УДК 004.4, 004.9

**Е. Н. Боженкова** [1, 2], **А. Д. Воронков** [1], **Д. В. Иртегов** [1], **Е. Н. Конышева** [1, 2], **С. А. Черненок** [1], **Т. Г. Чурина** [1, 2]

[1] Новосибирский государственный университет
ул. Пирогова, 2, Новосибирск, 630090, Россия

[2] Институт систем информатики им. А. П. Ершова СО РАН
пр. Акад. Лаврентьева, 6, Новосибирск, 630090, Россия

E-mail: bozhenko@iis.nsk.su; alxn1101r@gmail.com; dmitry.irtegov@gmail.com; kate.konysheva@gmail.com; chernenoksergey@gmail.com; tanch@iis.nsk.su

# МОДЕЛЬ РАЗГРАНИЧЕНИЯ ПРАВ ДОСТУПА В СИСТЕМЕ АВТОМАТИЗИРОВАННОЙ ПРОВЕРКИ КОРРЕКТНОСТИ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ *

В статье рассматриваются особенности и специфика ролевой модели для обеспечения безопасности системы тестирования на уровне доступа. На основе теоретико-множественного подхода приводится формализованное описание ролевой модели безопасности системы тестирования NSUts. Показывается, что в системе тестирования управление доступом осуществляется на основе системных ролей, назначаемых каждому пользователю, и что системные роли корректны по отношению к заявленным организационным ролям. Представлена реализация ролевой модели в системе NSUts.

*Ключевые слова*: информационная безопасность, модели разграничения доступа, автоматизированная система тестирования, олимпиадное программирование.

## Введение

Для поддержки исследований по анализу, верификации и тестированию распределенных систем и их компонентов разрабатываются различные программные средства. Зачастую в существующих программных продуктах на основе анализа и оценки реального и ожидаемого поведения программы проверяется ее соответствие функциональным требованиям. Такие же нефункциональные требования, как производительность, объем используемой памяти являются одним из ключевых параметров качества программного обеспечения. Оптимизация производительности важна в системах реального времени, в серверных приложениях, системах управления промышленным оборудованием.

Одной из целей исследовательских работ, проводимых в рамках проекта Федеральной целевой программы развития образования на 2011–2015 гг., является разработка и реализация современных решений тестирования нефункциональных средств программных приложений.

* Работа выполняется в рамках мероприятий проекта «Подготовка и переподготовка профильных специалистов на базе центров образования и разработок в сфере информационных технологий в Сибирском и Дальневосточном федеральных округах» Федеральной целевой программы развития образования на 2011–2015 годы.

Для проверки действенности предлагаемых решений апробация проводится в пилотной версии системы автоматизированной проверки корректности программных приложений в условиях внешних ограничений на объем используемой памяти и производительности. Система проходит проверку в условиях проведения олимпиад по программированию.

Система автоматизированной проверки корректности программ NSUts [1], являясь публично доступной системой с большим числом пользователей, требует особо жесткой политики безопасности, поскольку дает возможность полного администрирования и хранения конфиденциальных данных пользователей. Ключевым моментом в реализации политики безопасности на этапе проектирования было создание адекватной формальной модели информационной безопасности, называемой также моделью разграничения прав доступа [2].

В результате исследований существующих подходов к управлению правами доступа в качестве основной модели была выбрана модель с ролевым разграничением доступа (*Role Based Access Control*, *RBAC*) [3]. Ролевое управление доступом обладает наибольшей гибкостью и является современным и эффективным механизмом защиты компьютерных систем. Особенности и специфика ролевой модели подходят для обеспечения безопасности системы тестирования на уровне доступа. Это действительно так, поскольку, с одной стороны, задание ролей позволяет определить более четкие и понятные для пользователей тестирующей системы правила разграничения доступа, точно соответствующие должностным обязанностям. С другой стороны, задание ролей позволяет реализовать гибкие, изменяющиеся динамически в процессе функционирования системы правила разграничения доступа.

Большинство существующих систем для проведения соревнований по программированию используют в том или ином виде ролевую модель для разграничения прав пользователей.

Некоторые системы (например, система Contester [4]) используют ролевую модель в простейшем виде, когда имеется только две роли: привилегированный («администратор») и непривилегированный пользователь («участник»). Эти роли соответствуют организационным ролям участника и администратора олимпиады. Набор привилегий ролей статичен, отсутствует возможность «тонкой» настройки доступа к определенной функциональности для отдельных пользователей.

