УДК 004.94

УРОВЕНЬ ЗАПРЕЩАЮЩИХ РОЛЕЙ ИЕРАРХИЧЕСКОГО ПРЕДСТАВЛЕНИЯ МРОСЛ ДП-МОДЕЛИ

П. Н. Девянин

Федеральное учебно-методическое объединение в сфере высшего образования по УГСН 10.00.00 Информационная безопасность, г. Москва, Россия

Рассматривается построенный на основе базового уровня иерархического представления мандатной сущностно-ролевой ДП-модели управления доступом и информационными потоками в операционных системах семейства Linux (МРОСЛ ДП-модели) уровень запрещающих ролей. В рамках этого уровня при реализации ролевого управления доступом кроме ролей и административных ролей определяется порядок использования запрещающих ролей, права доступа которых не разрешают, а наоборот, запрещают получение субъект-сессиями соответствующих доступов к сущностям. Приводятся примеры изменений в условиях и результатах применения де-юре правил преобразования состояний заданной в рамках этого уровня МРОСЛ ДП-модели абстрактной системы и формулируется утверждение о корректности этих правил.

Ключевые слова: *компьютерная безопасность, ролевое управление доступом, запрещающая роль.*

DOI 10.17223/20710410/39/5

THE LEVEL OF NEGATIVE ROLES OF THE HIERARCHICAL REPRESENTATION OF MROSL DP-MODEL

P. N. Devyanin

Federal Educational and Methodological Association in Information Security, Moscow, Russia

E-mail: devyanin.peter@yandex.ru

MROSL DP-model is widely used as a mandatory entity-role model of access and information flows security control in Linux-type OS. To make the model to be more adequate for a number of special security features of the Russian OS Astra Linux Special Edition, it has been decided to extend MROSL DP-model by adding to it so called negative roles. In contrast to the ordinary roles, these ones contain access rights which prohibit entities or subject-sessions from getting some access. In this paper, an order of using negative roles in MROSL DP-model is defined, the corresponding changes of conditions and application results for state transformation de-jure rules in MROSL DP-model with negative roles are described, and the correctness of these modified rules are stated, namely: let G and G' be some states of MROSL DP-model with negative roles, G' be a result of transformation de-jure rules application to G, and G be satisfying all the conditions for mandatory role access control; then G' also satisfies all these conditions.

Keywords: *computer security, role-based access control, negative role.*

Введение

Ролевое управление доступом RBAC [1, 2] изначально являлось основой мандатной сущностно-ролевой ДП-модели управления доступом и информационными потоками в операционных системах семейства Linux (МРОСЛ ДП-модели) [2, 3]. При переходе к иерархическому представлению этой модели [4] именно ролевое управление доступом (без мандатного управления доступом и мандатного контроля целостности) задано на первом базовом уровне модели. На этом уровне по сложившейся традиции определены роли и административные роли, позволяющие субъект-сессиям соответственно получать через них права доступа к сущностям и другим субъект-сессиям и административные права доступа к ролям и административным ролям.

Однако практика использования МРОСЛ ДП-модели при реализации механизма управления доступом в отечественной защищенной операционной системе специального назначения (ОССН) Astra Linux Special Edition [5, 6] показала, что в ряде случаев применение только ролей, обладающих правами доступа к сущностям, разрешающим субъект-сессиям получение соответствующих доступов к ним, является недостаточным и создаёт неудобство при администрировании ОССН. В работе [7] описываются такие ситуации, когда, например, только одну учётную запись пользователя системы необходимо лишить права доступа, предоставляемого ей через роль, которой обладают все учётные записи пользователей системы, при этом все другие права доступа этой роли должны быть оставлены без изменений. Чтобы упростить администрирование ролевого управления доступом в подобных ситуациях, в этой же работе, а также в [8, 9] было предложено использование запрещающих ролей. Запрещающие роли, в отличие от «обычных», содержат права доступа к сущностям или субъект-сессиям, которые не разрешают, а наоборот, запрещают получение соответствующих доступов.

В то же время для анализа ролевого управления доступом с запрещающими ролями в [7] построена самостоятельная ДП-модель (мандатная сущностно-ролевая ДП-модель управления доступом в компьютерных сетях — МРОКС ДП-модель), элементы которой несколько отличаются от используемых в рамках МРОСЛ ДП-модели. Кроме того, в МРОКС ДП-модели запрещающие роли назначаются учётным записям пользователей через административные права доступа на чтение административных ролей. Такой подход может оказаться недостаточно удобным в реальной ОССН, так как, например, он не позволяет назначать явно запрещающие роли, соответствующие не только административным, но и «обычным» ролям.

В связи с этим в иерархическое представление МРОСЛ ДП-модели был добавлен описываемый в статье уровень запрещающих ролей. Этот уровень основан на базовом уровне модели, на нём не реализованы мандатное управление доступом и мандатный контроль целостности (это сделано автором на последующих уровнях модели), а для задания запрещающих ролей, в отличие от МРОКС ДП-модели, используется описанный ещё в ролевых моделях семейства RBAC механизм ограничений.

В п. 1 приводится описание элементов состояния рассматриваемой на уровне запрещающих ролей МРОСЛ ДП-модели абстрактной системы. В п. 2 задаётся порядок использования запрещающих ролей при реализации ролевого управления доступом. В п. 3 приводятся примеры изменений в условиях и результатах применения де-юре правил преобразования состояний системы и формулируется утверждение о корректности этих правил.

