УДК 004.056

ВОЛОКИТА А.Н.,
ВУ ДЫК ТХИНЬ

ИЕРАРХИЧЕСКИЕ АГЕНТЫ БЕЗОПАСНОСТИ В РАСПРЕДЕЛЕННЫХ КОМПЬЮТЕРНЫХ СИСТЕМАХ

В статье исследованы подходы управления безопасностью распределенных компьютерных систем на основе облачных вычислений, а также разработан метод повышения эффективности реагирования на основе иерархических агентов с горизонтальными связями.

In this paper are described the approaches to the implementation of effective security mechanims for the distributed computing systems based on cloud computing, suggested a method for the reaction efficiency increasing based on the hierarchical model of agents.

Введение

В связи со стремительным развитием сетевых технологий в современном мире происходит постоянная интеграция организаций в сеть, что обуславливает множество требований по защите информации от злоумышленников. Для каждого клиента, достижение максимальной степени качества облуживания возможно благодаря индивидуальному подходу по организации и управлению безопасностью его компьютерной системы. Современные средства управления безопасностью основываются на существующих стандартах и спецификациях, что приводит к вопросу, может ли продуктивная защита быть достигнута с помощью повышения эффективности управления безопасностью с использованием современных методов защиты. Одним из важных компонентов системы безопасности являются агенты системы.

Агенты безопасности РКС

Агенты являются одними из основных компонент системы безопасности распределенных компьютерных систем (РКС), которые организованы в иерархию и предназначены для управления безопасностью РКС, в том числе в организации реагирования на вторжения. Каждый агент рассматривается как представитель сети ресурсов на метауровне управления безопасностью ресурсов. Это означает, что агент может рассматриваться как поставщик услуг для выделения средств защиты. Также агенты обмениваются сообщениями внутри своей иерархии, про изменение или выделение ресурсов пользователю, что позволяет распределять ресурсы, внутри сети, равномерно [1].

Каждый агент использует базу данных информации и полномочий (БДИП) для управления безопасностью своего сегмента РКС, а также совместного доступа к служебной информации других агентов иерархии. БДИП это таблицы содержащие информацию об агенте и набор соответствующей сервисной информации о безопасности и возможностях ресурсов в соответствующем сегменте РКС. Агент может поддерживать различные БДИП, соответствующие различным сегментам иерархии РКС:

- сБДИП – используется для записи информации о ресурсах сегмента;
- нБДИП – используется для записи служебной информации, которая полученная от агентов нижнего уровня иерархии;
- вБДИП – для записи служебной информации, которая полученная от агентов верхнего уровня иерархии.

Есть два основных варианта взаимодействия агентов с БДИП – принимать и выдавать данные, каждый из которых периодически происходит самостоятельно или может управляться системой:

- Принятие Данных – агент запрашивает у других агентов служебную информацию, периодически, либо при получении запроса.
- Выдача Данных – агент представляет свою служебную информацию для других агентов в системе, периодически или когда служебная информация изменилась.

Агент использует БДИП в качестве базы знаний, для обнаружения сервисов, которые вызываются приходом запроса. Если агент воспользовавшись БДИП не получает необходимую сервисную информацию, он может подать запрос на его верхний агент или прекратить

процесс. Средство оценки интегрировано в каждый агент и используется для оказания поддержки в процессе передачи данных. Система агента направлена на преодоление разрыва между пользователями облака и ресурсами, позволяя, таким образом, эффективнее планировать заявки на выделения ресурсов с учетом безопасности размещения. Агент может выбрать различные стратегии объявления услуг и предоставления сведений, выбор которых может привести к разным результатам работы.

Структура иерархических агентов безопасности

Структура агента безопасности показана на рис. 1. Каждый слой имеет несколько модулей, которые взаимодействуют друг с другом для выполнения служебных сообщений, или для открытия и запуска передаваемых данных [2,3].

В начальный момент функционирования агент состоит из некоторого начального набора модулей и обладает базовыми знаниями о требуемом уровне защиты, желаемой архитектуре, числе своих ближайших соседей, с которыми он должен будет обмениваться всей необходимой информацией. Сразу же после установки на некотором узле облака агент начинает осуществлять сбор сведений о среде функционирования. На основе собранной модулями анализа среды данных модулем обучения формируются убеждения об условиях функционирования, ближайшем окружении агента, возможных атаках и имеющихся средствах защиты. Это позволяет агенту адаптироваться и подстраивать свою структуру (подключенные модули) под постоянно меняющуюся среду функционирования. Постоянный сбор статистической информации позволяет сформировать убеждения о нормальном режиме работы системы, на которую установлен агент, и на их основе своевременно обнаруживать аномалии в поведении пользователя или программного обеспечения.