Другие системы используют модель безопасности, похожую на модель, применяемую в системе NSUts. В них применяются функциональные роли для контроля доступа к той или иной функциональности. Организационные роли настраиваются через функциональные роли.

Для дальнейшего изложения нам потребуется понятие *тура* и *олимпиады*. В системе NSUts тур – ограниченное временными рамками соревнование с определенными правилами проведения и оценки, а также заданным подмножеством задач и тестов. Олимпиада – набор туров, объединенных единым списком пользователей.

В качестве примера рассмотрим систему Ejudge [5]. В системе Ejudge имеется 33 привилегии, контролирующих доступ к тем или иным функциям, таким как просмотр списка пользователей, создание и / или удаление пользователя, получение информации о пользователе.

В числе основных различий между системами привилегий NSUts и Ejudge можно выделить следующие.

1. Часть привилегий Ejudge вынесена на уровень тура в NSUts, это означает, что основные функции одинаково доступны или нет всем пользователям тура.
2. Часть привилегий Ejudge вынесены на уровень олимпиады, это означает, что существуют функции, одинаково доступные или нет всем пользователям олимпиады.
3. Часть привилегий Ejudge могут быть доступны или недоступны незарегистрированным пользователям системы, например просмотр списка участников или рейтинг.
4. Часть привилегий Ejudge объединена в одну привилегию системы NSUts.

Также стоит упомянуть системы, в которых не применяется система разграничения прав доступа. Такие системы (например, система Olympiads.ru [6]) предназначены для проверки собственных решений задач и не пригодны для проведения соревнований.

Таким образом, можно сделать вывод, что выбранная модель безопасности в системе NSUts является эффективной для решения проблемы разграничения доступа в системах для проведения соревнований по программированию.

**Описание системы**

Под объектами в системе тестирования будем понимать следующие функциональные элементы: олимпиада, тур, задача, условие задачи, решение, рейтинг, набор тестов, ответ тестирующей системы, очередь решений, список пользователей, атрибуты пользователей и другие.

Были проанализированы олимпиады различных уровней (Открытая Всесибирская олимпиада по программированию им. И. В. Поттосина, Всероссийская олимпиада школьников по информатике и прочие соревнования городского масштаба), в результате чего выделены следующие организационные роли (внешние роли) пользователей:

1) администраторы системы (а);
2) члены жюри (преподаватели):
   а) жюри-администратор (жа);
   б) жюри (авторы задач, решений) (ж);
   в) жюри-гость (жг);
   г) технический комитет (секретарь) (с);
3) участники соревнований (школьники, студенты) (у)
4) неавторизованные пользователи.

Применение таких организационных ролей на практике оказалось затруднительным. Были выявлены следующие их недостатки:

1) возможность пересечения друг с другом по функционалу;
2) охват не всех вариантов использования системы.

Несмотря на то, что использование выделенных организационных ролей покрывает базовые сценарии, существуют особые ситуации, когда такое деление по ролям является недостаточным. Поясним это на примере организационной роли «Участник соревнований». Одним из распространенных сценариев действий участника является редактирование своего профиля. Однако во время официальных соревнований изменять название команды не разрешается, чтобы не допустить путаницу в рейтинге и отчетах, например, для исключения совпадений, цензуры. Приведенные выше организационные роли не допускают такого сценария. Следовательно, необходимо разделить роль «Участник соревнований» на две функциональные роли, одна с правом редактирования профиля, другая без этого права. В дальнейшем по тем же причинам были выделены еще две функциональные роли: посылка решений и просмотр рейтинга. Введенные роли не пересекаются между собой, просты в реализации и использовании, являются достаточными и не требуют дальнейшей детализации.

Приведем описания некоторых сценариев, когда необходима тонкая настройка привилегий для членов жюри:

1) во время прогона тура членам жюри (авторам задач) может потребоваться решать задачи, управлять тестами, повторно протестировать свои решения, тогда как обычно они лишены таких прав;

2) во время тура может возникнуть необходимость запретить ряду членов жюри отвечать на вопросы, чтобы не было перекрытия ответов, но обычно это действие разрешено;

3) во время тура могут присутствовать приглашенные члены жюри, которым разрешено предоставлять всю информацию (рейтинг, тесты, очередь) только с правами на чтение, но не на редактирование;

4) возможность участия вне конкурса (сдача решения без отображения в рейтинге);

5) члены технического комитета могут распечатывать присланные решения, но не должны видеть никакой посторонней информации.