1. Состояние системы

С базового уровня МРОСЛ ДП-модели (его элементы и обозначения в целом соответствуют заданным в неиерархическом представлении МРОСЛ ДП-модели [2, 3]) на её уровне запрещающих ролей без изменений использованы следующие множества: U — учётных записей пользователей, S — субъект-сессий, O — объектов, C — контейнеров, E — сущностей, $NAMES$ — имён сущностей, R — ролей, AR — административных ролей, SAR — специальных административных ролей, R_r — видов прав доступа, R_a — видов доступа, P — прав доступа к сущностям и субъект-сессиям, A — доступов субъект-сессий к сущностям, G^* — всех возможных состояний системы, OP — правил преобразования состояний системы, U_ADMIN, U_ROLES — индивидуальных административных ролей и индивидуальных ролей учётных записей пользователей соответственно, $COMMON\_ROLES$ — общих ролей; функции: $user : S \to U$ — принадлежности субъект-сессии учётной записи пользователя, $entity\_name : C \times E \to 2^{NAMES}$ — имён сущностей, $H_E : E \to 2^E$ — иерархии сущностей, $H_S : S \to 2^S$ — иерархии субъект-сессий, $APA : AR \to 2^{AP}$ — административных прав доступа, $direct : E \cup R \cup AR \to \{\texttt{true}, \texttt{false}\}$ — вида метки; обозначения: $users\_admin\_role$, $entities\_admin\_role$, $subjects\_admin\_role$, $roles\_admin\_role$, $admin\_roles\_admin\_role$ — специальные административные роли, позволяющие администрировать учётные записи пользователей, сущности, субъект-сессии, роли или административные роли соответственно, u_admin, u_c, $common\_role$ — индивидуальные административные роли, роли и общие роли учётных записей пользователей, $G \vdash_{op} G'$ — переход системы из состояния G в состояние G' в результате применения правила op, $\Sigma(G^*, OP)$ — система, $\Sigma(G^*, OP, G_0)$ — система с начальным состоянием G_0.

Используем следующие дополнительные обозначения или переопределим некоторые обозначения базового уровня МРОСЛ ДП-модели:

- NR — множество запрещающих ролей, при этом по определению $AR \cap NR = R \cap NR = \varnothing$;
- $AP \subset (R \cup NR \cup AR) \times R_r$ — множество административных прав доступа к ролям или административным ролям (в данное множество добавлены права доступа к запрещающим ролям);
- $AA \subset S \times (R \cup NR \cup AR) \times R_a$ — множество доступов субъект-сессий к ролям, запрещающим ролям или административным ролям;
- $PA : R \cup NR \cup AR \to 2^P$ — функция прав доступа к сущностям ролей, запрещающих ролей и административных ролей, при этом для каждого права доступа $p \in P$ существует роль $r \in R \cup NR \cup AR$, такая, что выполняется условие $p \in PA(r)$;
- $shared\_container : C \cup R \cup NR \cup AR \to \{\texttt{true}, \texttt{false}\}$ — функция разделяемых контейнеров, такая, что сущность-контейнер, роль, запрещающая роль или административная роль $c \in C \cup R \cup NR \cup AR$ является разделяемой, когда $shared\_container(c) = \texttt{true}$, в противном случае $shared\_container(c) = \texttt{false}$;
- $role\_name : (R \cup NR \cup AR) \times (R \cup NR \cup AR) \to NAMES$ — функция имён ролей, запрещающих ролей и административных ролей в составе роли, запрещающей роли или административной роли (как контейнера);
- $H_R : R \cup NR \cup AR \to 2^R \cup 2^{NR} \cup 2^{AR}$ — функция иерархии ролей, запрещающих ролей или административных ролей, для задания которой по аналогии с базовым уровнем модели используется отношение частичного порядка $\leqslant$ на множестве $R \cup NR \cup AR$;
- $execute\_container : S \times C \times E \to \{\texttt{true}, \texttt{false}\}$ — функция доступа субъект-сессии к сущностям в контейнерах, такая, что по определению для субъект-

сессии $s \in S$, контейнера $c \in C$ и сущности $e \in E$ справедливо равенство $execute\_container(s, c, e) = \texttt{true}$ тогда и только тогда, когда существует последовательность сущностей $e_1, \ldots, e_n \in E$, где либо $n = 1$ и $c = e = e_1$, либо $n \geqslant 2$, $c = e_{n-1}$, $e = e_n$ и выполняются следующие условия:

У с л о в и е 1. Не существует сущности-контейнера $e_0 \in E$, такой, что $e_1 \in H_E(e_0)$ (без изменений по сравнению с базовым уровнем).

У с л о в и е 2. Верно $e_i \in H_E(e_{i-1})$, где $1 < i \leqslant n$ (без изменений по сравнению с базовым уровнем).

У с л о в и е 3. Существует роль или административная роль $r_i \in R \cup AR$, такая, что $(s, r_i, read_a) \in AA$ и $(e_i, execute_r) \in PA(r_i)$, где $1 \leqslant i < n$ (без изменений по сравнению с базовым уровнем).

У с л о в и е 4. Не существует запрещающей роли $nr \in NR$ и i, $1 \leqslant i < n$, таких, что $(s, nr, read_a) \in AA$ и $(e_i, execute_r) \in PA(nr)$.

Для обеспечения применения запрещающих ролей с использованием модели RBAC [1, 2] зададим механизм ограничений (*Constraints*) необходимого обладания запрещающей ролью с помощью следующей функции:

— $constraint_{NR} : R \cup AR \to 2^{NR}$ — функция необходимого обладания запрещающей ролью, задающая для каждой роли или административной роли множество запрещающих ролей, которые должна иметь субъект-сессия как текущие в случае обладания ею как текущей соответствующей ролью или административной ролью.

