МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

DOI 10.17223/20710410/22/3

УДК 004.94

АДМИНИСТРИРОВАНИЕ СИСТЕМЫ В РАМКАХ МАНДАТНОЙ СУЩНОСТНО-РОЛЕВОЙ ДП-МОДЕЛИ УПРАВЛЕНИЯ ДОСТУПОМ И ИНФОРМАЦИОННЫМИ ПОТОКАМИ В ОС СЕМЕЙСТВА LINUX

П. Н. Девянин

Учебно-методическое объединение по образованию в области информационной безопасности, г. Москва, Россия

E-mail: peter_devyanin@hotmail.com

Рассматриваются де-юре правила администрирования параметров механизма управления доступом системы, заданной в рамках мандатной сущностно-ролевой ДП-модели и ориентированной на реализацию в отечественной защищённой операционной системе *Astra Linux Special Edition*. Данные правила позволяют администрировать учётные записи пользователей, права доступа, множество ролей, иерархию ролей и административных ролей, уровни конфиденциальности и целостности сущностей, субъект-сессий, ролей и административных ролей.

Ключевые слова: *мандатная сущностно-ролевая ДП-модель, операционная система Linux.*

Введение

При разработке формальных моделей безопасности управления доступом и информационными потоками часто правилам преобразования состояний системы, задающим порядок её администрирования, уделяется второстепенное внимание. Это объясняется ориентированностью большинства таких моделей только на теоретический анализ условий безопасности системы, при осуществлении которого, как правило, предполагается, что все действия по администрированию системы выполнены её доверенными субъект-сессиями (субъектами) и функционирование системы осуществляется только в результате реализации правил, инициированных недоверенными субъект-сессиями. Однако в разрабатываемой в настоящее время автором мандатной сущностно-ролевой ДП-модели операционных систем (ОС) семейства Linux (МРОСЛ ДП-модели) [1 – 3] значительное внимание уделяется её адаптации к условиям функционирования механизма управления доступом отечественной защищённой операционной системы *Astra Linux Special Edition* [4]. Для этого в МРОСЛ ДП-модели детально задаётся порядок реализации её элементов непосредственно в коде программного обеспечения защищённой ОС. Следовательно, целесообразно, кроме типовых правил управления доступом (получения доступа к сущностям, создания, удаления, переименования сущностей или «жёстких» ссылок на них, создания или удаления субъект-сессий), чётко определить правила, позволяющие администрировать учётные записи пользователей, административные права доступа, множества ролей, иерархию ролей и административных ролей, уровни конфиденциальности и целостности сущностей, субъект-сессий, ролей и административных ролей.

1. Основные элементы МРОСЛ ДП-модели

Так как формирование МРОСЛ ДП-модели ещё не окончено и её элементы, определения, предположения корректируются с учётом опыта их практической реализации в защищённой ОС, приведём актуальное описание модели в объёме, достаточном для рассмотрения правил администрирования системы.

Предположение 1. В рамках МРОСЛ ДП-модели пользователям соответствуют их учётные записи. Каждая учётная запись пользователя является учётной записью либо доверенного, либо недоверенного пользователя. Каждая субъект-сессия является либо доверенной, либо недоверенной и функционирует от имени учётной записи доверенного или недоверенного пользователя соответственно.

Используем следующие обозначения:

— $E = O \cup C$ — множество сущностей, где O — множество объектов; C — множество контейнеров; $O \cap C = \varnothing$;
— S — множество субъект-сессий учётных записей пользователей, где $S \cap E = \varnothing$;
— $entity\_name : C \times E \to NAMES$ — функция имён сущностей в составе сущностей-контейнеров, где $NAMES$ — некоторое множество допустимых имён сущностей, включающее элемент «» — пустая строка. При этом, по определению, если $e \notin H_E(c)$, то $entity\_name(c, e) =$ «», иначе $entity\_name(c, e) \neq$ «».

Определение 1. Иерархией сущностей называется заданное на множестве сущностей E отношение частичного порядка $\leqslant$, удовлетворяющее условию: если для сущности-контейнера $e \in C$ существуют сущности $e_1, e_2 \in C$, такие, что $e \leqslant e_2$, $e \leqslant e_1$, то $e_1 \leqslant e_2$ или $e_2 \leqslant e_1$. В случае, когда для двух сущностей $e_1, e_2 \in E$ выполняются условия $e_1 \leqslant e_2$ и $e_1 \neq e_2$, будем говорить, что сущность e_1 содержится в сущности-контейнере e_2, и будем использовать обозначение $e_1 < e_2$. Определим $H_E : E \to 2^E$ — функцию иерархии сущностей (сопоставляющую каждой сущности $e \in E$ множество сущностей $H_E(e) \subset E$, непосредственно в ней содержащихся), удовлетворяющую следующим условиям:

— если $e \in H_E(c)$, то $e < c$, при этом если $e \in C$, то не существует сущности-контейнера $d \in C$, такой, что $e < d < c$;
— для любых сущностей $e_1, e_2 \in E$, $e_1 \neq e_2$, выполняется равенство $H_E(e_1) \cap H_E(e_2) \cap \cap C = \varnothing$;
— если $o \in O$, то справедливо равенство $H_E(o) = \varnothing$.

Определение 2. Иерархией субъект-сессий называется заданное на множестве сущностей S отношение частичного порядка $\leqslant$, удовлетворяющее условию: если для субъект-сессии $s \in S$ существуют субъект-сессии $s_1, s_2 \in S$, такие, что $s \leqslant s_2$, $s \leqslant s_1$, то $s_1 \leqslant s_2$ или $s_2 \leqslant s_1$. В случае, когда для двух субъект-сессий $s_1, s_2 \in S$ выполняются условия $s_1 \leqslant s_2$ и $s_1 \neq s_2$, будем говорить, что субъект-сессия s_1 является потомком s_2, и будем использовать обозначение $s_1 < s_2$. Определим $H_S : S \to 2^S$ — функцию иерархии субъект-сессий (сопоставляющую каждой субъект-сессии $s \in S$ множество субъект-сессий $H_S(s) \subset S$, непосредственно в ней содержащихся), удовлетворяющую условиям:

— если $s_1 \in H_S(s_2)$, то $s_1 < s_2$ и не существует субъект-сессии $s \in S$, такой, что $s_1 < s < s_2$;
— для любых субъект-сессий $s_1, s_2 \in S$, $s_1 \neq s_2$, выполняется равенство $H_S(s_1) \cap \cap H_S(s_2) = \varnothing$.

В рамках МРОСЛ ДП-модели учтено наличие механизма создания «жёстких» ссылок (*hard link*) в файловой системе ОС семейства Linux, обеспечивающего возможность

размещения сущностей-объектов одновременно в нескольких сущностях-контейнерах. Для всех сущностей заданы их имена в составе сущностей-контейнеров, что позволяет описать правила предоставления содержимого контейнеров (например, списков файлов и каталогов, выдаваемых в реальной защищённой ОС по команде `ls`) с учётом параметров мандатного управления доступом. Впервые, по сравнению с предыдущими ДП-моделями, определено, что множество субъект-сессий не входит в множество сущностей. Это связано с тем, что в реальной защищённой ОС функции управления доступом к субъект-сессиям (процессам) и сущностям (файлам, каталогам) реализуются раздельно. Таким образом, иерархия на множестве субъект-сессий определяется независимо от иерархии сущностей и при реализации в ОС может быть задана двумя способами. Первый способ, когда существует явная связь между родительскими субъект-сессиями и их потомками (например, когда при завершении работы родительской субъект-сессии завершают работу её субъект-сессии — потомки, подчинённые родительской). Второй способ, когда порождённая субъект-сессией другая субъект-сессия функционирует независимо, в этом случае подчинение её в иерархии родительской субъект-сессии нецелесообразно.