Поэтому для наличия возможности тонкой настройки доступа организационные роли не были выбраны в качестве статичного набора. Были введены системные или, иными словами, внутренние роли как совокупность прав доступа на определенную группу объектов.

Система тестирования предоставляет доступ к связанному набору функций над логически связанными наборами объектов, работа с которыми ведется определенной группой лиц. Например, системная роль «управление материалами олимпиады», относящаяся к организационной роли жюри-администратора, включает в себя работу с такими объектами, как тур, задача, условие.

Таким образом, в системе тестирования управление доступом осуществляется на основе системных ролей, назначаемых каждому пользователю. Организационные роли могут быть реализованы через механизм системных ролей.

**Формальное описание модели системы**

Формализованное описание ролевой модели безопасности системы тестирования осуществляется на основе теоретико-множественного подхода. Зададим следующие множества:

$U$ – множество пользователей;

$O$ – множество объектов системы;

$R_o$ – множество организационных ролей, $R_o = \{a, жа, ж, жг, c, y\}$;

$R_s$ – множество системных ролей, $R_s = \{sa, a, p, m, qa, n, s, ra, r, rg, av, st\}$;

$P$ – множество прав доступа на объекты системы тестирования.

Рассмотрим множество $R_o$, имеющее вид: $R_o = \{a, жа, ж, жг, c, y\}$. С помощью функции $PA_o : R_o \to 2^p$ определим множество прав доступа для каждой организационной роли системы. При этом для каждого права доступа $p \in P$ существует роль $r \in R_o$ такая, что $p \in PA_o(r)$.

Например, организационная роль жюри («*ж*») должна обладать следующими правами доступа: создание туров ($p_1$), удаление туров ($p_2$), редактирование настроек туров ($p_3$), настройка списка участников ($p_4$), выставление сдвигов ($p_5$), возможность повторного тестирования задач ($p_6$), редактирование тестов ($p_7$), раздача организационных привилегий ($p_8$), возможность сдавать задачи ($p_9$), возможность смотреть административный рейтинг ($p_{10}$), возможность смотреть очередь и статистику по туру ($p_{11}$), возможность смотреть тесты ($p_{12}$).

Поскольку организация ролевой модели безопасности в системе тестирования осуществляется на основе системных ролей, покажем, что выделение новых ролей корректно по отношению к организационным ролям.

Опишем далее модель разграничения доступа на основе выделенного множества системных ролей $R_s = \{sa, a, p, m, qa, n, s, ra, r, rg, av, st\}$. Определим функцию $PA_s : R_s \to 2^p$ как множество прав доступа для каждой системной роли системы (табл. 1). По построению для каждого права доступа $p \in P$ существует единственная роль $r \in R_s$ такая, что $p \in PA_s(r)$.

Так как в столбце «*P* (права доступа)» табл. 1 перечислены все права доступа, выделенные системные роли $R_s$ обеспечивают полное покрытие множества прав доступа *P*. Можно составить отображение организационных ролей в системные $F : R_o \to 2^{Rs}$ (табл. 2).

Из табл. 2 видно, что множество прав доступа, которыми должны обладать организационные роли, совпадает с множеством прав доступа сопоставленных им системных ролей, например, выделенные системные роли *a*, *m*, *s*, *ra*, *av*, *st* являются детализацией организационной роли «*ж*», причем $\bigcup \{PA_s(r) \mid r \in F(ж)\} = PA_O(ж)$.

Таким образом, выделенные системные роли полностью корректны по отношению к заявленным организационным ролям и могут быть использованы в качестве их замены.

**Реализация модели в системе**

Функция, определяющая для каждого пользователя системы тестирования множество ролей, на которые он может быть авторизован, имеет вид: $UA: U \to 2^{Rs}$.

Определим матрицу доступа [7] как матрицу *M*, строки которой соответствуют пользователям (субъектам), а столбцы – объектам. Каждая ячейка матрицы содержит набор прав, которые соответствующий субъект имеет по отношению к соответствующему объекту, т. е. выполнено $M[s,o] \subseteq \bigcup \{PA_s(r) \mid r \in UA(s)\} \subseteq P$, где $s \in U$ и $o \in O$ – конкретный субъект и объект соответственно.

