Агентный подход к реализации модели поведения пользователей Grid-систем

Шелестов А.Ю., Скакун С.В., Куссуль О.М.
Институт космических исследований НАНУ-НКАУ
inform@ikd.kiev.ua

Abstract

Shelestov A., Skakun S., Kussul O. Agent approach to implementation of model of users behavior in Grid systems. *We propose an agent approach to the implementation of model of users behavior in Grid systems. Proposed model is based on the analysis of statistical data about jobs that were completed by the user in Grid system. In order to distinguish normal and abnormal user's behavior we use feed-forwards neural networks. We verified the proposed approach on data that were collected by GridICE monitoring system in GILDA-EGEE. The results of experiments showed that our model was able to distinguish abnormal user's behavior in 90%.*

Введение

На сегодняшний день технология Grid стала фактическим стандартом построения распределенных систем для эффективного решения вычислительных задач в различных предметных областях [1-3]. Технология Grid предполагает использование программного обеспечения среднего уровня (middleware), предназначенного для объединения распределенных информационных и вычислительных ресурсов различных административных доменов в рамках единой виртуальной организации (ВО) [4]. При этом Grid-платформа призвана решить такие задачи, как гибкое, безопасное и согласованное совместное использование ресурсов, а также обеспечение параллельных высокопроизводительных вычислений и распределенной обработки данных. Очевидно, что одной из важнейших задач при разработке подобных сложных систем является реализация механизмов обеспечения безопасности [4], в частности, обеспечение аутентификации и авторизации, обмен сертификатами, обеспечение конфиденциальности и целостности данных, а также аудит и мониторинг ресурсов и пользователей [4]. В настоящее время большинство этих задач при построении Grid-систем решается на основе инфраструктуры Grid Security Infrastructure (GSI) [5], которая по существу представляет собой расширение инфраструктуры открытого ключа (PKI — Public Key Infrastructure) [6]. GSI-инфраструктура поддерживает одноразовую регистрацию (single sign on), делегирование полномочий и обмен сертификатами.

Вместе с тем следует отметить, что одним из других наиболее важных аспектов обеспечения безопасности является мониторинг действий пользователей при работе с удаленными ресурсами Grid-системы.

Анализ моделей безопасности Grid-систем

В настоящее время существует достаточно много средств мониторинга состояния ресурсов Grid-системы и запускаемых задач (например, GridICE [7] и MOGAS [8]). Однако эти средства не предоставляют средств анализа работы пользователей для выявления их аномальной деятельности. Поэтому на сегодняшний день разработка методов и моделей анализа поведения пользователей в сложных распределенных Grid-системах является актуальной задачей.

Рассмотрим существующие средства обеспечения безопасности в Grid-системах более подробно. В настоящее время при разработке большинства Grid-систем используется программное обеспечение Globus Toolkit. Как уже упоминалось выше, для обеспечения безопасности в данном программном обеспечении (а, следовательно, и в прикладных Grid-системах) используется инфраструктура GSI [5]. Основанный на технологии открытых ключей протокол GSI обеспечивает аутентификацию и однократную регистрацию пользователей, а тажк ограниченный набор средств делегирования полномочий. Для идентификации пользователей в инфраструктуре GSI используются сертификаты X.509 [9] рабочей группы IETF (Internet Engineering Task Force). Специалистами этой же группы был определен также документ, регламентирующий расширение сертификатов X.509 для обеспечения поддержки прокси-сертификатов [9]. При реализации в Grid-среде Web-сервисов (или Web-служб) защита обеспечивается уже на уровне сообщений протокола SOAP (Simple Object Access Protocol — простой протокол доступа к объектам). Таким образом, системные средства защиты в Grid-системах развиты достаточно хорошо. Отдельным вопросом обеспечения безопасности Grid-систем

является анализ поведения пользователей [10] и [11].

В работе [10] предложен механизм авторизации WAS (Workflow-based Authorization Service), основанный на анализе потока выполнения задач на ресурсах Grid-системы и используемых при этом пользовательских привилегий (или разрешений). Основная идея этого подхода состоит в следующем. При отправке пользователем задачи на выполнение в Grid-систему сервис WAS сначала автоматически анализирует исходный код программы и определяет набор разрешений, которые понадобятся для ее выполнения. (Возможность анализа исходного кода программы является обязательным условием.) Затем полученный набор разрешений наряду с информацией о пользователе передается специальному модулю (WAS-server module), который проверяет набор пользовательских разрешений на соответствие принятой политике безопасности и оценивает их корректность. После успешного прохождения проверки задача и набор разрешений отправляется непосредственно на ресурс Grid-системы. В процессе выполнения задачи сервис WAS осуществляет мониторинг запрашиваемых ею разрешений и сравнивает их со сгенерированным ранее набором. При обнаружении несоответствия выполнение задачи сервисом WAS будет прервано.