Также определим:

— U — множество учётных записей пользователей;
— L_U — множество учётных записей доверенных пользователей;
— N_U — множество учётных записей недоверенных пользователей, при этом по определению справедливы равенства $L_U \cup N_U = U$, $L_U \cap N_U = \varnothing$;
— L_S — множество доверенных субъект-сессий;
— N_S — множество недоверенных субъект-сессий, при этом по определению справедливы равенства $L_S \cup N_S = S$, $L_S \cap N_S = \varnothing$;
— $user : S \to U$ — функция принадлежности субъект-сессии учётной записи пользователя, задающая для каждой субъект-сессии учётную запись пользователя, от имени которой она активизирована;
— R — множество ролей;
— AR — множество административных ролей, при этом по определению $AR \cap R = \varnothing$ (административные роли — особый вид ролей, предназначенный для изменения множеств прав доступа ролей, авторизации на роли, а также выполнения функций по администрированию системы, например, управления мандатными уровнями конфиденциальности сущностей и субъект-сессий);
— $R_r = \{read_r, write_r, execute_r, own_r\}$ — множество видов прав доступа;
— $R_a = \{read_a, write_a\}$ — множество видов доступа;
— $R_f = \{write_m, write_t\}$ — множество видов информационных потоков (по памяти и по времени).

Поскольку традиционно в ОС семейства Linux рассматриваются три вида прав доступа к сущностям: на чтение, на запись и на выполнение (или на использование контейнера-каталога), а также при управлении доступом к сущностям и субъект-сессиям учитывается наличие у каждой из них уникального владельца, имеющего право передавать права доступа к ним другим учётным записям пользователей, то в рамках МРОСЛ ДП-модели используются виды прав доступа $read_r$, $write_r$, $execute_r$, own_r и виды доступов $read_a$ и $write_a$. В отличие от предыдущих ДП-моделей, доступ владения own_a исключён из рассмотрения, как не имеющий явной реализации в реальных ОС.

Далее введём следующие обозначения:

- $P \subset (E \times R_r) \cup (S \times \{own_r\})$ — множество прав доступа к сущностям и субъект-сессиям;
- $AP \subset (R \cup AR) \times R_r$ — множество административных прав доступа к ролям или административным ролям;
- $A \subset S \times E \times R_a$ — множество доступов субъект-сессий к сущностям;
- $AA \subset S \times (R \cup AR) \times R_a$ — множество доступов субъект-сессий к ролям или административным ролям;
- $F \subset (E \cup S \cup R \cup AR) \times (E \cup S \cup R \cup AR) \times R_f$ — множество информационных потоков;
- $PA : R \cup AR \to 2^P$ — функция прав доступа к сущностям ролей и административных ролей, при этом для каждого права доступа $p \in P$ существует роль $r \in R \cup AR$, такая, что выполняется условие $p \in PA(r)$;
- $APA : AR \to 2^{AP}$ — функция административных прав доступа к ролям и административным ролям административных ролей, при этом для каждого административного права доступа $ap \in AP$ существует административная роль $ar \in AR$, такая, что выполняется условие $ap \in APA(ar)$;
- $shared\_container : C \cup R \cup AR \to \{\texttt{true}, \texttt{false}\}$ — функция разделяемых контейнеров, такая, что сущность-контейнер, роль или административная роль $c \in C \cup R \cup AR$ является разделяемой, когда $shared\_container(c) = \texttt{true}$, в противном случае $shared\_container(c) = \texttt{false}$;
- $V : E \cup R \cup AR \to \{(a_1, \ldots, a_n) : a_i \in \{0, 1\}, 1 \leqslant i \leqslant n, n \geqslant 1\} \cup \{\varnothing\}$ — функция значений сущностей, ролей и административных ролей (как аналогов сущностей-контейнеров), задающая возможность для каждой из них принимать значение любой (в том числе пустой) конечной последовательности бит;
- $(LC, \leqslant)$ — решётка многоуровневой безопасности уровней конфиденциальности (декартово произведение линейной шкалы уровней конфиденциальности данных и множества всех подмножеств конечного множества неиерархических категорий данных);
- $(f_u, f_e, f_r, f_s) \in FC$ — четвёрка функций уровней конфиденциальности, при этом:
  - $f_u : U \to LC$ — функция, задающая для каждой учётной записи пользователя её уровень доступа — максимальный разрешённый уровень доступа субъект-сессий, функционирующих от её имени;
  - $f_e : E \to LC$ — функция, задающая уровень конфиденциальности для каждой сущности;
  - $f_r : R \cup AR \to LC$ — функция, задающая для каждой роли или административной роли её уровень конфиденциальности;
  - $f_s : S \to LC$ — функция, задающая для каждой субъект-сессии её текущий уровень доступа;
- FC — множество всех четвёрок функций заданного вида;
- $CCR : E \cup R \cup AR \to \{\texttt{true}, \texttt{false}\}$ — функция, задающая способ доступа к сущностям внутри контейнеров или ролям в иерархии ролей (с учётом их мандатных уровней конфиденциальности). Если сущность $e \in C$ является контейнером и доступ к сущностям, содержащимся внутри контейнера e, разрешён без учёта уровня конфиденциальности контейнера e, то по определению выполняется равенство $CCR(e) = \texttt{false}$, в противном случае $CCR(e) = \texttt{true}$. При этом по определению для каждой сущности-объекта $o \in O$ выполняется равенство $CCR(o) = \texttt{true}$, для ролей или административных ролей $r \in R \cup AR$ всегда выполняется равенство $CCR(r) = \texttt{false}$;

— $(LI, \leqslant)$ — линейная шкала двух уровней целостности данных, где $LI = \{i\_low, i\_high\}$, $i\_low < i\_high$;
— $(i_u, i_e, i_r, i_s) \in I$ — четвёрка функций уровней целостности, при этом:
  — $i_u : U \to LI$ — функция, задающая для каждой учётной записи пользователя её уровень целостности — максимальный разрешённый уровень целостности субъект-сессий, функционирующих от её имени;
  — $i_e : E \to LI$ — функция, задающая уровень целостности для каждой сущности;
  — $i_r : R \cup AR \to LI$ — функция, задающая для каждой роли или административной роли её уровень целостности;
  — $i_s : S \to LI$ — функция, задающая для каждой субъект-сессии её текущий уровень целостности;
— I — множество всех четвёрок функций заданного вида;
— $CCRI : E \cup R \cup AR \to \{\texttt{true}, \texttt{false}\}$ — функция, задающая способ доступа к сущностям внутри контейнеров, ролям или административным ролям в иерархии ролей (с учётом их мандатных уровней целостности). Если сущность $e \in C$ является контейнером и доступ к сущностям, содержащимся внутри контейнера e, разрешён без учёта уровня целостности контейнера e, то по определению выполняется равенство $CCRI(e) = \texttt{false}$, в противном случае $CCRI(e) = \texttt{true}$. При этом по определению для каждой сущности-объекта $o \in O$ выполняется равенство $CCRI(o) = \texttt{true}$, для ролей или административных ролей $r \in R \cup AR$ всегда выполняется равенство $CCRI(r) = \texttt{false}$.