Пусть определена функция $constraint_{NR}$ необходимого обладания запрещающей ролью. Переопределим $G = (APA, PA, user, A, AA, H_R, H_E, H_S, constraint_{NR})$ — состояние системы.

2. Дополнительные условия реализации ролевого управления доступом с запрещающими ролями

Зададим порядок предоставления прав доступа и доступов к запрещающим ролям, основанный на порядке реализации ролевого управления доступом, определённом на базовом уровне МРОСЛ ДП-модели.

Предположение 1. В системе на уровне запрещающих ролей МРОСЛ ДП-модели выполняются следующие условия:

У с л о в и е 1 (создание, удаление, переименование запрещающих ролей, административные права доступа, иерархия и администрирование запрещающих ролей):

— все запрещающие роли являются «разделяемыми контейнерами»: для каждой запрещающей роли $nr \in NR$ справедливо равенство $shared\_container(nr) = \texttt{true}$;

— у каждой административной роли есть права доступа $execute_r$ ко всем запрещающим ролям: для каждой административной роли $ar \in AR$ и запрещающей роли $nr \in NR$ выполняется условие $(nr, execute_r) \in APA(ar)$;

— существует единственная специальная административная роль $negative\_roles\_admin\_role$, обладающая к каждой запрещающей роли административным правом доступа владения: для каждой запрещающей роли $nr \in NR$ выполняется условие $\{ar' \in AR : (nr, own_r) \in APA(ar')\} = \{negative\_roles\_admin\_role\}$, где $negative\_roles\_admin\_role \in SAR$;

— для создания, переименования или удаления запрещающей роли или «жёсткой» ссылки на неё в запрещающей роли субъект-сессии необходимо иметь к последней доступ на запись, текущую административную роль, обладающую к последней правом доступа на выполнение $execute_r$, и иметь административные досту-

пы на чтение и запись (кроме случая переименования) к административной роли $negative\_roles\_admin\_role$;

— для управления доступом к запрещающей роли, получения её параметров, числа «жёстких» ссылок к ней, множества административных ролей, обладающих к ней правами доступа, субъект-сессия должна иметь административный доступ на чтение к административной роли $negative\_roles\_admin\_role$, за исключением случаев, когда создаваемым административным ролям назначаются административные права доступа $execute_r$ ко всем запрещающим ролям.

У с л о в и е 2 (администрирование функции необходимого обладания запрещающей ролью, получение её значения):

— для специальных административных ролей из множества SAR не задаются запрещающие роли: для $ar \in SAR$ верно $constraint_{NR}(ar) = \varnothing$;

— для изменения значения функции необходимого обладания запрещающей ролью $constraint_{NR}$ субъект-сессия должна иметь административный доступ на чтение к административной роли $negative\_roles\_admin\_role$, а также в зависимости от того, является параметр функции ролью или административной ролью, субъект-сессия должна иметь административный доступ на чтение к административной роли $roles\_admin\_role$ или $admin\_roles\_admin\_role$ соответственно;

— при добавлении для некоторой роли или административной роли запрещающих ролей ни одна субъект-сессия не должна иметь к ней административных доступов;

— для получения субъект-сессией запрещающих ролей, заданных с помощью функции $constraint_{NR}$ для роли или административной роли, или, наоборот, ролей или административных ролей, для которых задана запрещающая роль, субъект-сессия должна иметь административный доступ на чтение к административной роли $negative\_roles\_admin\_role$.

У с л о в и е 3 (доступы и права доступа запрещающих ролей):

— к запрещающим ролям субъект-сессии могут иметь любые виды административных доступов из множества R_a;

— в случае, когда для некоторой роли или административной роли задано непустое множество запрещающих ролей, административный доступ на чтение субъект-сессии к такой роли или административной роли предоставляется только при получении ею административного доступа на чтение ко всем соответствующим запрещающим ролям: если существуют субъект-сессия $s \in S$, роль или административная роль $r \in R \cup AR$, запрещающая роль $nr \in NR$, такие, что $nr \in constraint_{NR}(r)$, $(s, r, read_a) \in AA$, то $(s, nr, read_a) \in AA$;

— запрещающие роли могут обладать к субъект-сессиям только правом доступа владения: для запрещающей роли $nr \in NR$ и субъект-сессии $s \in S$ если $(s, \alpha_r) \in \in PA(nr)$, то $\alpha_r = own_r$;

— запрещающие роли могут иметь к сущностям любые права доступа из множества R_r, только административные роли могут иметь к запрещающим ролям права доступа из множества R_r;

— для изменения прав доступа запрещающей роли субъект-сессии необходимо иметь к ней доступ на запись, за исключением случаев, когда удаляются сущности или субъект-сессии и соответственно удаляются имеющиеся к ним у запрещающих ролей права доступа.

У с л о в и е 4 (доступ к сущностям в иерархии сущностей):

— для получения субъект-сессией любого доступа к сущности, создания «жёсткой» ссылки на неё, получения или изменения параметров, прав доступа к ней, активизации из неё субъект-сессии требуется существование последовательности непосредственно вложенных друг в друга сущностей-контейнеров, начинающейся с некоторой сущности-«корневой контейнер» (например, корневой контейнер «/» в ОССН) и заканчивающейся сущностью-контейнером, в состав которой непосредственно входит сама сущность, и наличие у субъект-сессии текущих ролей или административных ролей, обладающих в совокупности правами доступа $execute_r$ ко всем сущностям-контейнерам этой последовательности, а также отсутствие у субъект-сессии запрещающей роли, обладающей правом доступа $execute_r$ хотя бы к одной сущности-контейнеру этой последовательности: для состояний системы G и G', правила преобразования состояний системы $op \in OP$, таких, что $G \vdash_{op} G'$, если $s \in S$, $e \in E$, $(s, e, \alpha_a) \notin A$ и $(s, e, \alpha_a) \in A'$, где $\alpha_a \in R_a$, то существует контейнер $c \in C$ и $execute\_container(s, c, e) = \texttt{true}$.