Для проверки адекватности предложенного подхода сервис WAS был реализован в программном комплексе Globus Toolkit версий 3.x и 4.x. Однако результаты каких-либо проведенных экспериментов по оценке его эффективности в известной литературе не приводятся.

В работе [11] предложен модуль мониторинга поведения пользователей Grid-системы, основанный на применении механизма MOGAS [8]. Данное средство позволяет собирать данные о состоянии задач, запущенных в Grid-среде на основе Globus Toolkit, и размещать ее в централизованном хранилище. Однако информация о возможных отказах при аутентификации или авторизации при этом не предоставляется. Авторами предложен сценарий для службы Globus gatekeeper, который позволяет собирать информацию об отказах и просматривать ее через Web-интерфейс. Существенным недостатком данного подхода является отсутствие средств анализа собранных данных, что существенно снижает ценность предложенного подхода.

Идея предлагаемого подхода

Анализ существующих средств обеспечения безопасности в Grid-системах свидетельствует о том, что методы и средства мониторинга поведения пользователей в Grid-системах в настоящее время развиты недостаточно, а имеющиеся средства мониторинга состояния ресурсов Grid-систем (например, GridICE, MOGAS) не обеспечивают необходимой функциональности. Лишь в некоторых работах (например, в [10] и [11]) рассматриваются вопросы, связанные с мониторингом деятельности пользователей на основе анализа потока выполнения задач, требуемых для этого разрешений, а также об отказах системы.

В настоящее время разработано достаточно много методов анализа и моделей поведения пользователей компьютерных сетей.

Так, в работах [12] предложена комплексная модель поведения пользователей компьютерных систем, которая состоит из трех компонентов (интерактивной (прогнозной) составляющей, сеансовой (статистической) составляющей и модуля анализа трендов) и позволяет учесть как динамические, так и статистические свойства поведения пользователей, а также возможные тренды его поведения.

Вместе с тем работа пользователей в Grid-системе имеет свою специфику и особенности (решение сложных задач на распределенных высокопроизводительных ресурсах). Поэтому в данной работе ставится задача модифицировать комплексную модель и учесть специфику таких пользователей.

При построении моделей поведения пользователей можно выделить следующие общие этапы:

1. Сбор и предварительная обработка данных о работе пользователей.
2. Анализ данных для выделения информативных признаков или уменьшения размерности данных (создание так называемого профиля пользователя).
3. Разработка методов обработки данных и построение модели.
4. Верификация модели и интерпретация полученных результатов.

В данной статье модель поведения пользователя Grid-системы строится на основе нейросетевого подхода. При этом рассматриваются все перечисленные выше этапы построения модели. Такая модель должна обеспечить возможность выявления характерных (аномальных) действий пользователя Grid-системы. Если результаты работы пользователя соответствуют ранее построенной модели, то такое поведение можно считать нормальным. В противном случае поведение пользователя считается аномальным.

Для реализации модели поведения пользователей предлагается использовать агентный подход.

Структура модели

Для анализа статистических данных о работе пользователя с целью выявления аномалий предлагается использовать нейронные сети [13]. Применение нейронных сетей обеспечивает интеллектуальный и робастный подход к анализу и обобщению данных о работе пользователя. Для решения поставленной задачи наилучшим выбором является многослойная сеть прямого распространения, которая согласно теореме Колмогорова является универсальным аппроксиматором [13] и может эффективно применяться как для решения задач прогнозирования, так и классификации.

