуются приемлемым временем, затрачиваемым на обучение, отсутствием необходимости в экспертных знаниях, высокой скоростью обнаружения сетевых атак, возможностью дообучения на выборках;

б) разработан и исследован комитет классификаторов на основе: метода опорных векторов, нейронной сети прямого распространения, обучаемой по методу Левенберга – Марквардта, корреляционного анализатора.

В результате испытаний системы защиты многопроцессорных компьютерных систем были достигнуты следующие технические характеристики:

а) достаточно высокий уровень обнаружения атак на основе предложенных классификаторов (свыше 99% по полноте и точности обнаружения факта атаки распознающим автоматом, свыше 94% полноты и точности распознавания типа атаки нейронной сетью);

б) возможность гибкой настройки автоматов (за счёт генетических алгоритмов) и дообучения нейросетевых классификаторов с учётом новых сетевых атак;

в) способность к распознаванию модифицированных сетевых атак и обнаружению закономерностей и аномалий в потоках данных.

Литература

1. *Талалаев А. А., Тищенко И. П., Фраленко В. П., Хачумов В. М.* Эксперименты по нейросетевому мониторингу и распознаванию сетевых атак // Информационное противодействие угрозам терроризма, № 15, 2010, с. 81-86.
2. *Талалаев А. А., Тищенко И. П., Фраленко В. П., Хачумов В. М.* Разработка нейросетевого модуля мониторинга аномальной сетевой активности // Нейрокомпьютеры: разработка и применение, № 7, 2011, с. 32-38.
3. *Емельянова Ю. Г., Талалаев А. А., Тищенко И. П., Фраленко В. П.* Нейросетевая технология обнаружения сетевых атак на информационные ресурсы // Программные системы: теория и приложения: электрон. научн. журн. 2011. № 3(7), с. 3-15 [Электронный ресурс]. URL: `http://psta.psiras.ru/read/psta2011_3_3-15.pdf` (дата обращения: 27.12.2012).
4. *Емельянова Ю. Г., Фраленко В. П.* Анализ проблем и перспективы создания интеллектуальной системы обнаружения и предотвращения сетевых атак на облачные вычисления // Программные системы: теория и приложения: электрон. научн. журн. 2011. № 4(8), с. 17-31 [Электронный ресурс]. URL: `http://psta.psiras.ru/read/psta2011_4_17-31.pdf` (дата обращения: 27.12.2012).
5. *Кондратьев А. А., Тищенко И. П., Фраленко В. П.* Разработка распределённой системы защиты облачных вычислений // Программные системы: теория и приложения: электрон. научн. журн. 2011. № 4(8), с. 61-70 [Электронный ресурс]. URL: `http://psta.psiras.ru/read/psta2011_4_61-70.pdf` (дата обращения: 27.12.2012).
6. *Попов Г. Ю., Кондратьев А. А., Тищенко И. П., Фраленко В. П.* Разработка информационной системы поддержки коллективной разработки проектов // Программные системы: теория и приложения : электрон. научн. журн. 2012. Т. 3, № 2(11), с. 3-22 [Электронный ресурс]. URL: `http://psta.psiras.ru/read/psta2012_2_3-22.pdf` (дата обращения: 27.12.2012).
7. *Емельянова Ю. Г., Мбайкоджи Э., Соченков И. В.* Современный уровень и тенденции развития средств обеспечения сетевой безопасности систем облачных вычислений // Вестник Российского университета дружбы народов. Серия Математика. Информатика. Физика. М.: РУДН, №2, 2012, с. 116-126.

8. *Емельянова Ю. Г., Талалаев А. А.* Сетевые модели функционирования прикладных параллельных систем обработки потоков данных // Авиакосмическое приборостроение, №5, 2012, с. 10-19.
9. TCPDUMP/LIBPCAP public repository [Электронный ресурс]. URL: http://www.tcpdump.org/ (дата обращения: 27.12.2012).
10. Fifth ACM SIGKDD International Conference on Knowledge Discovery & Data Mining [Электронный ресурс]. URL: http://www.sigkdd.org/kdd1999/ (дата обращения: 27.12.2012).
11. *Danforth M.* Models for Threat Assessment in Networks [Электронный ресурс]. URL: http://www.cs.ucdavis.edu/research/tech-reports/2006/CSE-2006-13.pdf (дата обращения: 27.12.2012).
12. SNMP протокол – принципы, безопасность, применение [Электронный ресурс]. URL: http://www.codenet.ru/webmast/snmp/ (дата обращения: 27.12.2012).
13. *Загоруйко Н. Г.* Прикладные методы анализа данных и знаний. Новосибирск: ИМ СО РАН, 1999. – 270 с.
14. *Амелькин С. А., Хачумов В. М.* Обобщённое расстояние Евклида – Махаланобиса и его применение в задачах распознавания образов // Труды 12-ой Всероссийской конференции «Математические методы распознавания образов» (20 – 26 ноября 2005 года). М.: 2005, с. 7-9.
15. *Фраленко В. П.* Обнаружение сетевых атак с помощью генетически создаваемых конечных автоматов // Вестник Российского университета дружбы народов. Серия Математика. Информатика. Физика. М.: РУДН, № 4, 2012, с. 68-74.
16. *Corinna C., Vapnik V.* Machine Learning, vol. 20, num. 3, 1995. P. 273-297.
17. *Levenberg K.* A Method for the Solution of Certain Problems in Least Squares, Quart. Appl. Math, vol. 2, 1944. P. 164-168.
18. *Сатаров В. В.* Способ разрешения конфликта версий записей при асинхронной репликации БД [Электронный ресурс] // RSDN-Magazine: [сайт]. [2009]. URL: http://www.rsdn.ru/article/db/versionclash.xml (дата обращения: 27.12.2012).

Статья поступила в редакцию 09.01.2013;
переработанный вариант — 28.02.2013

Талалаев Александр Анатольевич
к.т.н., младший научный сотрудник лаборатории интеллектуального управления исследовательского центра мультипроцессорных систем ИПС им. А.К. Айламазяна РАН (152020, Переславль-Залесский, ул. Комитетская, д. 29) тел. (961) 974-0718, e-mail: arts@arts.botik.ru.

Тищенко Игорь Петрович
к.т.н., старший научный сотрудник лаборатории интеллектуального управления исследовательского центра мультипроцессорных систем ИПС им. А.К. Айламазяна РАН (152021, Переславль-Залесский, ул. Ростовская, д. 20, кв. 14) тел. (910) 813-6427, e-mail: billy@billy.botik.ru.

Фраленко Виталий Петрович
к.т.н., научный сотрудник лаборатории интеллектуального управления исследовательского центра мультипроцессорных систем ИПС им. А.К. Айламазяна РАН (152025, Переславль-Залесский, ул. Октябрьская, д. 39, кв. 41) тел. (910) 813-6427, e-mail: alarmod@pereslavl.ru.

Хачумов Вячеслав Михайлович
д.т.н., профессор, заведующий лабораторией интеллектуального управления исследовательского центра мультипроцессорных систем ИПС им. А.К. Айламазяна РАН (152020, г. Переславль-Залесский, ул. Трудовая, д. 5, кв. 17) тел. (910) 976-5814, e-mail: vmh48@mail.ru.

Distributed security system for cloud protection from network attacks

A. Talalaev, I. Tishchenko, V. Fralenko, V. Khachumov

The paper considers a new distributed security system for cloud protection from network attacks based on the methods of artificial intelligence. The general system architecture, its constituent software modules and components interaction are described. Graphical administrator interface is presented and the process of system provisioning is also shown.

Keywords: cloud, distributed system, protection, network attack, DDoS.