У с л о в и е 5 (создание, переименование, удаление сущности или «жёсткой» ссылки на неё, получение её параметров):

— при управлении доступом к сущности субъект-сессия не должна иметь текущую запрещающую роль, обладающую правом доступа владения к этой сущности;
— при создании, переименовании или удалении сущности или «жёсткой» ссылки на неё в сущности-контейнере субъект-сессия не должна иметь текущую запрещающую роль, обладающую правом доступа на выполнение к этой сущности-контейнеру;
— при изменении метки разделяемости сущности-контейнера в случае, когда субъект-сессия не имеет доступа на чтение к административной роли $entities\_admin\_role$, она не должна иметь текущую запрещающую роль, обладающую правом доступа владения к этой сущности-контейнеру;
— при получении субъект-сессией данных о ролях, запрещающих ролях или административных ролях, обладающих правами доступа к сущности, в случае, когда субъект-сессия не имеет доступа на чтение к административной роли $entities\_admin\_role$, она не должна иметь текущую запрещающую роль, обладающую правом доступа владения к этой сущности;
— при получении субъект-сессией данных о ролях или административных ролях, обладающих правом доступа владения к другой субъект-сессии или имеющихся у этой субъект-сессии текущих ролях или административных ролях, в случае, когда субъект-сессия не имеет доступа на чтение к административной роли $subjects\_admin\_role$, она не должна иметь текущую запрещающую роль, обладающую правом доступа владения ко второй субъект-сессии.

У с л о в и е 6 (создание и удаление субъект-сессии):

— при активизации субъект-сессией новой субъект-сессии из сущности первая субъект-сессия не должна иметь текущую запрещающую роль, обладающую правом доступа на выполнение к этой сущности;
— при удалении субъект-сессией субъект-сессии первая субъект-сессия не должна иметь текущую запрещающую роль, обладающую правом доступа владения ко второй субъект-сессии.

У с л о в и е 7 (вид метки):

— переопределяется функция $direct : E \cup R \cup NR \cup AR \to \{\texttt{true}, \texttt{false}\}$ и для каждой запрещающей роли вид её метки задаётся прямой: для $nr \in NR$ верно $direct(nr) = \texttt{true}$;

— для каждой сущности с косвенной меткой существует единственная старшая её в иерархии сущность-контейнер с прямой меткой, для которой все сущности, находящиеся ниже её в иерархии, имеют косвенные метки, а права доступа всех запрещающих ролей к ним равны правам доступа к этой сущности-контейнеру: для каждой сущности $e \in E$, такой, что $direct(e) = \texttt{false}$, существует единственная сущность-контейнер $c \in C$, такая, что $direct(c) = \texttt{true}$, $e < C$ и для любой сущности $e' \in E$, такой, что $e' < c$, верно $direct(e') = \texttt{false}$ и выполняется $(e', \alpha_r) \in PA(nr)$ тогда и только тогда, когда $(c, \alpha_r) \in PA(nr)$, где $nr \in NR$.

У с л о в и е 8 (доступы субъект-сессий к запрещающим ролям, соответствующим индивидуальным административным ролям, индивидуальным ролям или общим ролям):

— если для индивидуальной административной роли или индивидуальной роли учётной записи пользователя заданы запрещающие роли, то индивидуальная административная роль должна обладать административными правами доступа на чтение ко всем таким запрещающим ролям и при создании от имени этой учётной записи пользователя субъект-сессии она получает доступ на чтение ко всем соответствующим запрещающим ролям: если для учётной записи пользователя $u \in U$ выполняется $nr \in constraint_{NR}(u\_admin) \cup \cup constraint_{NR}(u\_c)$, то $(nr, read_r) \in APA(u\_admin)$; если для субъект-сессии $s \in S$ верно $nr \in constraint_{NR}(user(s)\_admin) \cup constraint_{NR}(user(s)\_c)$, то $(s, nr, read_a) \in \in AA$;

— если для общей роли заданы запрещающие роли, то индивидуальная административная роль каждой учётной записи пользователя должна обладать административными правами доступа на чтение ко всем таким запрещающим ролям и при создании любой субъект-сессии она получает доступ на чтение ко всем соответствующим запрещающим ролям: если выполняется $nr \in constraint_{NR}(common\_role)$, то для всех учётных записей пользователей $u \in U$ верно $(nr, read_r) \in APA(u\_admin)$ и для всех субъект-сессий $s \in S$ верно $(s, nr, read_a) \in AA$;

— при удалении субъект-сессии удаляются все её административные доступы к запрещающим ролям.

Рассмотрим особенности реализации в реальной защищённой ОССН условий сформулированного предположения.

Условие 1 содержит требования к запрещающим ролям и их иерархии, аналогичные требованиям к «обычным» ролям. Для создания, удаления, переименования, управления правами доступа запрещающих ролей, получения их параметров задаётся специальная административная роль $negative\_roles\_admin\_role$.

Условие 2 определяет порядок администрирования функции необходимого обладания запрещающей ролью $constraint_{NR}$, а также получения значений этой функции. При этом указывается на отсутствие запрещающих ролей, соответствующих специальным административным ролям, так как обратное едва ли потребуется в ОССН и может сильно затруднить её администрирование.

В условии 3 основным является описание порядка использования функции необходимого обладания запрещающей ролью $constraint_{NR}$, заключающегося в обязатель-

ном получении субъект-сессией запрещающей роли как текущей в случае, когда эта субъект-сессия получает как текущую соответствующую роль или административную роль.