Центральным звеном предлагаемой архитектуры является главный модуль системы защиты, который обеспечивает синхронизацию и взаимодействие всех остальных модулей. Он же отвечает за организацию хранения и управления базами знаний, убеждений и оперативных данных.

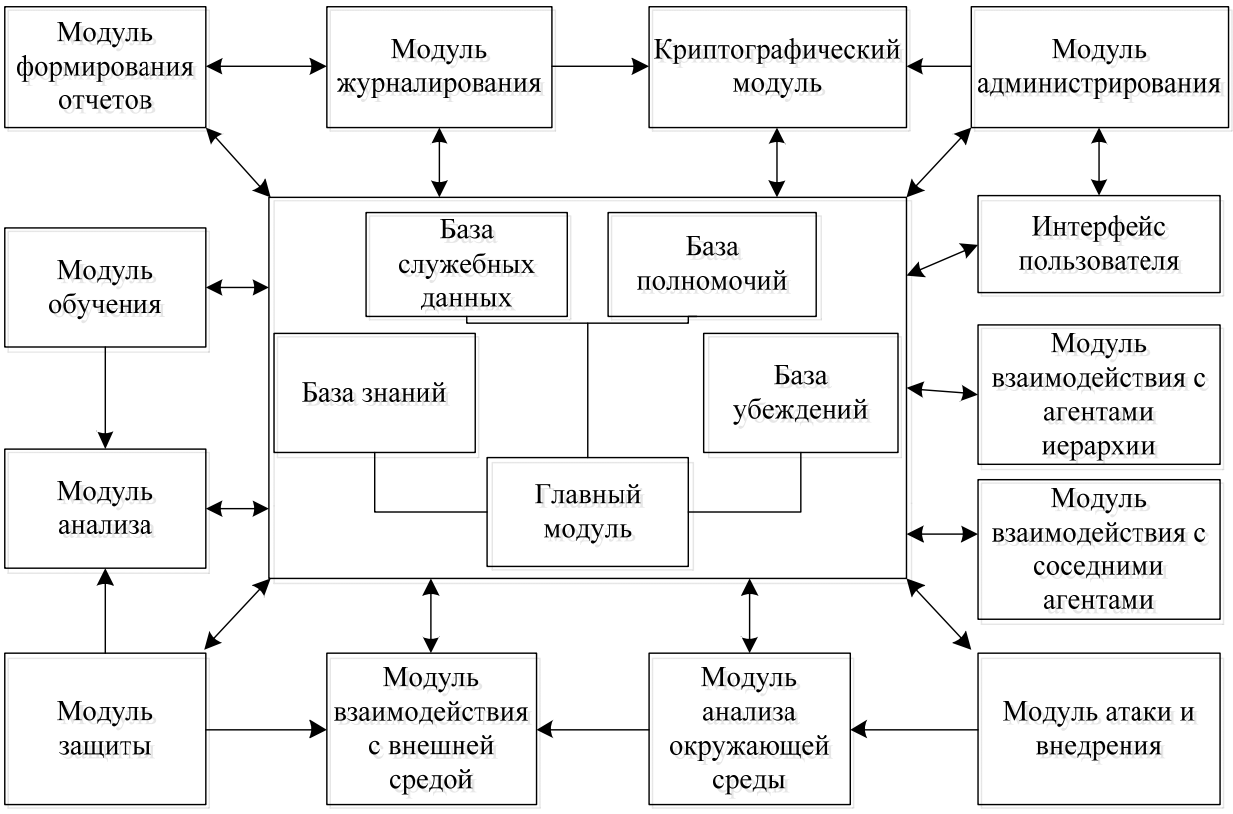


Рис. 1. Структура иерархического агента безопасности РКС

Модуль анализа – осуществляют сбор данных, накопление статистической информации о среде функционирования агента и обнаружение атак.

Модуль обучения – осуществляет формирование убеждений о нормальном функционировании агента и защищаемого элемента сети, ближайшем сетевом окружении агента, состоянии и поведении соседних агентов.

Модуль защиты – отвечают за пассивное и активное противодействие обнаруженным модулями анализа атакам и вредоносным программам, проникшим на защищаемый объект облака, как самостоятельно, так и во взаимодействии с соседними агентами.