Определение 3. Иерархией ролей называется заданное на множестве ролей R отношение частичного порядка $\leqslant$. При этом по определению справедливо: для административной роли $ar \in AR$, если роли $r, r' \in R$ таковы, что $(r, read_r) \in APA(ar)$ и $r' \leqslant r$, то выполняется условие $(r', read_r) \in APA(ar)$. Иерархией административных ролей называется заданное на множестве ролей AR отношение частичного порядка $\leqslant$. При этом по определению справедливо: для административной роли $ar \in AR$, если административные роли $r, r' \in AR$ таковы, что $(r, read_r) \in APA(ar)$ и $r' \leqslant r$, то выполняется условие $(r', read_r) \in APA(ar)$. По определению роли и административные роли несравнимы друг с другом, т. е. их иерархии всегда независимы. В случае, когда для двух ролей или административных ролей $r, r' \in R \cup AR$ выполняются условия $r \leqslant r'$ и $r \neq r'$, будем использовать обозначение $r < r'$. Определим $H_R : R \cup AR \to 2^R \cup 2^{AR}$ — функцию иерархии ролей и административных ролей, сопоставляющую каждой роли $r \in R \cup AR$ множество ролей $H_R(r) \subset R \cup AR$ и удовлетворяющую условию: если $r' \in H_R(r)$, то $r' < r$ и не существует роли $r'' \in R \cup AR$, такой, что $r' < r''$, $r'' < r$.

Используем следующее обозначение:

— $role\_name : (R \cup AR) \times (R \cup AR) \to NAMES$ — функция имён ролей и административных ролей в составе роли или административной роли (как контейнера). При этом по определению если $r \in H_R(r')$, то $role_name(r', r) = $ «», иначе $role\_name(r', r) \neq$ «».

Предположение 2. Для каждой учётной записи пользователя $u \in U$ в зависимости от её уровня целостности задаются следующие индивидуальные административные роли, на которые авторизована только эта учётная запись:

— $u\_admin\_i\_low \in AR$, $i_r(u\_admin\_i\_low) = i\_low$, $f_r(u\_admin\_i\_low) = \otimes LC$, когда $i_u(u) = i\_low$ ($\otimes LC$ — минимальный элемент решётки уровней конфиденциальности LC);

— $u\_admin\_i\_low$, $u\_admin\_i\_high \in AR$, $i_r(u\_admin\_i\_low) = i\_low$, $i_r(u\_admin\_i\_high) = i\_high$, $f_r(u\_admin\_i\_low) = f_r(u\_admin\_i\_high) = \otimes LC$, $u\_admin\_i\_low < u\_admin\_i\_high$, когда $i_u(u) = i\_high$.

Множество других ролей учётной записи пользователя u задается с использованием административных прав доступа соответственно административной роли $u\_admin\_i\_low$ или административных ролей $u\_admin\_i\_low$, $u\_admin\_i\_high$, определяемых функцией APA. Иерархия индивидуальных административных ролей изменяется только при создании, удалении и изменении уровней доступа или целостности соответствующих учётных записей пользователей. На траекториях функционирования системы у этих административных ролей не изменяются имена, уровни конфиденциальности или целостности.

В рамках МРОСЛ ДП-модели, в отличие от моделей семейства $RBAC$ и других ролевых ДП-моделей, впервые вместо функций UA и AUA для задания авторизованных ролей и административных ролей учётных записей пользователей используются права доступа их индивидуальных административных ролей, задаваемых функцией APA. Такой подход, с одной стороны, позволяет реализовать ролевое управление доступом — назначение прав доступа учётным записям пользователей только через роли, с другой стороны, он направлен на обеспечение приоритетности и «монолитности» мандатного управления доступом как при управлении доступом к сущностям, так и при использовании или администрировании ролей. Достигается также большая совместимость со штатными механизмами управления доступом реальной защищённой ОС. Минимальный уровень конфиденциальности административных ролей обеспечивает возможность получения их в качестве текущих (получения доступа $read_a$) субъект-сессией с любым текущим уровнем доступа.

В моделях семейства $RBAC$ и других ролевых ДП-моделях предполагалось, что функция авторизованных ролей учётных записей пользователей UA обладает свойством «наследования» подчинённых ролей, т. е. выполняется следующее условие: для учётной записи пользователя $u \in U$, если роли $r, r' \in R$ таковы, что $r \in UA(u)$ и $r' \leqslant r$, то выполняется условие $r' \in UA(u)$. В рамках МРОСЛ ДП-модели с учётом предположения 2 это обеспечивается «наследованием» административной ролью права доступа $read_r$ от данной роли ко всем подчинённым ей ролям в иерархии. При этом права доступа $write_r$ и own_r таким свойством не обладают, что может позволить гибко задавать «диапазоны» администрируемых ролей по аналогии с моделью ролевого администрирования $ARBAC$ [1]. При реализации иерархий ролей в реальной защищённой ОС по аналогии с файловой системой можно использовать виртуальную структуру сущностей-ролей, отличающуюся от структур для файлов наличием «жёстких» ссылок не только на роли-«объекты» (роли, которым в иерархии не подчинена ни одна другая роль), но и на роли-«контейнеры» (роли, которым в иерархии подчинена хотя бы одна роль).

Используем следующее обозначение:

— $execute\_container : S \times E \to \{\texttt{true}, \texttt{false}\}$ — функция доступа субъект-сессии к сущностям в контейнерах, такая, что по определению для субъект-сессии $s \in S$ и сущности $e \in E$ справедливо равенство $execute\_container(s, e) = \texttt{true}$ тогда и только тогда, когда существует последовательность сущностей $e_1, \ldots, e_n \in E$, где $n \geqslant 1$, $e = e_n$, удовлетворяющих следующим условиям:
  — не существует сущности-контейнера $e_0 \in E$, такой, что $e_1 \in H_E(e_0)$;
  — $e_i \in H_E(e_{i-1})$, где $1 < i \leqslant n$;

— существует $r_i \in R \cup AR$, такая, что $(s, r_i, read_a) \in AA$ и $(e_i, execute_r) \in PA(r_i)$, $[f_e(e_i) \leqslant f_s(s)$ или $CCR(e_i) = \texttt{false}]$ и $[i_e(e_i) \leqslant i_s(s)$ или $CCRI(e_i) = \texttt{false}]$, где $1 \leqslant i < n$.

Данная функция используется для краткости и удобства описания элементов модели с учётом мандатного управления доступом и мандатного контроля целостности и не требует непосредственной реализации в реальной защищённой ОС (она «реализуется» путём последовательной проверки прав доступа к сущностям-контейнерам, содержащим данную сущность, на которую указывает её полное имя, начиная с корневой сущности-контейнера).

Зададим в рамках МРОСЛ ДП-модели механизм ограничений ($Constraints$), при этом используем следующие обозначения:

— APA^* — множество всех функций административных прав доступа административных ролей;
— PA^* — множество всех функций прав доступа ролей и административных ролей;
— AA^* — множество множеств доступов субъект-сессий к ролям или административным ролям;
— $C^{APA} : APA^* \to \{\texttt{true}, \texttt{false}\}$ — функция, задающая ограничение на значения множеств административных прав доступа административных ролей, т. е. по определению множества административных прав доступа административных ролей, заданные функцией $APA \in APA^*$, удовлетворяют ограничению C^{APA}, если выполняется равенство $C^{APA}(APA) = \texttt{true}$;
— $C^{PA} : PA^* \to \{\texttt{true}, \texttt{false}\}$ — функция, задающая ограничение на значения множеств прав доступа ролей и административных ролей, т. е. по определению множества прав доступа ролей и административных ролей, заданные функцией $PA \in PA^*$, удовлетворяют ограничению C^{PA}, если выполняется равенство $C^{PA}(PA) = \texttt{true}$;
— $C^{AA} : AA^* \to \{\texttt{true}, \texttt{false}\}$ — функция, задающая ограничение на значение множества административных доступов субъект-сессий к ролям или административным ролям, т. е. по определению множества административных доступов субъект-сессий к ролям или административным ролям, заданные множеством $AA \in AA^*$, удовлетворяют ограничению C^{AA}, если выполняется равенство $C^{AA}(AA) = \texttt{true}$;
— $Constraint_{APA} = \{C_1^{APA}, \ldots, C_{n_{APA}}^{APA}\}$ — множество функций, задающих некоторый набор ограничений на значения множеств административных прав доступа административных ролей, где $n_{APA} \geqslant 0$. При этом будем писать $Constraint_{APA}(APA) = \texttt{true}$ тогда и только тогда, когда либо $n_{APA} = 0$, либо $n_{APA} > 0$ и $C_i^{APA}(APA) = \texttt{true}$ для $1 \leqslant i \leqslant n_{APA}$;
— $Constraint_{PA} = \{C_1^{PA}, \ldots, C_{n_{PA}}^{PA}\}$ — множество функций, задающих некоторый набор ограничений на значения множеств прав доступа ролей, где $n_{PA} \geqslant 0$. При этом будем писать $Constraint_{PA}(PA) = \texttt{true}$ тогда и только тогда, когда либо $n_{PA} = 0$, либо $n_{PA} > 0$ и $C_i^{PA}(PA) = \texttt{true}$ для $1 \leqslant i \leqslant n_{PA}$;
— $Constraint_{AA} = \{C_1^{AA}, \ldots, C_{n_{AA}}^{AA}\}$ — множество функций, задающих некоторый набор ограничений на значение множества административных доступов субъект-сессий к ролям или административным ролям, где $n_{AA} \geqslant 0$. При этом будем писать $Constraint_{AA}(AA) = \texttt{true}$ тогда и только тогда, когда либо $n_{AA} = 0$, либо $n_{AA} > 0$ и $C_i^{AA}(AA) = \texttt{true}$ для $1 \leqslant i \leqslant n_{AA}$.