В условии 4 дополнительно при переопределении функции доступа к сущностям в иерархии сущностей $execute\_container$ учитывается требование отсутствия даже одной текущей запрещающей роли, обладающей правом доступа на выполнение к сущности-контейнеру, через которую соответствующая субъект-сессия делает попытку получить какой-либо доступ к сущности.

Условия 5 и 6 раскрывают содержание ограничений, которые накладываются правами доступа («запрещающими» правами доступа) запрещающих ролей при попытке осуществления субъект-сессиями соответствующих доступов к сущностям, а также при создании или удалении субъект-сессий. При этом учитывается, что наличие запрещающих ролей не накладывает ограничений на использование специальных административных ролей $entities\_admin\_role$ и $subjects\_admin\_role$.

В условии 7 по аналогии с ролями и административными ролями определяется порядок задания прав доступа запрещающих ролей к сущностям, обладающим косвенной меткой.

Выполнение условия 8 обеспечивает возможность получения субъект-сессией заданной запрещающей роли как текущей в случае, если эта роль соответствует индивидуальной роли или индивидуальной административной роли учётной записи пользователя этой субъект-сессии. Правом доступа на чтение к такой запрещающей роли должна обладать индивидуальная административная роль учётной записи пользователя. В этом случае при создании субъект-сессии она сразу получает соответствующую запрещающую роль. Аналогично в случае, когда запрещающая роль задана для общей роли, то правом доступа на чтение к запрещающей роли должны обладать все учётные записи пользователей. Таким образом, условие обеспечивает согласованность задания значений функции необходимого обладания запрещающей ролью $constraint_{NR}$ с правами доступа индивидуальных административных ролей учётных записей пользователей. Кроме того, условие не указывает, какой административной роли должно быть дано право доступа на чтение к запрещающей роли, заданной для какой-то роли или административной роли. Это означает, что для того, чтобы получить эту роль или административную роль как текущую, субъект-сессии необходимо иметь право доступа на чтение к запрещающей роли через индивидуальную административную роль учётной записи пользователя субъект-сессии или через любую другую административную роль, которую как текущую должна предварительно получить субъект сессия. В противном случае субъект-сессия не сможет получить роль или административную роль как текущую.

В условии 8 описан способ задания запрещающей роли для индивидуальной административной роли или индивидуальной роли некоторой учётной записи пользователя, делающий такую запрещающую роль текущей для всех субъект-сессий, функционирующих от имени этой учётной записи пользователя. Возможны другие более «гибкие» способы применения запрещающих ролей. Например, в некоторых случаях в ОССН требуется ограничить права доступа субъект-сессии (процесса), активизированного из конкретной сущности (исполняемого файла). Тогда роли, которая обладает правом доступа на выполнение $execute_r$ к исполняемому файлу, можно с помощью функции $constraint_{NR}$ назначить соответствующую запрещающую роль. После этого родительский процесс должен сначала получить как текущую эту роль вместе с запрещающей

ролью и активизировать из исполняемого файла новый процесс, который «наследует» от родительского и запрещающую роль.

3. Де-юре правила преобразования состояний системы и их корректность

С учётом добавления в модель по сравнению с базовым уровнем запрещающих ролей модифицируются её де-юре правила преобразования состояний. В таблице приведены примеры таких правил, при этом для каждого правила в первой строке указаны его параметры, условия и результаты применения, наследованные с базового уровня без изменений, а во второй строке — внесённые в них дополнения и изменения, соответствующие уровню запрещающих ролей МРОСЛ ДП-модели.

Таблица

Примеры де-юре правил преобразования состояний уровня запрещающих ролей МРОСЛ ДП-модели

Параметры	Исходное состояние G	Результирующее состояние G'
$create\_user(x, u)$		
x, u	$x \in S$, $u \notin U$, $(x, users\_admin\_role, read_a) \in AA$, $(x, roles\_admin\_role, \alpha_a) \in AA$, $(x, admin\_roles\_admin\_role, \alpha_a) \in AA$, где $\alpha_a \in \{read_a, write_a\}$	$U' = U \cup \{u\}$, $AR' = AR \cup \{u\_admin\}$, $PA'(u\_admin) = \varnothing$, $direct'(u\_admin) =$ $= shared\_container'(u\_admin) =$ `true`, $role\_name'(u\_admin) =$ "u_admin", $H'_R(u\_admin) = \varnothing$, $R' = R \cup \{u\_c\}$, $PA'(u\_c) = \varnothing$, $direct'(u\_c) =$ $= shared\_container'(u\_c) =$ `true`, $role\_name'(u\_c) =$ "u_c", $H'_R(u\_c) = \varnothing$, $APA'(admin\_roles\_admin\_role) =$ $= APA(admin\_roles\_admin\_role) \cup$ $\cup \{(u\_admin, own_r)\}$, $APA'(roles\_admin\_role) =$ $= APA(roles\_admin\_role) \cup \{(u\_c, own_r)\}$, для $ar \in AR$ выполняется $APA'(ar) = APA(ar) \cup$ $\cup \{(u\_admin, execute_r), (u\_c, execute_r)\}$
—	—	$APA'(u\_admin) = \{(u\_admin, \alpha_r) :$ $\alpha_r \in \{read_r, write_r, execute_r\}\} \cup$ $\cup \{(u\_c, \alpha_r), (common\_role, \alpha_r) :$ $\alpha_r \in \{read_r, write_r, execute_r\}\} \cup$ $\cup \{(r, execute_r) : r \in R' \cup NR \cup AR'\} \cup$ $\cup \{(nr, read_r) :$ $nr \in constraint_{NR}(common\_role)\}\}$
$access\_write(x, y)$		
x, y	$x \in S$, существует $r \in R \cup AR :$ $(x, r, read_a) \in AA$, [если $y \in E$, то $(y, write_r) \in PA(r)$, существует контейнер $c \in C$, такой, что $execute\_container(x, c, y) =$ `true`], [если $y \in R \cup AR$, то $(y, write_r) \in APA(r)$]	если $y \in E$, то $A' = A \cup \{(x, y, write_a)\}$, $AA' = AA$, если $y \in R \cup AR$, то $AA' = AA \cup \{(x, y, write_a)\}$, $A' = A$
—	$y \in E \cup R \cup NR \cup AR$, [если $y \in E$, то не существует запрещающей роли $nr \in NR$, такой, что $(x, nr, read_a) \in AA$ и $(y, write_r) \in PA(nr)$], [если $y \in NR$, то $(y, write_r) \in APA(r)$]	—