Модуль взаимодействия с внешней средой – обеспечивают возможность обмена данными с окружающей средой, мониторинг, отправку и получение необходимой информации.

Модуль анализа окружающей среды – предназначены для сбора информации о текущей обстановке в РКС, сообщениях соседних агентов с целью выявления распределенных атак и накопления статистической информации о нормальном состоянии сети и отдельных ее компонентов.

Модуль атаки и внедрения – предназначены для поиска известных уязвимостей элементов защищаемой сети с целью их своевременного обнаружения и установки необходимых модулей защиты.

Модули взаимодействия с соседними агентами и агентами иерархии – отвечают за организацию взаимодействия отдельных агентов с соседними агентами в рамках единой системы защиты РКС или с агентами верхнего или нижнего уровня соответственно.

Модуль формирования отчетов – обеспечивает формирование отчетов системы безопасности на основе хранящейся в базе данных агента информации. При создании отчетов могут направляться дополнительные запросы соседним агентам для получения более подробной информации и формирования общего для всей защищаемой сети отчета.

Модуль журналирования – отвечает за своевременную запись в базу данных сведений о состоянии агента, подключении новых модулей, об обнаруженных аномалиях и атаках.

Криптографический модуль – предоставляет функции шифрования, проверки сертификатов и цифровой подписи.

Модуль администрирования – предоставляет интерфейсы управления как непосредственно данным агентом, так и остальными агентами сети, за счет отправки общих команд и введения новых модулей и компонентов защиты, пополнения или обновления базы исходных знаний агентов.

Управление агентами и модифицированная модель дискреционного доступа Хартсона

Для управления агентами безопасности РКС используется менеджер БДИП, который контролирует доступ агента к базе данных БДИП со служебной информацией [4]. На рис. 2 показано содержание этой служебной информации.

Рассмотрим ресурс с n каналами связи, где каждый канал P_i имеет свой тип ty_i. С учетом защищенных каналов PS_i каналы связи ресурса могут быть выражены следующим образом:

$$P = \{P_i | i = 1,2, \dots, n\}$$
$$PS = \{PS_i | i = 1,2, \dots, n\}$$
$$ty = \{ty_i | i = 1,2, \dots, n\}.$$

Пусть m количество потоков, которые запущены, или в очереди на передачу. Каждый поток данных A_j имеет два атрибута – время начала передачи ts_j и время завершения te_j. Применение облачного ресурса может быть выражено следующим образом:

$$A = \{A_j | j = 1,2, \dots, m\}$$
$$ts = \{ts_j | j = 1,2, \dots, m\}$$
$$te = \{te_j | j = 1,2, \dots, m\}.$$

MA_j множество каналов связи, MSA_j множество защищенных каналов связи, которые выделено на поток данных A_j:

$$MA = \{MA_j | j = 1,2, \dots, m\}$$
$$MA_j = \{P_j | l = 1,2, \dots, k_j\},$$
$$MSA_j = \{PS_j | l = 1,2, \dots, k_j\},$$

Где k_j число каналов связи, которые выделяются для потоков данных A_j. M и MS – это два 2-д массива, которые описывает взаимосвязи между каналами и потоками с помощью логических значений.

$$M = \{M_{ij} | i = 1,2, \dots, n;\ j = 1,2, \dots, m\}$$
$$M_{ij} = \begin{cases} 1\ if\ P_i \in MA_j \\ 0\ if\ P_i \notin MA_j \end{cases}$$
$$MS = \{MS_{ij} | i = 1,2, \dots, n;\ j = 1,2, \dots, m\}$$
$$MS_{ij} = \begin{cases} 1\ if\ PS_i \in MSA_j \\ 0\ if\ PS_i \notin MSA_j \end{cases}$$

В качестве модели дискреционного доступа будем использовать модифицированное 5-мерное пространство Хартсона [5]. Для этого расширим область безопасности до совокупности шести наборов, включив в модель множество агентов $\{A\}$:

- множество пользователей U;
- множество ресурсов R;
- множество состояний S;
- множество полномочий A;
- множество операций E;
- множество иерархических агентов I;

Тогда область безопасности представляется декартовым произведением:

$$A \times U \times E \times R \times S \times I$$

Пользователи подают запрос на доступ к ресурсам, осуществление которых переводит систему в новое состояние. Запросы на доступ представляются пятимерными кортежами:

$q = (u, e, R', s, i)$,

где $u \in U,\ e \in E,\ s \in S,\ R' \subseteq R,\ I' \in I$.