Механизм ограничений как мощное средство ролевого управления доступом целесообразно реализовать, как минимум, на перспективу. Ограничения, например, вза-

имного исключения ролей, количественные на обладание ролью, могут стать удобным средством администрирования защищённой ОС. Ограничениям на функции UA и $roles$, заданным в моделях семейства $RBAC$ и других ролевых ДП-моделях, в МРОСЛ ДП-модели соответствуют ограничения на функцию APA и множество AA.

Определение 4. Пусть определены:

— множества учётных записей пользователей U, сущностей E, субъект-сессий S, прав доступа к сущностям P, учётных записей доверенных пользователей L_U, доверенных субъект-сессий L_S, доступов субъект-сессий к сущностям A, доступов субъект-сессий к ролям и административным ролям AA, информационных потоков F, ограничений на значения множеств административных прав доступа административных ролей $Constraint_{APA}$, множеств прав доступа ролей $Constraint_{PA}$, множеств доступов субъект-сессий к ролям или административным ролям $Constraint_{AA}$;

— функции административных прав доступа к ролям административных ролей APA, прав доступа ролей и административных ролей PA, принадлежности субъект-сессий учётным записям пользователей $user$, уровней конфиденциальности (f_u, f_e, f_r, f_s), мандатных атрибутов конфиденциальности CCR, уровней целостности (i_u, i_e, i_r, i_s), мандатных атрибутов целостности $CCRI$, иерархии ролей H_R, иерархии сущностей H_E, иерархии субъект-сессий H_S.

Определим $G = (APA, PA, user, (f_u, f_e, f_r, f_s), CCR, (i_u, i_e, i_r, i_s), CCRI, A, AA, F, H_R, H_E, H_S, Constraint_{APA}, Constraint_{PA}, Constraint_{AA}, L_U, L_S)$ — состояние системы.

Используем обозначение:

— $\Sigma(G^*, OP)$ — система, где G^* — множество всех возможных её состояний; OP — множество правил преобразования её состояний.

Используемые в МРОСЛ ДП-модели правила преобразования состояний из множества OP классифицированы на де-юре правила, реализуемые в реальной защищённой ОС, т. е. приводящие к «реальным» изменениям её параметров (изменению множеств прав доступа ролей, получению доступов субъект-сессий к сущностям или ролям и т. д.), и де-факто правила, не требующие реализации в ОС, так как используются в модели для отражения факта получения субъект-сессией де-факто владения субъект-сессиями или факта реализации информационного потока по памяти или по времени.

2. Параметрически ассоциированные сущности. Специальные административные роли

Сформулируем предположения, позволяющие учесть наличие в реальных защищённых ОС сущностей, параметрически ассоциированных с учётными записями пользователей или ролями.

Предположение 3. Для каждой учётной записи пользователя задаётся множество сущностей, параметрически с ней ассоциированных, получение доступов на чтение или на запись к которым субъект-сессией позволяет ей создать субъект-сессию от имени данной учётной записи пользователя. Множество сущностей, параметрически ассоциированных с учётной записью пользователя, не изменяется в процессе функционирования системы.

Предположение 4. Для каждой роли или административной роли задаётся (возможно, пустое) множество сущностей, параметрически с ней ассоциированных. При этом для получения или удаления субъект-сессией доступа к роли, кроме наличия к данной роли соответствующего права доступа, этой субъект-сессии необходимо

получить доступ на чтение или на запись ко всем сущностям, параметрически ассоциированным с данной ролью. Множество сущностей, параметрически ассоциированных с ролью или административной ролью, не изменяется в процессе функционирования системы. Множество сущностей, параметрически ассоциированных с каждой индивидуальной административной ролью учётной записи пользователя, совпадает с множеством сущностей, параметрически ассоциированных с данной учётной записью пользователя.

Примерами параметрически ассоциированных с учётными записями пользователей или ролями сущностей являются файлы паролей, `cookies` или ключей и т. д., т. е. то, «знание» (получение доступа на чтение) или «подмена» (получение доступа на запись) чего позволяет недоверенной субъект-сессии создать в системе субъект-сессию от имени данной учётной записи пользователя. Лишь у небольшого числа ролей могут быть такие сущности, их наличие позволяет обеспечить дополнительную защиту при получении субъект-сессией доступа к роли, например, потребовав дополнительно повторно ввести пароль.

С учётом сделанных предположений используем следующие обозначения:

— $]u[\in E$ — множество сущностей, параметрически ассоциированных с учётной записью пользователя $u \in U$;
— $UE = \{e \in]u[\colon u \in U\}$ — множество сущностей, каждая из которых параметрически ассоциирована хотя бы с одной учётной записью пользователя;
— $]r[\in E$ — множество сущностей, параметрически ассоциированных с ролью или административной ролью $r \in R \cup AR$, при этом для каждой учётной записи пользователя $u \in U$ в соответствии с предположением 4 выполняется условие $]u[=]u\_admin\_i\_low[$, и если $i_u(u) = i\_high$, то $]u[=]u\_admin\_i\_high[$;
— $RE = \{e \in]r[\colon r \in R \cup AR\}$ — множество сущностей, параметрически ассоциированных хотя бы с одной ролью или административной ролью.

Предположение 5. В рамках МРОСЛ ДП-модели на траекториях функционирования системы не изменяются функции $Constraint_{APA}$, $Constraint_{PA}$ и $Constraint_{AA}$. Новые значения для функций f_s и i_s задаются только при создании соответствующих новых субъект-сессий. В дополнение к предположению 1, недоверенная субъект-сессия всегда может создать недоверенную субъект-сессию с низким уровнем доступа. В начальном состоянии G_0 любой системы $\Sigma(G^*, OP)$ выполняются следующие условия:

— функции APA_0, PA_0 и множество AA_0 удовлетворяют соответствующим ограничениям $Constraint_{APA}$, $Constraint_{PA}$ и $Constraint_{AA}$;
— для любых субъект-сессии $s \in S_0$ и сущности $e \in E_0$, если $(s, e, \alpha_a) \in A_0$, где $\alpha_a \in R_a$, то справедливо равенство $execute\_container_0(s, e) = \texttt{true}$.