В работе [12] сеансовая модель основывалась на анализе следующей информации о работе пользователя: количество команд, относительное количество правильно спрогнозированных команд за сеанс, номер компьютера в сети, за которым работал пользователь; продолжительность сеанса; время начала сеанса. При этом за сеанс пользователь может выполнять десятки (а иногда и сотни) команд, необходимых для решения его задач. При работе пользователей в Grid-среде существуют свои особенности: пользователь выполняет небольшое количество трудоемких задач на высокопроизводительных ресурсах Grid-системы. Поэтому целесообразно собирать и анализировать информацию о запускаемой задаче (т.е. аналогом сеанса в исходной модели будет процесс запуск задачи в предлагаемой сеансовой модели). Таким образом, при построении модели поведения пользователя предлагается учитывать следующую информацию:

$$\{S, ET, CPU, WT, CW, ES, CT, STD, RAM, VM, VO, RB\}, \quad (1)$$

где S (Site) — узел, на котором выполнялась задача; ET (Execution Target) — ресурс узла, на котором выполнялась задача; CPU (CPU Time) — время работы процессора ресурса при выполнении задачи; WT (Wall Time) — полное время выполнения задачи; CW (CPUWall = CPU/W) — отношение времени работы процессора к общему времени выполнения задачи; ES (ExitStatus) — статус завершения задачи (успешное или с ошибкой); CT (Creation Time) — время отправки (создания) задачи в Grid-систему; STD (Start Time Difference) — разница между временем начала выполнения задачи на выбранном ресурсе Grid-системы (выбор конкретного ресурса, на котором будет выполняться задача, обеспечивается брокером ресурсов) и временем отправки задачи в Grid-систему; RAM (RAM Used) — используемая оперативная память; VM (Virtual Memory Used) — используемая виртуальная память; VO (Virtual Organization Name) — принадлежность к виртуальной организации; RB (Resource Broker Hostname) — брокер ресурсов, который использовался для распределения задачи.

Этот набор данных выступает в качестве входных признаков для выявления нормальной или аномальной работы пользователя. Для решения этой задачи используется нейросетевая модель. Для каждого пользователя Grid-системы строится нейронная сеть, которая на основе доступной информации относит поведение пользователя к классу нормального или аномального. При этом ожидаемый выход нейронной сети может принимать два значения: 1 — для нормального поведения пользователя и 0 — для аномального. Другими словами, нейронная сеть должна функционировать в качестве классификатора.

Пусть $u \in U$ — некоторый пользователь Grid-системы (U — множество пользователей), s_t^u ($t \in \{1, 2, \ldots\}$) — набор задач, запущенных пользователем u в Grid-системе, для которых имеется следующий набор данных (1):

$$\mathbf{x}_{s_t^u} = \begin{Bmatrix} S_{s_t^u}, ET_{s_t^u}, CPU_{s_t^u}, WT_{s_t^u}, CW_{s_t^u}, ES_{s_t^u}, \\ CT_{s_t^u}, STD_{s_t^u}, RAM_{s_t^u}, VM_{s_t^u}, VO_{s_t^u}, RB_{s_t^u} \end{Bmatrix}.$$

Тогда выход нейронной сети по завершении задачи s_t^u определяется следующим соотношением (рис. 1):

$$\Delta_{s_t^u} = F(\mathbf{x}_{s_t^u}),$$

где F — нелинейное преобразование нейронной сети прямого распространения; $\mathbf{x}_{s_t^u}$, $\Delta_{s_t^u}$ — вход и выход сети (в данном случае размерность вектора $\mathbf{x}_{s_t^u}$ составляет 12, а $\Delta_{s_t^u}$ — 1).

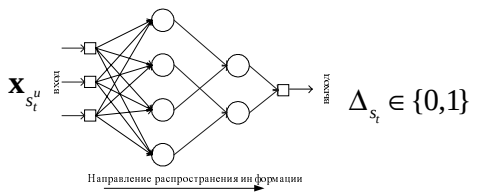

Рисунок 1 – Структура сеансовой модели

На вход нейросетевой модели поступает следующая информация:

– Ресурс сегмента, виртуальную организацию и брокер ресурсов будем нумеровать целыми числами 1, 2, 3, ...;

– время работы процессора, полное время выполнение задачи и разница между временем начала выполнения задачи на ресурсе Grid-системы и временем отправки задачи в Grid-систему измеряется в секундах;

– отношение времени работы процессора к общему времени выполнения задачи представляется числом из отрезка [0; 1];

– статус завершения задачи описывается бинарным значением: 0 — успешное выполнение и 1 — с ошибкой;

– время отправки (создания) задачи в Grid-систему измеряется в минутах (с начала дня) и нормируется на 24 часа;
– оперативная и виртуальная память измеряются в Гбайтах.