Запрос удовлетворяется, если оно полностью заключен в область безопасности.

Процесс организации доступа алгоритмически описывается следующим образом:

1. Определить из U те группы пользователей, к которым принадлежит u. Затем выбрать из A те спецификации, которым соответствуют выделенные группы. Этот набор полномочий $F(u)$ определяет привилегию пользователя u.
2. Определить из множества A набор полномочий $P = F(e)$, которые устанавливают e как основную операцию. Полномочия $P = F(e)$ определяет привилегию операции e.
3. Определить из множества A набор полномочий $P = F(R')$, разрешающих доступ к набору ресурсов R'. Полномочия $P = F(R')$ определяет привилегию ресурсов R'.
4. Определить из множества A набор полномочий $P=F(I')$, определяющие полномочия агентов i, отвечающих за безопасность запрашиваемых ресурсов R'. Полномочия $P=F(I')$ определяет привилегию агентов I'.

Полномочия, которые являются общими для всех четырех привилегий, образуют так называемый домен полномочий запроса $D(q)$:

$$D(q) = F(u) \cap F(e) \cap F(R') \cap F(I') .$$

5. Убедиться, что запрашиваемый набор ресурсов R' полностью содержится в домене запроса $D(q)$, т. е. любой r из набора R' хотя бы один раз присутствует среди элементов $D(q)$.

Убедиться, что набор агентов I' полностью содержится в домене запроса $D(q)$, т. е. любой i из набора I' хотя бы один раз присутствует среди элементов $D(q)$.

6. Осуществить разбиение $D(q)$ на эквивалентные классы так, что бы в один класс попадали полномочия, когда они специфицируют один и тот же ресурс r из набора R', а соответственно и один и тот же агент i из набора I'.

В каждом классе произвести операцию логического ИЛИ элементов $D(q)$ с учетом типа операции e. В результате формируется новый набор полномочий на каждую единицу ресурса (и соответственно агента), указанного в $D(q)$ – $F(u, q)$. Набор $F(u, q)$ – фактическая привилегия пользователя u по отношению к запросу q.

7. Вычислить условие фактического доступа, соответствующее запросу q, через операции логического ИЛИ по элементам полномочий $F(u, q)$, запрашиваемым ресурсам r из набора R' и соответствующих агентов i из набора I'. Тем самым получаем набор R'' – набор фактически доступных по запросу ресурсов и соответственно I'' – набор отвечающих за безопасность агентов.
8. Оценить условие фактического доступа и принять решение о доступе:

– разрешить доступ, если R'' и R', и I'' и I' полностью перекрываются;

– отказать в доступе в противном случае.

Агенты иерархии могут быть представлены тремя следующими слоями: коммуникаций, координаций, управления.

Коммуникационный слой каждого агента выполняет функции связи и выступает в качестве интерфейса к внешней среде. Из коммуникационных модулей, агент может получить служебные и открытые сообщения. Он интерпретирует содержимое каждого сообщения и отправляет информацию в соответствующие модули в координационный слой агента. Например, служебное сообщение от другого агента будет направлено в БДИП менеджер в слое координации агентов. Коммуникационный модуль также отвечает за отправку служебных и открытых сообщений другим агентам.

В координационном слое агента есть четыре компонента: менеджер БДИП, средство оценки безопасности, планировщик и сопоставитель. Они работают вместе, для принятия решений о том, как агент должен действовать при получении сообщения с коммуникационного слоя. Например, окончательный ответ на открытое сообщение службы, будет включать передачу данных по локальному ресурсу или отправку запроса на другой агент.

Основные функции слоя управления ресурсами в агенте это: управление потоками данных, распределение и мониторинг ресурсов. Команды передачи данных посылаются от координационного слоя к локальному менеджеру агента, эти команды включают планирование информации для данных и каналов связи. Распределение ресурсов включает в себя оболочки для различных данных.

Экспериментальные исследования

Для проведения экспериментальных исследований была использована распределенная компьютерная система с развернутым средством облачных вычислений в виде файлового хранилища (рис.2). Система состояла из семи компьютеров, на каждом из которых, было установлено программное обеспечение, реализующее облачное хранилище данных, а также один компьютер выделен под роль главного сервера. Каждый файловый сервер оснащен специальными агентами безопасности, задачей которых является определение вторжений на ранних стадиях и быстрое реагирование на них (отправка служебных сообщений на компьютер администратора).