В предположении 5 требуется, чтобы в начальном состоянии выполнялись ограничения и доступы субъект-сессий к сущностям соответствовали мандатному управлению доступом и мандатному контролю целостности. Кроме того, для строгого задания правил преобразования состояний системы, ориентированных на реализацию в реальной защищённой ОС, впервые по сравнению с предшествующими ДП-моделями в рамках МРОСЛ ДП-модели допускается администрирование существенных параметров системы, влияющих на её безопасность (множеств учётных записей пользователей, ролей или административных ролей, значений уровней доступа, конфиденциальности или целостности учётных записей пользователей, ролей, административных ролей, сущностей). Для этого в модель включены специальные административные роли.

Используем следующие обозначения:

— $ADMIN\_ROLES \subset AR$ — множество специальных административных ролей, которые не могут создаваться, удаляться, переименовываться, менять свои параметры в процессе функционирования системы. Все административные роли из множества $ADMIN\_ROLES$ по определению обладают высоким уровнем целостности: для административной роли $ar \in ADMIN\_ROLES$ справедливо равенство $i_r(ar) = i\_high$;

— $users\_admin\_role \in ADMIN\_ROLES$ — специальная административная роль, доступ субъект-сессии на чтение к которой требуется для изменения уровня доступа или целостности, удаления учётной записи пользователя;

— $entities\_admin\_role \in ADMIN\_ROLES$ — специальная административная роль, доступ субъект-сессии на чтение к которой требуется для изменения ею единственной роли-владельца сущности (роли, обладающей правом доступа владения own_r к сущности), получения данных о ролях или административных ролях, обладающих правами доступа к сущности, получения числа «жёстких» ссылок к сущности, изменения уровня конфиденциальности или целостности сущности, изменения мандатного атрибута конфиденциальности CCR, мандатного атрибута целостности $CCRI$ или метки разделяемости сущности-контейнера, переименования, удаления сущности-контейнера или «жёсткой» ссылки на него с мандатным атрибутом целостности $CCRI$ равным `false`;

— $subjects\_admin\_role \in ADMIN\_ROLES$ — специальная административная роль, доступ субъект-сессии на чтение к которой требуется для изменения единственной роли-владельца другой субъект-сессии, получения данные о ролях или административных ролях, обладающих правами доступа владения к другой субъект-сессии;

— $roles\_admin\_role, admin\_roles\_admin\_role \in ADMIN\_ROLES$ — специальные административные роли, являющиеся ролями-владельцами ролей или административных ролей соответственно, доступ субъект-сессии на чтение к каждой из которых требуется для управления доступом к роли или административной роли, изменения её уровней конфиденциальности или целостности, создания, переименования, удаления, получения параметров роли или административной роли, создания, изменения уровня доступа или целостности учётной записи пользователя;

— $downgrade\_admin\_role \in ADMIN\_ROLES$ — специальная административная роль, доступ субъект-сессии на чтение к которой требуется для нарушения ею правил мандатного управления доступом.

По определению справедливы равенства

$$f_r(users\_admin\_role) = f_r(entities\_admin\_role) = f_r(subjects\_admin\_role) = \\ = f_r(roles\_admin\_role) = f_r(admin\_roles\_admin\_role) = \otimes LC, \\ f_r(downgrade\_admin\_role) = \oplus LC.$$

3. Де-юре правила администрирования системы

Всего в рамках МРОСЛ ДП-модели заданы 34 де-юре правила преобразования состояний, условия и результаты применения которых соответствуют предположениям модели. Правила предназначены для формального описания (спецификации) следующих основных функций механизма управления доступом защищённой ОС:

— создания, удаления, переименования, получения или изменения параметров учётных записей пользователей, ролей, административных ролей, сущностей или «жёстких» ссылок на них, субъект-сессий;

— получения доступов субъект-сессий к сущностям, ролям или административным ролям;
— изменения прав доступа ролей или административных ролей к сущностям, субъект-сессиям, ролям или административным ролям;
— изменения иерархии, уровней конфиденциальности или целостности сущностей, ролей или административных ролей.

В таблице приведено детальное описание де-юре правил администрирования системы, при этом в ней не указываются элементы последующего состояния G', которые не изменяются в результате применения соответствующего правила к исходному состоянию G.

Де-юре правила администрирования системы в рамках МРОСЛ ДП-модели

Правило: *create_user*(x, x', u, uc, ui, ue)
Условия применения: $x, x' \in S$, $u \notin U$, $(x, users\_admin\_role, read_a) \in AA$, $(x, roles\_admin\_role, \alpha_a) \in AA$, $(x, admin\_roles\_admin\_role, \alpha_a) \in AA$, где $\alpha_a \in \{read_a, write_a\}$; $ue \subset UE$; [$uc = f_s(x)$ или ($uc \leqslant f_s(x)$ и $(x, downgrade\_admin\_role, read_a) \in AA$)]; $ui \leqslant i_s(x)$; [для $e \in ue$ выполняется $f_e(e) = uc$, $i_e(e) = ui$]; $Constraint_{APA}(APA') = \texttt{true}$; $Constraint_{PA}(PA') = \texttt{true}$; [если $ui = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$]
Результаты применения: $U' = U \cup \{u\}$, $]u[= ue$, $f'_u(u) = uc$, $i'_u(u) = ui$, если $ui = i\_high$, то $L'_U = L_U \cup \{u\}$, иначе $N'_U = N_U \cup \{u\}$, для $u' \in U$ выполняется $f'_u(u') = f_u(u')$, $i'_u(u') = i_u(u')$, $AR' = AR \cup \{u\_admin\_li : li \leqslant ui\}$, $f'_r(u\_admin\_li) = \otimes LC$, $i'_r(u\_admin\_li) = li$, $shared\_container'(u\_admin\_li) = \texttt{true}$, $CCR'(u\_admin\_li) = CCRI'(u\_admin\_li) = \texttt{false}$, $role\_name'(u\_admin\_li) =$“u_admin_li”, где $li \leqslant ui$, $H'_R(u\_admin\_i\_low) = \varnothing$, если $ui = i\_high$, то $H'_R(u\_admin\_i\_high) = \{u\_admin\_i\_low\}$, $R' = R \cup \{u\_c\_l\_li : l \leqslant uc, li \leqslant ui\}$, $f'_r(u\_c\_l\_li) = l$, $i'_r(u\_c\_l\_li) = li$, $shared\_container'(u\_c\_l\_li) = \texttt{true}$, $CCR'(u\_c\_l\_li) = CCRI'(u\_c\_l\_li) = \texttt{false}$, $role\_name'(u\_c\_l\_li) =$ = “$u\_c\_l\_li$”, где $l \leqslant uc$, $li \leqslant ui$, $APA'(admin\_roles\_admin\_role) = APA(admin\_roles\_admin\_role) \cup \{(u\_admin\_li, own_r) : li \leqslant ui\}$, $APA'(roles\_admin\_role) = APA(roles\_admin\_role) \cup \{(u\_c\_l\_li, own_r) : l \leqslant uc, li \leqslant ui\}$, для $ar \in AR$ выполняется $APA'(ar) = APA(ar) \cup \{(u\_admin\_li, execute_r) : li \leqslant ui\} \cup \{(u\_c\_l\_li, execute_r) : l \leqslant uc, li \leqslant ui\}$, $APA'(u\_admin\_i\_low) = \{(u\_admin\_i\_low, \alpha_r) : \alpha_r \in \{read_r, write_r, execute_r\}\} \cup$ $\cup \{(u\_c\_l\_(i\_low), \alpha_r) : l \leqslant uc, \alpha_r \{read_r, write_r, execute_r\}\}$, если $ui = i\_high$, то $APA'(u\_admin\_i\_high) = \{(u\_admin\_li, \alpha_r) : li \leqslant ui, \alpha_r \in \{read_r, write_r, execute_r\}\} \cup \{(u\_c\_l\_(i\_high), \alpha_r) : l \leqslant uc, \alpha_r \in \{read_r, write_r, execute_r\}\}$, $H'_R(u\_c\_l\_li) = \{u\_c\_l'\_li' : l' < l, li' < li$ и не существуют $l'' \in LC (l' < l'' < l)$ или $li'' \in LI$ $(li' < li'' < li)\}$, $PA'(u\_c\_l\_li) = \varnothing$, где $l \leqslant uc$, $li \leqslant ui$
Правило: *set_user_labels*(x, x', u, uc, ui, ue)
Условия применения: $x, x' \in S$, $u \in U$, $user^{-1}(u) = \varnothing$, ($x$, $users\_admin\_role$, $read_a$) $\in AA$, (x, $roles\_admin\_role$, α_a) $\in AA$, (x, $admin\_roles\_admin\_role$, α_a) $\in AA$, где $\alpha_a \in \{read_a, write_a\}$, [если $f_u(u) \neq uc$, то $\max(f_u(u), uc) \leqslant f_s(x)$ и (x, $downgrade\_admin\_role$, $read_a$) $\in AA$], $\max(i_u(u), ui) \leqslant i_s(x)$, [для $e \in ue$ верно $f_e(e) = uc$, $i_e(e) = ui$], $Constraint_{APA}(APA') = \texttt{true}$, $Constraint_{PA}(PA') = \texttt{true}$, [если $ui = i\_high$, то (x', $f_s(x)\_i\_entity$, $write_a$) $\in A$]