Продолжение таблицы

Параметры	Исходное состояние G	Результирующее состояние G'
$add\_negative\_owner(x, nr, y)$		
—	—	—
x, nr, y	$x \in S$, $y \in E \cup S$, $(x, nr, write_a) \in AA$, [существует $r' \in R \cup AR$: $(x, r', read_a) \in AA$, $(y, own_r) \in PA(r')$], [не существует запрещающей роли $nr' \in NR$, такой, что $(x, nr', read_a) \in \in AA$ и $(y, own_r) \in PA(nr')$], [если $y \in E$, то $direct(y) =$ `true`, существует контейнер $c \in C$, такой, что $execute\_container(x, c, y) =$ `true`]	$PA'(nr) = PA(nr) \cup \{(y, own_r)\}$, если $y \in E$ и $H_E(y) = \{y' \in H_E(y) :$ $direct(y') =$ `false`$\}$, то для всех $y' < y$ верно $PA'(r) = PA(r) \cup \{(y', own_r)\}$
$add\_negative\_role(x, r, nr)$		
—	—	—
x, r, nr	$x \in S$, $r \in (R \cup AR) \setminus SAR$, $nr \in NR$, [не существует $s \in S$, такого, что $(s, r, read_a) \in AA$], $(x, negative\_roles\_admin\_role, read_a) \in AA$, [если $r \in R$, то $(x, roles\_admin\_role, read_a) \in AA$, если $r \in AR$, то $(x, admin\_roles\_admin\_role, read_a) \in AA$], [если существует $u \in U$, такая, что $r \in \{u\_admin, u\_c\}$, то $(nr, read_r) \in APA(u\_admin)$], [если $r = common\_role$, то для всех $u \in U$ $((nr, read_r) \in APA(u\_admin))$]	$constraint'_{NR}(r) = constraint_{NR}(r) \cup \{nr\}$
$create\_hard\_link(x, y, name, z)$		
$x, y, name, z$	$x \in S$, $y \in O$, $z \in C$, [существует $c \in C$: $execute\_container(x, c, y) =$ `true`], [$(x, z, write_a) \in A$, существует $r \in R \cup AR$, такая, что $(x, r, read_a) \in \in AA$ и $(z, execute_r) \in PA(r)$], $name \in \in NAME \setminus (\{\text{“”}\} \cup \{entity\_name(z, y'):$ $y' \in H_E(z)\})$, $H_E(z) = \{y' \in H_E(z):$ $direct(y') = direct(y)\}$, [если $direct(y) =$ `true`, то $direct(z) =$ `true`], [если $direct(y) =$ `false`, то существует $z' \in C$, такая, что $direct(z) =$ `true`, $y < z'$, $z \leqslant z'$ и для всех $y' < z'$ верно $direct(y') =$ `false`]	$entity\_name'(z, y) = entity\_name(z, y) \cup$ $\cup \{name\}$, $H'_E(z) = H_E(z) \cup \{y\}$
—	не существует запрещающей роли $nr \in \in NR$, такой, что $(x, nr, read_a) \in AA$ и $(z, execute_r) \in PA(nr)$	—
$rename\_role(x, ry, name)$		
$x, ry, name$	$x \in S$, $name \in NAME \setminus (\{\text{“”}\} \cup$ $\cup \{role\_name(rz, ry') : ry', rz \in R \cup AR,$ $ry' \in H_R(rz)\})$, [если $ry \in R$, то $(x, roles\_admin\_role, read_a) \in AA$, если $ry \in AR$, то $(x,$ $admin\_roles\_admin\_role, read_a) \in AA$]	для $rz \in R \cup AR$, таких, что $ry \in H_R(rz)$, выполняется $role\_name'(rz, ry) = name$
—	$ry \in (R \cup NR \cup AR) \setminus (U\_ADMIN \cup$ $\cup U\_ROLES \cup COMMON\_ROLES \cup$ $\cup SAR)$, $name \notin \{role\_name(nr', nr'') :$ $nr', nr'' \in NR\}$, [если $ry \in NR$, то $(x,$ $negative\_roles\_admin\_role, read_a) \in AA$], [для $rz \in R \cup NR \cup AR : ry \in H_R(rz)$, выполняется $(x, rz, write_a) \in AA$]	для $rz \in NR$, таких, что $ry \in H_R(rz)$, выполняется $role\_name'(rz, ry) = name$