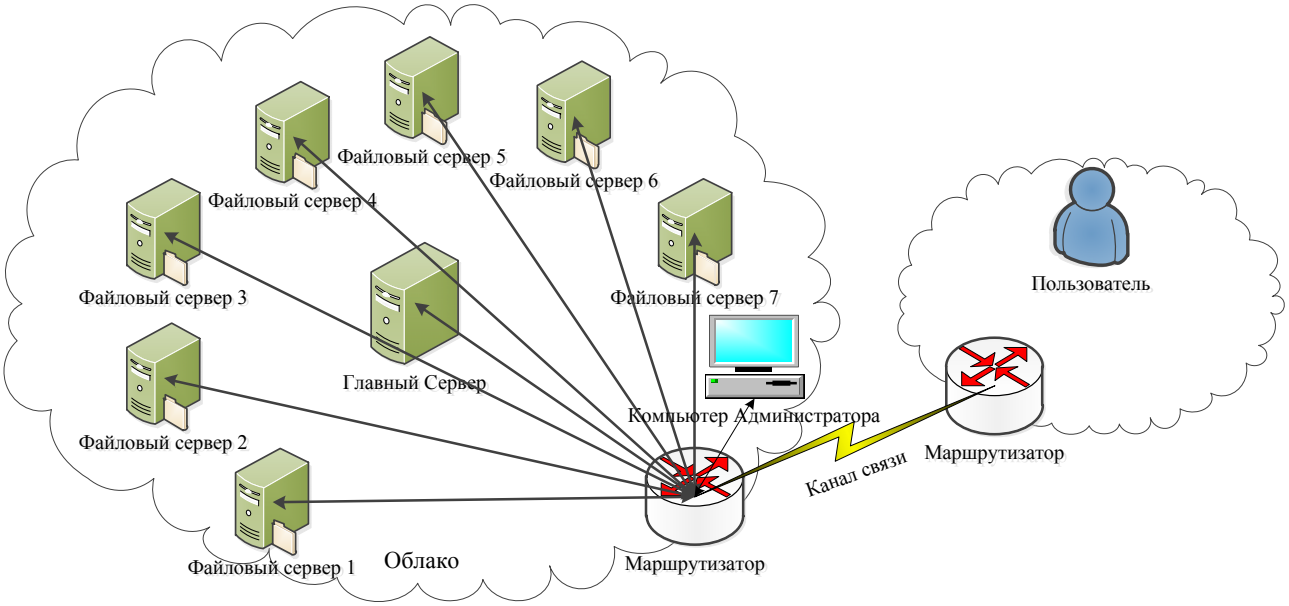


Рис. 2. Схема распределенной компьютерной системы

Следующим этапом эксперимента является внесение части псевдо-вредоносного кода в файлы разных типов (текстовый, графический, аудио, видео) и попытка их передачи на файловые серверы. После того как зараженный файл будет определен, агент безопасности останавливает его передачу и основываясь на собранной информации о сети, определяет самый быстрый путь передачи служебного сообщения с обнаруженной угрозой в блок управления безопасностью распределенной сети (в нашем случае на компьютер администратора сети) для последующей реакции на инцидент безопасности.

Также отслеживается скорость передачи сообщений в блок управления безопасностью для сбора статистики и вычисления средней скорости реагирования на вторжения.

Результаты экспериментальных исследований по оценке эффективности агентов безопасности

Процесс сбора результатов проходит в несколько этапов. Вначале смоделирована классическая модель агентов с вертикальными связями, которая реализует такую последовательность действия во время обнаружения атаки [6]:

1. Агент нижнего уровня обнаруживает угрозу на своем узле и отправляет сообщение об обнаруженной атаке на агент верхнего уровня.

2. Агент верхнего уровня, получив сообщение об угрозе, передает его главному агенту системы (агенту 1-го уровня).

3. Главный агент системы после полученного сообщения об атаке, проводит анализ сообщения.

4. После завершения анализа, всем агентам верхнего уровня системы выдается один из возможных результатов:

- да, обнаружена атака.
- нет, ложная угроза.

5. Агенты верхнего (2-го) уровня соответствующе реагируют и пересылают сообщение об угрозе на агенты нижнего (3-го) уровня.

6. Получив сообщение об угрозе от агентов 2-го уровня, агенты 3-го уровня выполняют соответствующие правила реагирования.