Продолжение таблицы

Результаты применения: $]u[= ue$, $f'_u(u) = uc$, $i'_u(u) = ui$, если $ui = i\_high$, то $L'_U = L_U \cup \{u\}$, $N'_U = N_U \setminus \{u\}$, иначе $N'_U = N_U \cup \{u\}$, $L'_U = L_U \setminus \{u\}$, для $u' \in U \setminus \{u\}$ верно $f'_u(u') = f_u(u')$, $i'_u(u') = i_u(u')$, $AR' = (A \cup \{u\_admin\_li{:}\ i_u(u) < li \leqslant ui\}) \setminus \{u\_admin\_li{:}\ ui < li \leqslant i_u(u)\}$, $f'_r(u\_admin\_li) = \otimes LC$, $i'_r(u\_admin\_li) = li$, $shared\_container'(u\_admin\_li) =$ `true`, $CCR'(u\_admin\_li) = CCRI'(u\_admin\_li) =$ `false`, $role\_name'(u\_admin\_li) =$ "u_admin_li", где $i_u(u) < li \leqslant ui$, если $ui = i\_high$, то $H'_R(u\_admin\_i\_high) = \{u\_admin\_i\_low\}$, $R' = (R \cup \{u\_c\_l\_li : f_u(u) < l \leqslant uc, i_u(u) < li \leqslant ui\}) \setminus \{u\_c\_l\_li : uc < l \leqslant f_u(u), ui < li \leqslant i_u(u)\}$, $f'_r(u\_c\_l\_li) = l$, $i'_r(u\_c\_l\_li) = li$, $shared\_container'(u\_c\_l\_li) =$ `true`, $CCR'(u\_c\_l\_li) = CCRI'(u\_c\_l\_li) =$ `false`, $role\_name'(u\_c\_l\_li) =$ "$u\_c\_l\_li$", где $f_u(u) < l \leqslant uc$, $i_u(u) < li \leqslant ui$, $APA'(admin\_roles\_admin\_role) = (APA(admin\_roles\_admin\_role) \cup \{(u\_admin\_li, own_r) : i_u(u) < li \leqslant ui\}) \setminus \{(u\_admin\_li, own_r) : ui < li \leqslant i_u(u)\}$, $APA'(roles\_admin\_role) = (APA(roles\_admin\_role) \cup \{(u\_c\_l\_li, own_r) : f_u(u) < l \leqslant uc, i_u(u) < li \leqslant ui\}) \setminus$ $\setminus \{(u\_c\_l\_li, own_r) : uc < l \leqslant f_u(u), ui < li \leqslant i_u(u)\}$, для $ar \in AR$ выполняется $APA'(ar) = (APA(ar) \cup \{(u\_admin\_li, execute_r) : i_u(u) < li \leqslant ui\} \cup$ $\cup \{(u\_c\_l\_li, execute_r) : f_u(u) < l \leqslant uc, i_u(u) < li \leqslant ui\}) \setminus (\{(u\_admin\_li, execute_r) : ui < li \leqslant$ $\leqslant i_u(u)\} \cup \{(u\_c\_l\_li, execute_r) : uc < l \leqslant f_u(u), ui < li \leqslant i_u(u)\})$, $APA'(u\_admin\_i\_low) =$ $= (APA'(u\_admin\_i\_low) \cup \{(u\_c\_l\_(i\_low), \alpha_r) : f_u(u) < l \leqslant uc, \alpha_r \in \{read_r, write_r, execute_r\}\}) \setminus \{(u\_c\_l\_(i\_low), \alpha_r) : uc < l \leqslant f_u(u), \alpha_r \in \{read_r, write_r, execute_r\}\}$, если $ui = i\_high$, то $APA'(u\_admin\_i\_high) = (APA(u\_admin\_i\_high) \cup \{(u\_admin\_li, \alpha_r) : li \leqslant ui, \alpha_r \in \{read_r, write_r, execute_r\}\} \cup \{(u\_c\_l\_(i\_high), \alpha_r) : f_u(u) < l \leqslant uc, \alpha_r \in \{read_r, write_r, execute_r\}\}) \setminus \{(u\_c\_l\_(i\_high), \alpha_r) : uc < l \leqslant f_u(u), \alpha_r \in \{read_r, write_r, execute_r\}\}$, $H'_R(u\_c\_l\_li) = \{u\_c\_l'\_li' : l' < l$, $li' < li$ и не существуют $l'' \in LC(l' < l'' < l)$ или $li'' \in LI$ $(li' < li'' < li)\}$, $PA'(u\_c\_l\_li) = \varnothing$, где $f_u(u) < l \leqslant uc$, $i_u(u) < li \leqslant ui$
Правило: $delete\_user(x, x', u)$
Условия применения: $x, x' \in S$, $u \in U$, $user^{-1}(u) = \varnothing$, $(x, users\_admin\_role, read_a) \in AA$, $(x, users\_admin\_role, read_a) \in AA$, $(x, roles\_admin\_role, read_a) \in AA$, $[f_u(u) = f_s(x)$ или $(f_u(u) \leqslant f_s(x)$ и $(x, downgrade\_admin\_role, read_a) \in AA)]$, $i_u(u) \leqslant i_s(x)$, $Constraint_{APA}(APA') =$ `true`, $Constraint_{PA}(PA') =$ `true`, [если $i_u(u) = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$]
Результаты применения: $U' = U \setminus \{u\}$, для $u' \in U'$ выполняется $f'_u(u') = f_u(u')$, $i'_u(u') = i_u(u')$, $AR' = AR \setminus \{u\_admin\_li : li \leqslant ui\}$, $R' = R \setminus \{u\_c\_l\_li : l \leqslant uc, li \leqslant ui\}$, для $ar \in AR'$ выполняются равенства $APA'(ar) = APA(ar) \setminus (\{(u\_admin\_li, \alpha_r) : li \leqslant ui$, $\alpha_r \in R_r\} \cup \{(u\_c\_l\_li, \alpha_r) : l \leqslant uc, li \leqslant ui, \alpha_r \in R_r\})$
Правило: $get\_user\_attr(x, u, z)$
Условия применения: $x \in S$, $u \in U$, $z \in O$, $[f_u(u) \leqslant f_s(x)$ или $(x, downgrade\_admin\_role, read_a) \in AA]$, $(x, z, write_a) \in A$
Результаты применения: $V'(z) = (f_u(u), i_u(u))$, (если $user(x) = u$ или $(x, users\_admin\_role, read_a) \in AA$, то $\{(r', \alpha_r) : (r', \alpha_r) \in APA(u\_admin\_li)$, $li \leqslant i_u(u)$ и $(f_r(r') \leqslant f_s(x)$ или $(x, downgrade\_admin\_role, read_a) \in AA)\}$, иначе "$\varnothing$"), (если $user(x) = u$ или $(x, users\_admin\_role, read_a) \in AA$, то $\{s \in S{:}\ user(s) = u$ и либо $f_s(s) \leqslant f_s(x)$ или $(x, downgrade\_admin\_role, read_a) \in AA)\}$, иначе "$\varnothing$"))
Правило: $grant\_rights(x, x', r, \{(y, \alpha_{r_j}){:}\ 1 \leqslant j \leqslant k\})$
Условия применения: $x, x' \in S$, $y \in E$, $r \in R \cup AR$, $\alpha_{r_j} \in \{read_r, write_r, execute_r\}$, $(x, r, write_a) \in AA$, $i_e(y) \leqslant i_s(x)$, [существует $r' \in R \cup AR$ $((x, r', read_a) \in AA$, $(y, own_r) \in PA(r'))]$, [либо $(execute\_container(x, y) =$ `true` и $f_e(y) = f_s(x))$, либо $(x, downgrade\_admin\_role, read_a) \in AA]$, [если $\alpha_{r_j} = write_r$, то $i_e(y) \leqslant i_r(r)]$, $Constraint_{PA}(PA') =$ `true`, [если $i_e(y) = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A]$, где $1 \leqslant j \leqslant k$
Результаты применения: $PA'(r) = PA(r) \cup \{(y, \alpha_{r_j}) : 1 \leqslant j \leqslant k\}$ и для $r' \in (R \cup AR) \setminus \{r\}$ выполняется равенство $PA'(r') = PA(r')$