В работе [13] показано, что если при обучении нейронной сети желаемый выход принимает два значения (например, 0 и 1; т.е. нейронная сеть разделяет входное пространство на два класса), то при подаче на ее вход независимого образа на выходе будет получена вероятность принадлежности этого образа к одному или другому классу. Таким образом, значение $\Delta_{s_t^u}$ будет принадлежать отрезку [0; 1] и определять вероятность нормального (соответствующего модели) поведения пользователя.

Такая модель обладает следующими преимуществами:
– независимость от количества пользователей в системе, поскольку для каждого пользователя строится своя нейросетевая модель;
– адаптация к изменению поведения пользователей;
– использование интеллектуальных методов обработки данных.

Поскольку нейросетевые модели по своей природе являются индукционными, первоочередную роль в их построении играют экспериментальные данные. Рассмотрим структуру данных и источники их получения более подробно.

Описание структуры данных

При построении модели были использованы данные, полученные в результате работы пользователей в обучающей системе GILDA (https://gilda.ct.infn.it/) европейского проекта EGEE (http://www.eu-egee.org/). Система GILDA объединяет ресурсы 12 организаций, насчитывая в общей сложности 112 процессоров и возможность хранения до 5,2 Тбайт данных.

Для мониторинга состояния ресурсов и задач в Grid-системе GILDA используется распределенная система GridICE, которая интегрируется с локальной системой мониторинга ресурса и предоставляет стандартный интерфейс для отображения данных на уровне Grid-системы. Распространение данных может выполняться на двух уровнях иерархии: локальном (конкретного ресурса) и всей Grid-системы. При этом существует несколько подходов для отображения данных мониторинга: в графическом или текстовом виде посредством Web-интерфейса или с использованием формата XML. Среди полезных свойств системы GridICE можно выделить следующие:
– автоматическое обнаружение новых ресурсов посредством использования службы Grid Index Information Service (GIIS);
– мощный инструментарий для отображения данных мониторинга через Web-интерфейс;
– наличие служб уведомления;
– полный набор метрик мониторинга (для отдельного ресурса и всей системы в целом);
– поддержка таких систем управления задачами, как OpenPBS, Torque, LSF;
– предоставление данных в формате XML;
– открытый код.

В табл. 1 приведены данные о задачах, которые предоставляет система мониторинга GridICE.

Таблица 1. Данные системы GridICE

Название	Описание
Job LocalID	Локальный идентификатор задачи
Name	Название задачи
JobStatus	Статус выполнения задачи (E — выполнена, W — ожидание, R — в процессе выполнения)
LocalOwner	Владелец задачи
Execution Target	Ресурс сайта, на котором выполнялась задача
CPU Time	Время работы процессора ресурса при выполнении задачи
Wall Time	Полное время выполнения задачи
CPUWall	Отношение времени работы процессора к общему времени выполнения задачи
Exit Status	Статус завершения задачи (успешное выполнение или с ошибкой)
Creation Time	Время отправки (создания) задачи в Grid-систему
Start Time	Время начала выполнения задачи
Start Time Difference	Разница между временем начала выполнения задачи на выбранном ресурсе Grid-системы (выбор конкретного ресурсаобеспечивается брокером ресурсов) и временем отправки задачи в Grid-систему
End Time	Время завершения задачи
RAM Used	Используемая оперативная память
Virtual Memory Used	Используемая виртуальная память
Virtual Organization Name	Принадлежность к виртуальной организации
Site	Сайт организации, на котором выполнялась задача
Resource Broker Hostname	Брокер ресурсов, который использовался для распределения задачи
GlobalID	Глобальный идентификатор задачи

Приведем пример данных в формате XML с описанием задачи:

```
<Job LocalID="5794">
<Name>STDIN</Name>
<JobStatus>E</JobStatus>
<LocalOwner>gilda001</LocalOwner>
<ExecutionTarget>iceage-wn-13</ExecutionTarget>
<CPUTime UnixTime="7">00:00:07</CPUTime>
<WallTime UnixTime="27">00:00:27</WallTime>
<CPUWall>0.25925925925926</CPUWall>
<ExitStatus>0</ExitStatus>
<CreationTime UnixTime="1175270319">2007-03-30 17:58</CreationTime>
<StartTime UnixTime="1175270320">2007-03-30 17:58</StartTime>
<StartTimeDiff UnixTime="1">00:00:01</StartTimeDiff>
<EndTime UnixTime="1175270347">2007-03-30 17:59</EndTime>
<RAMUsed>19576</RAMUsed>
<VirtualUsed>45000</VirtualUsed>
<VOName>gilda</VOName>
<Site>ICEAGE-CATANIA</Site>
<GlobalID>https://glite-rb.ct.infn.it:9000/Ba5Xi27XOjie4e2sk8ZJug</GlobalID>
<RBHostname>glite-rb.ct.infn.it</RBHostname>
</Job>
```