Далее на смоделированной классической системе проведены эксперименты, состоящие в проведении атак на сервера хранилища данных.

При этом собирается статистика по скорости реагирования на возникающие угрозы, а именно – общее время реакции всей системы на обнаруженную угрозу.

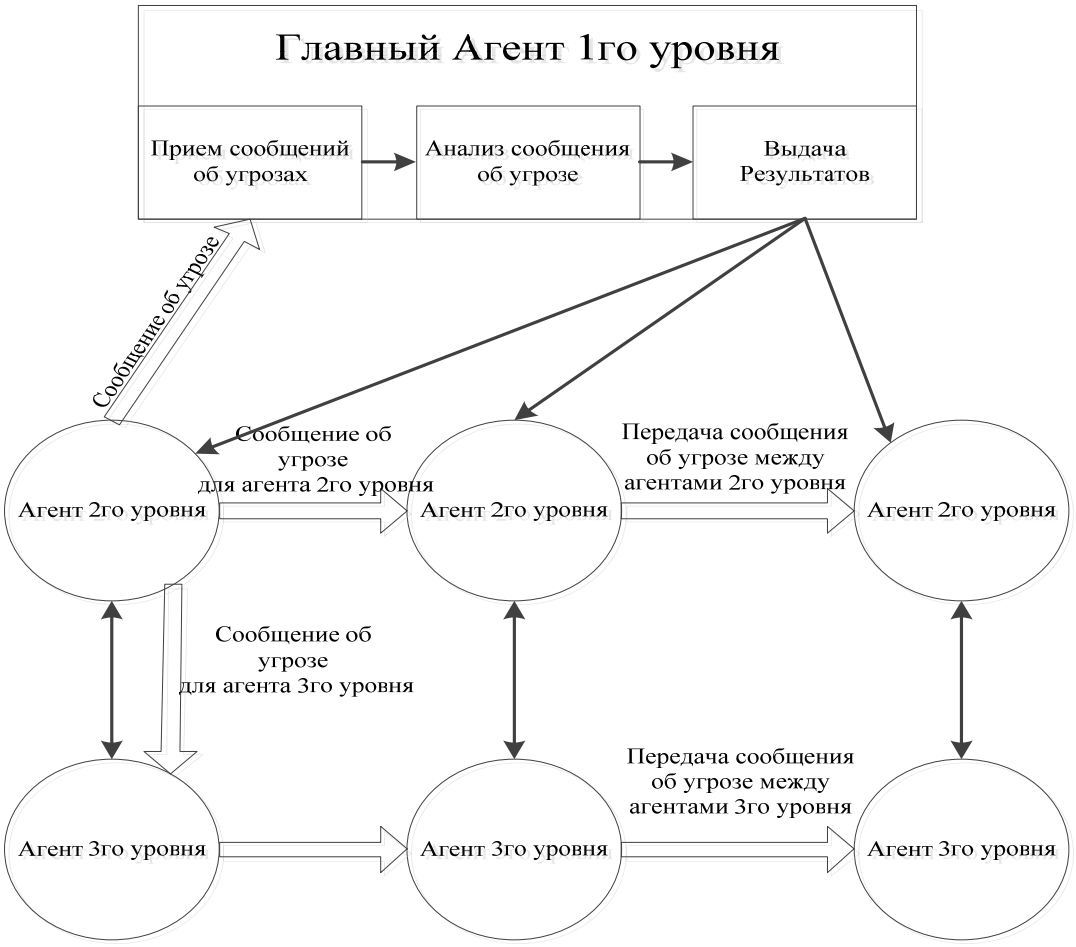


Рис. 3. Иерархическая модель агентов с горизонтальными связями

На рисунке 3 показана предложенная иерархическая модель агентов с горизонтальными связями, которая реализует следующую последовательность действия при обнаружении атаки:

1. Агент нижнего уровня обнаруживает угрозу на своем узле и отправляет сообщение об обнаруженной атаке на агент верхнего уровня, а также на соседние агенты своего уровня.

2. Агент верхнего уровня обнаруживает угрозу на своем узле и отправляет сообщение об обнаруженной атаке на Главный агент 1го уровня, а также на соседние агенты своего уровня.

3. Главный агент 1го уровня проводит анализ полученного сообщения об атаке.

4. Агент 2-го уровня после полученного сообщения об атаке от соседнего агента, выполняет соответствующее правило из базы полномочий, а также передает сообщение своему соседнему агенту 2-го уровня, аналогично агенты 3-го уровня.