Окончание таблицы

Параметры	Исходное состояние G	Результирующее состояние G'
$create\_first\_session(x, u, y, z)$		
x, u, y, z	$x \in S$, $u \in U$, $y \in E$, $z \notin S$, существует $r \in R \cup AR$, такая, что $(x, r, read_a) \in$ $\in AA$ и $(y, execute_r) \in PA(r)$, существует контейнер $c \in C$, такой, что $execute\_container(x, c, y) =$ `true`	$S' = S \cup \{z\}$, $user'(z) = u$, $H'_S(z) = \varnothing$, $PA'(u\_c) = PA(u\_c) \cup \{(z, own_r)\}$
—	[не существует запрещающей роли $nr \in NR$: $(x, nr, read_a) \in AA$ и $(y, execute_r) \in PA(nr)$], [для всех $nr \in constraint_{NR}(u\_admin) \cup$ $\cup\, constraint_{NR}(u\_c) \cup$ $\cup\, constraint_{NR}(common\_role)$ верно $(nr, read_r) \in APA(u\_admin)$]	$AA' = AA \cup \{(z, u\_admin, read_a)$, $(z, u\_c, write_a)$, $(z, common\_role, write_a)$, $(z, u\_c, read_a)$, $(z, common\_role, read_a)\} \cup$ $\cup \{(z, nr, read_a)$: $nr \in constraint_{NR}(u\_admin) \cup$ $\cup\, constraint_{NR}(u\_c) \cup$ $\cup\, constraint_{NR}(common\_role)\}$

Всего в рамках уровня запрещающих ролей иерархического представления МРОСЛ ДП-модели задано 35 де-юре правил преобразования состояний, из которых четыре правила вида $add\_negative\_owner(x, nr, y)$, $remove\_negative\_owner(x, nr, y)$, $add\_negative\_role(x, r, nr)$ и $remove\_negative\_role(x, r, nr)$ отсутствовали на базовом уровне, так как предназначены для добавления или удаления запрещающим ролям права доступа владения к сущностям или субъект-сессиям, а также для управления множествами запрещающих ролей и административных ролей, задаваемых функцией $constraint_{NR}$. При этом так же, как на базовом уровне, на текущем уровне не анализируются информационные потоки, поэтому на нём также не задаются де-факто правила преобразования состояний.

Рассмотрим приведённые в таблице примеры де-юре правил преобразования состояний.

В де-юре правило вида $create\_user(x, u)$ добавляется только требование, чтобы индивидуальной административной роли создаваемой учётной записи пользователя были даны административные права доступа на чтение ко всем запрещающих ролям, заданным для общей роли.

В де-юре правило вида $access\_write(x, y)$ включены дополнительные условия, позволяющие субъект-сессиям получать доступ на запись к запрещающим ролям. Кроме того, в этом правиле, как во многих других де-юре правилах, используется переопределённая функция доступа субъект-сессии к сущностям в контейнерах $execute\_container$, учитывающая необходимость проверки отсутствия у субъект-сессии x запрещающих ролей, обладающих правами доступа на выполнение к соответствующим сущностям-контейнерам, в которых содержится сущность y. Требуется также отсутствие у субъект-сессии x текущей запрещающей роли, обладающей правом доступа на запись.

Де-юре правило вида $add\_negative\_owner(x, nr, y)$ является новым и во многом аналогичным правилам вида $grant\_rights(x, r, \{(y, \alpha_{rj}) : 1 \leqslant j \leqslant k\})$, $set\_entity\_owner(x, r, r', y)$ и $set\_subject\_owner(x, r, r', y)$. Оно позволяет субъект-сессии x, обладающей ролью или административной ролью, имеющей право доступа владения к сущности или субъект-сессии y, а также не обладающей запрещающей ролью, имеющей это право доступа к y, добавить запрещающей роли nr (к которой субъект-сессия x должна иметь административный доступ на запись) право доступа владения к y. При этом запрещающая роль не становится ролью-«владельцем» или ролью-«антивладельцем» сущности или субъект-сессии y, допускается, что запрещаю-

щих ролей, обладающих правом доступа владения к сущности или субъект-сессии, может быть несколько. С этим связано добавление этого правила на рассматриваемом уровне модели и неиспользование для этого правил вида $set\_entity\_owner(x, r, r', y)$ и $set\_subject\_owner(x, r, r', y)$.

Де-юре правило вида $add\_negative\_role(x, r, nr)$ также является новым и позволяет субъект-сессии x добавить для роли или административной роли r запрещающую роль nr. Для этого субъект-сессия x должна обладать текущей специальной административной ролью $negative\_roles\_admin\_role$, а также соответственно либо специальной административной ролью $roles\_admin\_role$, либо $admin\_roles\_admin\_role$. При этом нельзя задавать запрещающие роли для любых специальных административных ролей из множества SAR и для упрощения администрирования системы ни одна субъект-сессия не должна иметь к роли или административной роли r административных доступов на чтение.

В де-юре правиле вида $create\_hard\_link(x, y, name, z)$ добавлена проверка отсутствия у субъект-сессии x текущей запрещающей роли, обладающей правом доступа на выполнение к сущности-контейнеру z, в которой создаётся «жёсткая» ссылка на сущность y.

Переименование запрещающих ролей осуществляется аналогично «обычным» ролям, поэтому дополнительным условием де-юре правила вида $rename\_role(x, ry, name)$ является наличие у субъект-сессии x административного доступа на чтение к специальной административной роли $negative\_roles\_admin\_role$. Требуется также исключение совпадения нового имени $name$ с именем какой-либо запрещающей роли.