Для проведения экспериментов и проверки адекватности предложенной модели в системе GILDA были собраны данные с 30 марта 2006 г. по 2 апреля 2007 г. (всего 34000 записи). Данные, полученные в формате XML, были преобразованы в формат, пригодный для дальнейших экспериментов. Затем для каждого пользователя данных разбивались на обучающую (85%) и тестовую (15%) выборки.

Структурная идентификация модели

В качестве нейросетевой модели была выбрана многослойная нейронная сеть прямого распространения информации (персептронного типа) с одним скрытым слоем, обучаемая по методу обратного распространения ошибки. Были проведены эксперименты по определению оптимальной размерности скрытого слоя нейронной сети. Оптимальной считалась такая размерность, при которой средний процент правильной классификации поведения пользователей для тестовой выборки всех пользователей был максимальным. В процессе экспериментов оказалось, что оптимальная размерность скрытого слоя составляет 20 нейронов, при которой достигается 85,81% правильной классификации. При большем количестве нейронов в скрытом слое наступало насыщение, и дальнейшее увеличение не позволяло повысить процент правильной классификации.

Для каждой модели пользователя были определены весовые коэффициенты и параметры обучения (η=0,3, μ=0,15). При этом использовался некумулятивный вариант метода обратного распространения ошибки. При таком подходе в процессе обучения присутствует элемент случайности, что позволяет повысить вероятность непопадания в локальный минимум.

Экспериментальная верификация модели

Для проверки эффективности предложенной модели пользователей Grid-системы была проведена серия экспериментов. Для того чтобы проверить, насколько нейронная сеть была способна отличить поведение одного пользователя от другого, использовалась процедура подмены пользователя. На вход нейронной сети, которая была обучена для одного пользователя (легального), подавались данные другого пользователя (нелегального). Так имитировалась ситуация, когда нелегальный пользователь работает под именем (учетной записью) легального пользователя.

Результаты экспериментов показали, что процент правильной классификации для легального пользователя на тестовой выборке составил 99,14% (соответствует ошибке первого рода). В случае моделирования процедуры подмены пользователя процент правильной классификации нелегального пользователя составил 99,30% (соответствует ошибке второго рода). Приведенные результаты показывают, что предложенная модель позволяет уверенно обнаружить подмену пользователя, поэтому является достаточно эффективной.

Агентная реализация модели поведения пользователей

При реализации предложенной модели поведения пользователей необходимо учитывать распределенность и гетерогенность Grid-системы, а также большое количество пользователей, для каждого из которых строится своя модель. При этом с целью уменьшения нагрузки на систему модель поведения целесообразно реализовывать как автономный модуль, который будет и инкапсулировать нейросетевые методы анализа статистических данных о работе пользователей. Кроме того, при реализации необходимо обеспечить взаимодействие с системой мониторинга для получения информации о задачах запущенных пользователями и сертификационным центром (CA — Certificate Authority). Этим требованиям лучше всего удовлетворяет агентная парадигма [14].

В качестве базового определения агента выберем следующее: агент — это сущность, которая может принимать информацию из внешней среды и реагировать на внешнее возмущение [14]. В общем случае агент определяется набором:

<S, Prog, Eff, Arch, P, A, G, E>, (2)

где E (environment) — внешняя среда, в которой функционирует агент;

S (sensors) — множество входов, с помощью которых агент воспринимает информацию из внешней среды;

Eff (effectors) — множество выходов, с помощью которых агент влияет на внешнюю среду;

P (percepts) — информация, которую получает агент;

A (actions) — реакция агента;

Prog (program) Prog: P–>A — функция, определяющая зависимость реакции агента от входных воздействий;

G (goal) — цели, которые достигает агент;

Arch (architecture) — физическая оболочка, которая объединяет все базовые элементы агента.