5. Главный агент, проведя анализ полученного сообщения об атаке, выдает результаты агентам верхнего уровня:

- да, обнаружена атака.
- нет, ложная угроза.

6. Получив сообщение об угрозе от агентов 2-го уровня, агенты 3-го уровня выполняют соответствующие правила реагирования.

7. Все агенты иерархии, после получения соответствующего приказа о реагировании от главного агента:

- при необходимости изменяют свои правила реагирования;
- отменяют уже принятые действия, на основе сохраненных резервных копий.

Далее на смоделированной иерархической системе агентов с горизонтальными связями проведены эксперименты, аналогичные исследованиям для классической системы.

Рассмотрим фрагменты полученной статистики соотношения размера передаваемых файлов на хранилище данных со скоростью их передачи для агентов с горизонтальными связями и классических агентов (таблица 1).

Как видно из таблицы наблюдается стабильность скорости передачи данных в облачное хранилище, как для классических иерархических агентов, так и для агентов с горизонтальными связями.

Таблица 1.
Соотношение размера файлов и скорости записи в хранилище данных

Размер файла	Скорость при иерархических агентах с горизонтальными связями	Скорость при классических агентах
0,015555	0,323200632	0,323200632
0,135202	2,095765129	2,129567792
0,211943	3,285326761	3,285326761
0,314252	3,264752327	3,934445112
0,534301	4,174226563	4,786957067
1,168256	6,100935829	6,068484043
1,812223	6,653191817	6,298039229
2,395851	6,821277674	7,133226348
3,096903	6,904838667	6,920639213
3,848665	7,078082701	7,297984299
4,269056	7,212802768	7,030354132
4,830636	7,54786875	7,191185928
5,144544	6,987439152	7,30228016
6,894529	7,423305928	7,491437816
7,705764	7,647520484	7,406653697
8,478208	7,262719298	7,359555556
9,65472	7,634362348	7,447422986
10,22208	7,608612805	7,696607556
12,00078	7,498088112	7,611611562
13,1072	7,450523865	7,446189645
15,605589	7,619916504	7,446189645
17,355912	7,690169153	7,746404393
30,360102	7,712938894	7,712938894
44,655521	7,752694618	7,797051176
54,955644	7,806199432	7,650409279
64,183644	7,743926346	7,714380288
66,819222	7,835392229	7,734164571
77,499994	7,800019364	7,725179942
86,663502	7,770849892	9,654611701
89,828166	9,515437505	9,597682534
108,000054	9,506832769	9,669826967
117,143208	7,830093365	7,698362319
120,892616	7,813315375	7,702191761
217,396657	9,622509308	9,691031307

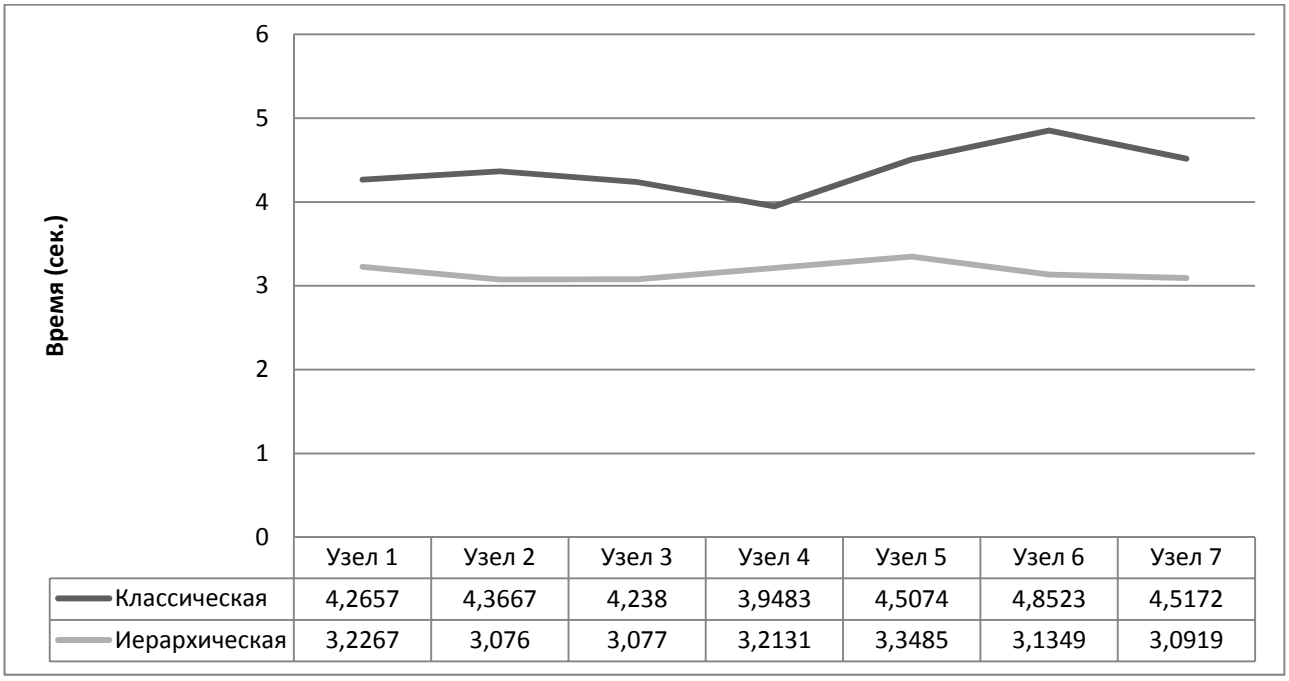
Рассмотрим полученную статистику времени реагирования на угрозу. Всего было проведено 10 экспериментов для каждой из системы, во время которых по сети выполнялась передача данных на сервера хранилища данных.