Продолжение таблицы

Правило: *remove_rights*(x, x', r, $\{(y, \alpha_{r_j})$: $1 \leqslant j \leqslant k\}$)
Условия применения: $x, x' \in S$, $y \in E$, $r \in R \cup AR$, $\alpha_{r_j} \in \{read_r, write_r, execute_r\}$, $\{(y, \alpha_{r_j}) : 1 \leqslant j \leqslant k\} \subset PA(r)$, $(x, r, write_a) \in AA$, $i_e(y) \leqslant i_s(x)$, [существует $r' \in R \cup AR$ ($(x, r', read_a) \in AA$, $(y, own_r) \in PA(r')$)], [либо ($execute\_container(x, y) =$ `true` и $f_e(y) = f_s(x)$), либо $(x, downgrade\_admin\_role, read_a) \in AA$], $Constraint_{PA}(PA') =$ `true`, [если $i_e(y) = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$], где $1 \leqslant j \leqslant k$
Результаты применения: $PA'(r) = PA(r) \setminus \{(y, \alpha_{r_j}) : 1 \leqslant j \leqslant k\}$, и для $r' \in (R \cup AR) \setminus \{r\}$ выполняется равенство $PA'(r') = PA(r')$
Правило: *set_entity_owner*(x, x', r, r', y)
Условия применения: $x, x' \in S$, $y \in E$, $r, r' \in R \cup AR$, $\{(x, r, read_a)$, $(x, r, write_a)$, $(x, r', write_a)$, $(x, entities\_admin\_role, read_a)\} \subset AA$, $(y, own_r) \in PA(r)$, $Constraint_{PA}(PA') =$ `true`, $i_e(y) \leqslant \min(i_r(r'), i_s(x))$, [либо ($execute\_container(x, y) =$ `true` и $f_e(y) = f_s(x)$), либо $((x, downgrade\_admin\_role, read_a) \in AA)$], [если $i_e(y) = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$]
Результаты применения: $PA'(r) = PA(r) \setminus \{(y, own_r)\}$, $PA'(r') = PA(r') \cup \{(y, own_r)\}$, для $r'' \in (R \cup AR) \setminus \{r, r'\}$ выполняется равенство $PA'(r'') = PA(r'')$
Правило: *set_subject_owner*(x, x', r, r', y)
Условия применения: $x, x' \in S$, $y \in S$, $r, r' \in R \cup AR$, $\{(x, r, read_a)$, $(x, r, write_a)$, $(x, r', write_a)$, $(x, subjects\_admin\_role, read_a)\} \subset AA$, $(y, own_r) \in PA(r)$, $Constraint_{PA}(PA') =$ `true`, $i_s(y) \leqslant$ $\leqslant \min(i_r(r'), i_s(x))$, [либо $f_s(y) = f_s(x)$, либо $(x, downgrade\_admin\_role, read_a) \in AA$], [если $i_s(y) = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$]
Результаты применения: $PA'(r) = PA(r) \setminus \{(y, own_r)\}$, $PA'(r') = PA(r') \cup \{(y, own_r)\}$, для $r'' \in (R \cup AR) \setminus \{r, r'\}$ выполняется равенство $PA'(r'') = PA(r'')$
Правило: *grant_admin_rights*(x, x', ar, $\{(r, \alpha_{r_j})$: $1 \leqslant j \leqslant k\}$)
Условия применения: $x, x' \in S$, $r \in R \cup AR$, $ar \in AR$, $\alpha_{r_j} \in \{write_r, read_r\}$, $(x, ar, write_a) \in AA$, $i_r(r) \leqslant i_r(ar)$, $i_r(r) \leqslant i_s(x)$, [если $r \in R$, то $(x, roles\_admin\_role, read_a) \in AA$, если $r \in AR$, то $(x, admin\_roles\_admin\_role, read_a) \in AA$], [либо $f_r(r) = f_s(x)$, либо $(x, downgrade\_admin\_role, read_a) \in AA$], $Constraint_{APA}(APA') =$ `true`, [если $i_r(r) = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$], где $1 \leqslant j \leqslant k$
Результаты применения: $APA'(ar) = APA(ar) \cup \{(r, \alpha_{r_j}) : \alpha_{r_j} = write_r, 1 \leqslant j \leqslant k\} \cup \{(r', \alpha_{r_j})$: $\alpha_{r_j} = read_r$, $r' \leqslant r$, $1 \leqslant j \leqslant k\}$, для $ar' \in AR \setminus \{ar\}$ выполняется равенство $APA'(ar') = APA(ar')$
Правило: *remove_admin_rights*(x, x', ar, $\{(r, \alpha_{r_j})$: $1 \leqslant j \leqslant k\}$)
Условия применения: $x, x' \in S$, $r \in R \cup AR$, $ar \in AR$, $\alpha_{r_j} \in \{write_r, read_r\}$, $\{(r, \alpha_{r_j}) : \alpha_{r_j} = write_r, 1 \leqslant j \leqslant k\} \subset APA(ar)$, $(x, ar, write_a) \in AA$, $i_r(r) \leqslant i_s(x)$, [если $r \in R$, то $(x, roles\_admin\_role, read_a) \in AA$, если $r \in AR$, то $(x, admin\_roles\_admin\_role, read_a) \in AA$], [либо $f_r(r) = f_s(x)$, либо $(x, downgrade\_admin\_role, read_a) \in AA$], $Constraint_{APA}(APA') =$ `true`, [если $i_r(r) = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$], где $1 \leqslant j \leqslant k$
Результаты применения: $APA'(ar) = APA(ar) \setminus (\{(r, \alpha_{r_j}) : \alpha_{r_j} = write_r, 1 \leqslant j \leqslant k\} \cup \{(r', \alpha_{r_j})$: $\alpha_{r_j} = read_r$, $r \leqslant r'$, $1 \leqslant j \leqslant k\})$, для $ar' \in AR \setminus \{ar\}$ верно равенство $APA'(ar') = APA(ar')$
Правило: *create_role*(x, x', r, rc, ri, re, $name$, rz)
Условия применения: $x, x' \in S$, $r \in R \cup AR$, [если $rz \in R$, то $(x, roles\_admin\_role, \alpha_a) \in AA$, если $rz \in AR$, то $(x, admin\_roles\_admin\_role, \alpha_a) \in AA$, где $\alpha_a \in \{read_a, write_a\}$], $(x, rz, write_a) \in AA$, $re \subset RE$, $name \in NAME \setminus \{«»\}$, [либо $r = f_r(rz) = f_s(x)$, либо $r \leqslant f_r(rz)$ и $(x, downgrade\_admin\_role, read_a) \in AA$], $ri \leqslant \min(i_r(rz), i_s(x))$, [для $e \in re$ выполняется $f_e(e) = rc$, $i_e(e) = ri$], $Constraint_{APA}(APA') =$ `true`, $Constraint_{PA}(PA') =$ `true`, [если $i_r(rz) = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$]
Результаты применения: если $rz \in R$, то $R' = R \cup \{r\}$, $APA'(roles\_admin\_role) = APA(roles\_admin\_role) \cup \{(r, own_r), (r, execute_r)\} \cup \{(r, read_r) : (rz, read_r) \in APA(roles\_admin\_role)\}$, если $rz \in AR$, то $AR' = AR \cup \{r\}$, $APA'(admin\_roles\_admin\_role) = APA(admin\_roles\_admin\_role) \cup \{(r, own_r), (r, execute_r)\} \cup \{(r, read_r) : (rz, read_r) \in APA(admin\_roles\_admin\_role)\}$, для $ar \in AR \setminus \{admin\_roles\_admin\_role, roles\_admin\_role\}$ выполняется $APA'(ar) = APA(ar) \cup \{(r, execute_r)\} \cup \{(r, read_r) : (rz, read_r) \in APA(ar)\}$, $f'_r(r) = rc$, $i'_r(r) = ri$, $CCR'(r) =$ `false`, $CCRI'(r) =$ `false`, $role\_name'(rz, r) = name$, $]r[= re$, $shared\_container'(r) =$ `true`, $PA'(r) = \varnothing$, $H'_R(rz) = H_E(z) \cup \{r\}$, $H'_R(r) = \varnothing$, для $r \in (R \cup AR) \setminus \{r\}$ выполняется равенство $H'_R(r) = H_R(r)$