Следует отметить, что множество пользователей *U* определяется отдельно для каждой олимпиады, являющейся своего рода обособленным доменом. Это позволяет пользователям иметь разные системные роли в разных олимпиадах и препятствует образованию значительного роста привилегий вышестоящих звеньев к объектам системы тестирования в целом, и не приводит к нарушению правила минимальных привилегий.

*Таблица 1*

Отображение $PA_s$ для системы тестирования

| *R* (название роли / обозначение) | *P* (права доступа) |
|---|---|
| Суперпользователь (*superadmin / s*) | Возможность изменять привилегии других пользователей, возможность создавать, удалять, редактировать олимпиады (открывать регистрацию, список участников, настраивать список полей для заполнения пользователями); генерация паролей; автоматическая регистрация пользователей |
| Администратор (*admin / a*) | Мета-привилегия |
| Распечатка (*print / p*) | Возможность распечатывать текст |
| Управление материалами олимпиады (*manage / m*) | Возможность создавать ($p_1$), удалять туры ($p_2$); редактировать настройки тура ($p_3$), настраивать видимый всем список участников ($p_4$); выставлять сдвиги ($p_5$), перетестировать задачи($p_6$), редактировать тесты ($p_7$); раздавать привилегии кроме привилегии «суперпользователь» ($p_8$) |
| Вопросы и ответы (*qna / qa*) | Возможность отвечать на вопросы, удалять вопросы, опубликовывать и закрывать ответы |
| Администрирование новостей (*news / n*) | Возможность добавлять, редактировать и удалять новости |
| Сдавать задачи (*send / s*) | Возможность сдавать задачи («Сдать») ($p_9$) |
| Административный рейтинг (*raitingAdmin / ra*) | Возможность смотреть административный рейтинг (после заморозки обновляется) ($p_{10}$) |
| Публичный рейтинг (*raiting / r*) | Возможность смотреть публичный рейтинг (до заморозки) |
| Редактирование информации (*registration / rg*) | Возможность редактировать информацию о себе, добавлять / редактировать / удалять вузы |
| Просмотр материалов олимпиады (*adminview / av*) | Возможность смотреть очередь, статистику по туру, делать выборку, делать «лапшу» ($p_{11}$) |
| Просмотр тестов (*showtests / st*) | Возможность смотреть тесты ($p_{12}$) |

*Таблица 2*

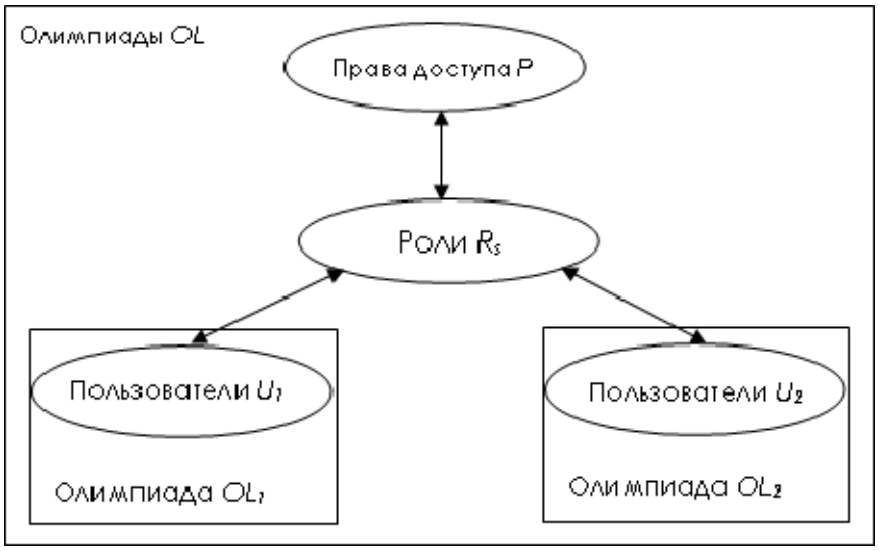
Отображение *F* организационных ролей в системные

| Организационная роль | Системная роль | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *sa* | *a* | *p* | *m* | *qa* | *n* | *s* | *ra* | *r* | *rg* | *av* | *st* |
| администратор (*а*) | X | X | X | X | X | X | X | X | X | X | X | X |
| жюри-админ (*жа*) | | X | | X | X | X | | X | | X | X | X |
| жюри (*ж*) | | X | | X | | | X | X | | | X | X |
| жюри-гость (*жг*) | | X | | | | | | X | | | X | X |
| секретарь (*с*) | | X | X | | | | | | | | | |
| участник (*у*) | | | | | | | X | | X | X | | |

Общая схема модели безопасности системы тестирования представлена на рисунке.