При каждом эксперименте менялась структура файлов, записываемых в облачное хранилище данных. В файлы добавлялся псевдовредоносный код, на который агенты должны реагировать.

Время реакции системы классических агентов колеблется по результатам проведенных экспериментов в рамках минимального значения 2 секунды и максимального 6 секунд.

Для системы с иерархическими агентами время реакции колеблется от 2,5с до 4,3с., что объясняется наличием упреждающего реагирования на обнаруженную угрозу за счет горизонтальной передачи сообщения соседним агентам.

Как видно из рисунка 4 среднее время реакции каждого узла на потенциальную угрозу модели с классическими агентами составляет от 4 до 5 секунд, в свою очередь система с горизонтальными связями реагирует на угрозы в среднем в рамках от 3 до 3.35 секунд.



	Узел 1	Узел 2	Узел 3	Узел 4	Узел 5	Узел 6	Узел 7
Классическая	4,2657	4,3667	4,238	3,9483	4,5074	4,8523	4,5172
Иерархическая	3,2267	3,076	3,077	3,2131	3,3485	3,1349	3,0919

Рис. 4. Сравнение среднего времени реакции классической и иерархической моделей с горизонтальными связями

Как видно из графиков (рис. 4), разработан-ный метод на основе иерархических агентов с горизонтальными связями позволяет повысить эффективность управления безопасностью компьютерных систем на основе cloud computing за счет снижения скорости реакции системы на вторжение, практически не понизив скорости записи файлов в облачное хранилище данных.

Эксперименты показали, что предложенные средства управления безопасностью на основе иерархических агентов имеют меньшее время реагирования на вторжение, построены таблицы и диаграммы соотношения времени реакции данных и классических агентов

Заключение

В больших распределенных компьютерных системах, на которых построены современные “гиганты” облачных вычислений, очень часто, если не постоянно происходит реконфигурация системы, в связи с постоянным появлением, модификацией и уничтожением виртуальных машин и пользователей. В связи с этим требуется повысить эффективность управления безопасностью. Предложенный метод иерархических агентов с горизонтальными связями позволяет повысит эффективность управления безопасностью распределенной компьютерной системы.

Список литературы

1. Тарасов В. Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – М.: Эдиториал УРСС, 2002.
2. Tesfatsion L., Judd K. L., Handbook of computational economics: agent-based computational economics. Vol. 2. North-Holland, 2006.
3. Weiming Shen, Distributed Manufacturing Scheduling Using Intelligent Agents, National Research Council Canada, 2002
4. Junwei C., Stephen A. Jarvis, Subhash Saini, Darren J. Kerbyson and Graham R. Nudd Department of Computer Science University of Warwick, Coventry, CV4 7AL, UK ARMS: An agent-based resource management system for grid computing 2002.
5. Гайдамакин Н.А. Разграничение доступа к информации в компьютерных системах. – Екатеринбург: изд-во Урал. Ун-та, 2003. – 328 с.
6. Радченко Г.И., Распределенные вычислительные системы, Челябинск, 2012