В де-юре правиле вида $create\_first\_session(x, u, y, z)$ дополнительно требуется отсутствие у субъект-сессии x текущей запрещающей роли, обладающей правом доступа на выполнение к сущности y, из которой активизируется новая субъект-сессия. Кроме того, чтобы создаваемая субъект-сессия z могла получить в качестве текущих все запрещающие роли для индивидуальной административной роли, индивидуальной роли её учётной записи пользователя и для общей роли, проверяется наличие права доступа на чтение ко всем таким запрещающим ролям у индивидуальной административной роли учётной записи пользователя субъект-сессии z. После применения правила созданная субъект-сессия z получает административный доступ на чтение ко всем этим запрещающим ролям.

На уровне запрещающих ролей МРОСЛ ДП-модели верно утверждение о том, что условия и результаты правил преобразования состояний системы заданы корректно, т. е. соответствуют предположениям базового и текущего (предположение 1) уровней модели.

Утверждение 1. Пусть G_0 — начальное состояние системы $\Sigma(G^*, OP, G_0)$, удовлетворяющее условиям предположений, задающих порядок реализации ролевого управления доступом на базовом уровне и уровне запрещающих ролей (предположение 1) МРОСЛ ДП-модели. Тогда на любой траектории $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$, где $N \geqslant 1$, состояние G_N и переход $G_{N-1} \vdash_{op_N} G_N$ удовлетворяют условиям этих предположений.

Заключение

Таким образом, в МРОСЛ ДП-модель добавлены запрещающие роли, применение которых в ОССН позволит в перспективе упростить настройку и администрирование реализуемого в ней ролевого управления доступом. Кроме того, вместе с уровнем мандатного контроля целостности с невырожденной решёткой уровней целостности [10]

текущий уровень модели стал основой для её дальнейшего развития и построения нескольких новых уровней МРОСЛ ДП-модели, включающих также мандатное управление доступом с контролем информационных потоков по памяти и по времени.

ЛИТЕРАТУРА

1. *Sandhu R.* Role-Based Access Control. Advanced in Computers. Academic Press, 1998. V. 46.
2. *Девянин П. Н.* Модели безопасности компьютерных систем. Управление доступом и информационными потоками: учеб. пособие для вузов. 2-е изд., испр. и доп. М.: Горячая линия — Телеком, 2013. 338 с.
3. *Буренин П. В., Девянин П. Н., Лебеденко Е. В. и др.* Безопасность операционной системы специального назначения Astra Linux Special Edition: учеб. пособие для вузов / под ред. П. Н. Девянина. 2-е изд., стереотип. М.: Горячая линия — Телеком, 2016. 312 с.
4. *Девянин П. Н.* О результатах формирования иерархического представления МРОСЛ ДП-модели // Прикладная дискретная математика. Приложение. 2016. № 9. С. 83–87.
5. `www.astralinux.com` — Операционные системы Astra Linux. 2017.
6. `ru.wikipedia.org/wiki/Astra_Linux` — Astra Linux. 2017.
7. *Тележников В. Ю.* Правила преобразования состояний системы в рамках ДП-модели управления доступом в компьютерных сетях, построенных на основе ОС семейства Linux // Прикладная дискретная математика. 2016. № 1(31). С. 67–85.
8. *Armando A. and Ranise S.* Automated and efficient analysis of role-based access control with attributes // LNCS. 2012. V. 7371. P. 25–40.
9. *Kuijper W. and Ermolaev V.* Sorting out role based access control // Proc. SACMAT'14. N. Y.: ACM, 2014. P. 63–74.
10. *Девянин П. Н.* Реализация невырожденной решётки уровней целостности в рамках иерархического представления МРОСЛ ДП-модели // Прикладная дискретная математика. Приложение. 2017. № 10. С. 111–114.

REFERENCES

1. *Sandhu R.* Role-Based Access Control. Advanced in Computers, Academic Press, 1998, vol. 46.
2. *Devyanin P. N.* Modeli bezopasnosti kompyuternyh sistem. Upravlenie dostupom i informacionnymi potokami [Security Models of Computer Systems. Access Control and Information Flows]. Moscow, Goryachaya liniya — Telekom, 2013. 338 p. (in Russian)
3. *Burenin P. V., Devyanin P. N., Lebedenko E. V., et al.* Bezopasnost' operacionnoy sistemy special'nogo naznacheniya Astra Linux Special Edition [Security of Operating System Astra Linux Special Edition]. Moscow, Goryachaya liniya — Telekom, 2016. 312 p. (in Russian)
4. *Devyanin P. N.* O rezultatah formirovaniya ierarhiceskogo predstavleniya MROSL DP-modeli [About results of design hierarchical representation of MROSL DP-model]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2016, no. 9, pp. 83–87. (in Russian)
5. `www.astralinux.com` — Operating System Astra Linux, 2017.
6. `ru.wikipedia.org/wiki/Astra_Linux` — Astra Linux, 2017.
7. *Telezhnikov V. Y.* Pravila preobrazovaniya sostoyaniy sistemy v ramkah DP-modeli upravleniya dostupom v komp'yuternyh setyah, postroennyh na osnove OS semeystva Linux [System state transformation rules in DP-model of access control in computer networks based on operating systems of Linux]. Prikladnaya Diskretnaya Matematika, 2016, no. 1(31), pp. 67–85. (in Russian)
8. *Armando A. and Ranise S.* Automated and efficient analysis of role-based access control with attributes. LNCS, 2012, vol. 7371, pp. 25–40.

9. *Kuijper W. and Ermolaev V.* Sorting out role based access control. Proc. SACMAT'14, New York, ACM, 2014, pp. 63–74.
10. *Devyanin P. N.* Realizaciya nevyrozhdennoy reshyotki urovney celostnosti v ramkah ierarhiceskogo predstavleniya MROSL DP-modeli [Implementation of a non-degenerate lattice integrity levels within the hierarchical representation of MROSL DP-model]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2017, no. 10, pp. 111–114. (in Russian)