Набор S, Prog, Eff, Arch определяет базовую конструкцию агента (его каркас), в то время как P, A, G, E — его содержательное "наполнение".

Для реализации модели поведения пользователей будем использовать так называемые программные агенты (software agents). Их отличительная особенность состоит в том, что они являются компьютерными программами и функционируют в компьютерных системах. Для программного агента (в терминах данного выше определения (2)): E=Grid-система, Arch=программа (код), а S и Eff — некоторые функции открытого программного интерфейса, посредством которых агент обменивается информацией с внешней средой.

Архитектура агентной системы безопасности в контексте системы мониторинга GridICE представлена на рис. 3.

Система мониторинга GridICE является централизованной. Для сбора информации о задачах, выполняемых пользователями, на каждом ресурсе запущена служба Grid Resource Information Service (GRIS). В свою очередь, службы доменов Grid Index Information Service (GIIS) взаимодействуют с локальными службами GRIS, агрегируют полученную информацию и отправляют на централизованный сервер GridICE (рис. 2).

Для реализации модели поведения пользователя предлагается создать следующие типы агентов:

– агент, инкапсулирующий модель пользователя Grid-системы (User Agent);

– агент-контроллер (Controller Agent), управляющий работой агентов в системе.

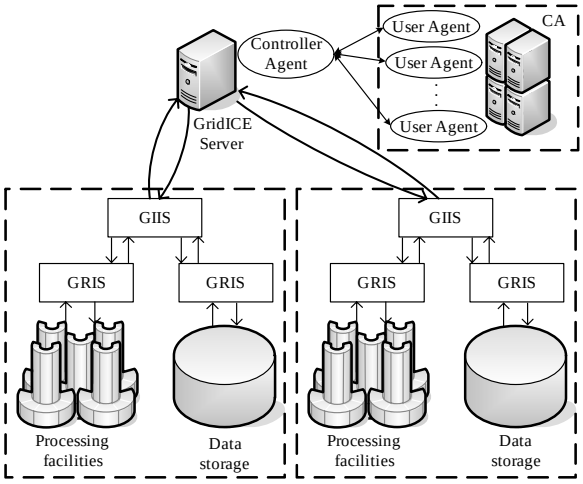

Рисунок 2 – Архитектура системы обеспечения безопасности Grid-систем на основе модели поведения пользователей и агентного подхода

Агент, инкапсулирующий модель пользователя, должен функционировать на том же ресурсе, что и сертификационный центр. Данный агент активируется после выполнения пользователем задач и имеет следующие значения параметров набора, который определяется формулой (2):

G — выявление нехарактерной деятельности пользователя;

P — статистические данные о деятельности пользователя;

A — вероятность характерной деятельности пользователя;

Prog: P–>A — вычисление вероятности $\Delta_{s_t^u}$ нехарактерной деятельности пользователя с использованием нейронной сети.

Агент-контроллер размещен на сервере системы мониторинга GridICE и ответственен за создание агентов, инкапсулирующих модель поведения пользователя. В терминах выражения (2) этот тип агента имеет следующее описание:

– G — управление системой мониторинга в целом и создание пользовательских агентов;

– P — данные о различных элементах системы;

– A — организация взаимодействия между элементами системы;

– Prog: P–>A — создание агентов, получение информации из системы мониторинга, переобучение нейронных сетей (т.е. адаптация модели).

Этот агент получает из системы GridICE статистические данные о работе пользователя в Grid-системе. После этого для каждого пользователя он создает агента User Agent на ресурсах сертификационного центра. На основе полученной информации и параметрах

нейросетевой модели, которые хранятся в СА, пользовательский агент выясняет, на сколько деятельность пользователя была аномальной (что определяется числом из промежутка [0, 1], указывающим вероятность нормального поведения пользователя). В случае выявления аномальной деятельности соответствующая информация помещается в файлах-аудита Grid-системы и отправляется администраторам.

Выводы

В данной статье рассмотрены существующие средства обеспечения безопасности в Grid-системах и подходы к построению моделей поведения пользователей таких систем. Существующие средства обеспечения безопасности в Grid-системах (например, Globus GSI), позволяют обеспечить аутентификацию, авторизацию, обмен сертификатами и использовать ряд других важных подходов к обеспечению безопасности. Вместе с тем средства мониторинга поведения пользователей в Grid-системах недостаточно развиты и не позволяют выявлять их аномальное поведение. Поэтому в данной работе предложен новый подход к анализу поведения пользователей и выявлению аномалий в их работе.