В данной модели предполагается, что множества $R_s$, $P$ и функция $PA_s$ не изменяются с течением времени. При этом существует единственная роль главного администратора олимпиады (системная роль *superadmin*), которая позволяет изменять множество *UA*.

Существуют также статические ограничения необходимого обладания ролью или правами доступа, накладываемые на данную модель. Для того чтобы пользователь был авторизован на некоторую роль, могут быть определены роли, на которые этот пользователь также должен




Отображение «многие ко многим» между множествами пользователей $U_1$, $U_2$, …, ролей $R_s$ и правами доступа $P$, $OL_i \subseteq OL \forall\ i$

быть авторизован. Например, для обладания ролью «управление материалами олимпиады» необходимо также наличие роли «администратор». Аналогично, для того чтобы роль обладала определенным правом доступа, могут быть определены другие права доступа, которыми данная роль также должна обладать.

Проверка прав доступа пользователей к определенным объектам в системе осуществляется по специально определенным правилам, не предусмотренным ролевой моделью. Эти правила учитывают привилегии субъекта в рамках олимпиады, к которой относится исследуемый объект, а также свойства этого объекта. Формально это можно описать при помощи предиката:

```
may_user_do_something (s, o)
        if ( p ∈ M[s, o] & conditions(...) ) then
            do_something();
        endif
end,
```

где $p \in P$ – права доступа; $M$ – матрица доступа; *conditions*() – состояние системы; *do_something*() – функция, выполняемая в случае успешной проверки прав доступа субъекта $s$ на объект $o$. При этом субъектом в данном случае является пользователь с заданной ролью, которая имеет определенный набор прав, содержащихся в матрице доступа. Например, для сдачи решения в систему пользователь должен обладать соответствующей привилегией «Сдавать задачи», а тур должен быть открытым и незавершенным.

**Заключение**

Модель безопасности системы тестирования базируется на ролевой модели разграничения доступа. Выделенные системные роли лишь частично связаны с организационными ролями пользователей системы. Показаны ситуации, когда выделенные организационные роли не покрывают реальных сценариев. Применяемая модель безопасности, возможно, обладает меньшей интуитивностью, однако она обеспечивает более гибкую настройку системы и реализует все практические сценарии.

## Список литературы

1. *Боженкова Е. Н., Иртегов Д. В., Киров А. В., Нестеренко Т. В., Чурина Т. Г.* Автоматизированная система тестирования NSUts: Требования и разработка прототипа // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2010. Т. 8, вып. 4. С. 46–53.

2. *Зегжда Д. П., Ивашко А. М.* Основы безопасности информационных систем. М.: Горячая линия – Телеком, 2000. 452 с.

3. *Девянин П. Н.* Модели безопасности компьютерных систем: Учеб. пособие для студ. высш. учеб. заведений. М.: Изд. центр «Академия», 2005. 144 с.

4. Система для проведения турниров и индивидуального решения задач по олимпиадному программированию Contester. URL: http://www.contester.ru/ (дата обращения 01.10.2011).

5. Документация к системе тестирования Ejudge. URL: http://www.ejudge.ru/download/ejudge-doc-20060304-pdf.tgz (дата обращения 01.10.2011).

6. Документация к системе самотестирования Olympiads.ru. URL: http://olympiads.ru/school/system/download/olympiads/tsystem-doc.rar (дата обращения 01.10.2011).

7. *Краснокутский А. В., Лепешкин О. М., Харечкин П. В.* Анализ функциональной применимости ролевой модели разграничения доступа в системах управления // Инфокоммуникационные технологии. 2007. Т. 5, № 3 С. 162–165.

*Материал поступил в редколлегию 15.11.2011*

**E. N. Bozhenkova, A. D. Voronkov, D. V. Irtegov, E. N. Konysheva,**
**S. A. Chernenok, T.G. Churina**

**MODEL ACCESS CONTROL IN SYSTEMS OF AUTOMATED TESTING OF SOFTWARE CORRECTNESS**

This article discusses design of role-based access control for security of automated testing system using set-theory formalism. Access control is based on system roles assigned to user accounts; it is demonstrated that these roles or their combinations are correct according to specified organizational roles. Implementation of this role system in NSUts is presented.

*Keywords*: information security, model access control, automated testing system, programming contests