В предложенной модели поведения пользователей Grid-системы учитывается ряд параметров (профиль пользователя), получаемые при выполнении задачи на ресурсах Grid-системы. Для реализации предложенной модели была использована нейронная сеть прямого распространения, которая функционирует в режиме классификатора. На основе проведенных экспериментов была определена оптимальная структура нейросетевой модели, а именно 12-20-1 (20 нейронов в скрытом слое), а также значения весовых коэффициентов и параметров обучения (η=0,3, μ=0,1).

Для проверки эффективности предложенной модели были проведены эксперименты на реальных данных, которые были собраны с помощью средства мониторинга GridICE в Grid-системе GILDA-EGEE. Результаты экспериментов показали, что в 90% случаев использование модели позволяет обнаружить подмену пользователя. Таким образом, верификация модели на реальных данных подтвердила эффективность ее применения для выявления аномальной деятельности пользователей в Grid-системах.

Для реализации предложенной модели используется агентный подход. Это дало возможность использовать предложенный подход в гетерогенной Grid-среде, обеспечить взаимодействие с системой мониторинга Grid-системы (на примере, системы GridICE), а также обеспечить автономность модулей, инкапсулирующих нейростевые методы обработки статистических данных. Использование агентной технологии позволило обеспечить распределенность и масштабируемость системы выявления аномалий.

Работа выполнена при поддержке гранта INTAS-CNES-NSAU “Data Fusion Grid Infrastructure” (Ref. Nr 06-1000024-9154).

Литература

1. Shelestov A.Yu., Kussul N.N., Skakun S.V. Grid Technologies in Monitoring Systems Based on Satellite Data, J. of Automation and Information Science, 2006, Vol. 38, Issue 3, pp. 69-80.
2. Fusco L., Goncalves P., Linford J., Fulcoli M., Terracina A., D’Acunzo G. Putting Earth-Observation on the Grid, ESA Bulletin, 2003, 114, pp. 86-91.
3. Peltier S.T., et al. The Telescience Portal for Advanced Tomography Applications, J. of Parallel and Distributed Computing: Computational Grid, 2002, 63(5), pp. 539-550.
4. Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Int. J. Supercomputer Applications, 15(3), 2001.
5. Foster I., Kesselman C., Tsudik G., Tuecke S. A Security Architecture for Computational Grids. In ACM Conf. on Computers and Security, 1998, pp. 83-91.
6. Adams C., Lloyd S. Understanding PKI: Concepts, Standards, and Deployment Considerations. 2nd ed. Addison-Wesley, 2000.
7. S. Andreozzi, N. De Bortoli, S. Fantinel, A. Ghiselli, G.L. Rubini, G. Tortone and M.C. Vistoli. GridICE: a Monitoring Service for Grid Systems. In Future Generation Computer Systems Journal, Elsevier, 2005, 21(4), pp. 559-571.
8. MOGAS. — http://ntu-cg.ntu.edu.sg/pragma/index.jsp.
9. Tuecke S., Engert D., Foster I., Thompson M., Pearlman L., Kesselman C. Internet X.509 Public Key Infrastructure Proxy Certificate Profile // IETF, Draft draft-ietfpkix-proxy-01.txt, 2001.
10. Seung-Hyun K., Kyong H.K., Jong K., Sung-Je H., Sangwan K. Workflow-Based Authorization Service in the Grid. J. of Grid Computing, 2004, Num. 2, P. 43–55.
11. Shingo T., Susumu D., Shinji S. A user-oriented secure filesystem on the Grid // The 3rd IEEE/ACM Int. Symp. on Cluster Computing and the Grid (CCGrid 2003), May, 2003.
12. Куссуль Н.Н., Скакун С.В. Нейросетевая модель пользователей компьютерных систем // Кибернетика и вычислительная техника. — 2004. — Выпуск 143. — С. 55–68.
13. Haykin S. Neural Networks: a comprehensive foundation. Upper Saddle River, New Jersey: Prentice Hall, 1999.
14. Russel, S., Norvig, P. Artificial Intelligence: A Modern Approach. — Upper Saddle River NJ: Prentice Hall, 1995. — 932 p.