Окончание таблицы

Правило: *delete_role*(x, x', r, rz)
Условия применения: $x, x' \in S$, $r \in (R \cup AR) \setminus (\{u\_admin\_li : u \in U, li \leqslant i_u(u)\} \cup \{u\_c\_l\_li : u \in U,\ l \leqslant f_u(u),\ li \leqslant i_u(u)\} \cup ADMIN\_ROLES)$, [если $r \in R$, то $rz \in R$, $(x, roles\_admin\_role, \alpha_a) \in AA$, если $r \in AR$, то $rz \in AR$, $(x, admin\_roles\_admin\_role, \alpha_a) \in AA$, где $\alpha_a \in \{read_a, write_a\}$], $r \in H_R(rz)$, $HR(r) = \varnothing$, [не существует $rz' \in R \cup AR$, такой, что $rz' \neq rz$ и $r \in H_R(rz')$], [$(x, rz, write_a) \in AA$, $Constraint_{AA}(AA') =$ `true`, $Constraint_{APA}(APA') =$ `true`, $Constraint_{PA}(PA') =$ `true`], [либо $f_r(r) = f_s(x)$, либо $f_r(r) \leqslant f_s(x)$ и $(x, downgrade\_admin\_role, read_a) \in AA$], $i_r(r) \leqslant i_s(x)$, [если $i_e(z) = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$]
Результаты применения: $R' = R \setminus \{r\}$, $AR' = AR \setminus \{r\}$, $H'_R(rz) = H_R(rz) \setminus \{r\}$, для всех $r' \in (R' \cup AR') \setminus \{rz\}$ выполняется равенство $H'_R(r') = H_R(r')$, для $ar \in AR'$ выполняются равенства $APA'(ar) = APA(ar) \setminus \{(r, \alpha_r) : \alpha_r \in R_r\}$, $AA' = AA \setminus \{(s, r, \alpha_a) : s \in S, \alpha_a \in R_a\}$
Правило: *create_hard_link_role*$(x, x', r, name, rz)$
Условия применения: $x, x' \in S$, $r \in (R \cup AR) \setminus (\{u\_admin\_li : u \in U, li \leqslant i_u(u)\} \cup \{u\_c\_l\_li : u \in U, l \leqslant f_u(u), li \leqslant i_u(u)\} \cup ADMIN\_ROLES)$, не выполняется условие $rz \leqslant r$, $Constraint_{APA}(APA') =$ `true`, [если $rz \in R$, то $(x, roles\_admin\_role, \alpha_a) \in AA$, если $rz \in AR$, то $(x, admin\_roles\_admin\_role, \alpha_a) \in AA$, где $\alpha_a \in \{read_a, write_a\}$], $(x, rz, write_a) \in AA$, $name \in NAME \setminus \{«»\}$, [либо $f_r(r) = f_r(rz) = f_s(x)$, либо $f_r(r) \leqslant f_r(rz)$ и $(x, downgrade\_admin\_role, read_a) \in AA$], $i_r(r) \leqslant \min(i_r(rz), i_s(x))$, [если $i_r(rz) = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$]
Результаты применения: $role\_name'(rz, r) = name$, $H'_R(rz) = H_R(rz) \cup \{r\}$, $H'_R(r) = \varnothing$, для $r' \in (R \cup AR) \setminus \{r\}$ выполняется равенство $H'_R(r') = H_R(r')$, для $ar \in AR$, таких, что $(rz, read_r) \in APA(ar)$, выполняется $APA'(ar) = APA(ar) \cup \{(r', read_r) : r' \leqslant r\}$
Правило: *delete_hard_link_role*(x, x', r, rz)
Условия применения: $x, x' \in S$, $r \in (R \cup AR) \setminus (\{u\_admin\_li : u \in U, li \leqslant i_u(u)\} \cup \{u\_c\_l\_li : u \in U, l \leqslant f_u(u), li \leqslant i_u(u)\} \cup ADMIN\_ROLES)$, [если $rz \in R$, то $(x, roles\_admin\_role, \alpha_a) \in AA$, если $rz \in AR$, то $(x, admin\_roles\_admin\_role, \alpha_a) \in AA$, где $\alpha_a \in \{read_a, write_a\}$], $r \in H_R(rz)$, [существует $rz' \in R \cup AR$, такая, что $rz' \neq rz$ и $r \in H_R(rz')$], $(x, rz, write_a) \in AA$, [либо $f_r(r) = f_s(x)$, либо $f_r(r) \leqslant f_s(x)$ и $(x, downgrade\_admin\_role, read_a) \in AA$], $i_r(r) \leqslant i_s(x)$, [если $i_e(z) = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$]
Результаты применения: $H'_R(rz) = H_R(rz) \setminus \{y\}$, для $r' \in (R \cup AR) \setminus \{r\}$ верно $H'_R(r') = H_R(r')$
Правило: *set_container_attr*$(x, x', y, ccr, ccri, t)$
Условия применения: $x, x' \in S$, $y \in C$, $ccr, ccri, t \in \{$`true`, `false`$\}$, $i_e(y) \leqslant i_s(x)$, [либо существует $r \in R \cup AR$ ($(x, r, read_a) \in AA$, $(y, own_r) \in PA(r)$), либо $(x, entities\_admin\_role, read_a) \in AA$], [либо ($execute\_container(x, y) =$ `true` и $f_e(y) = f_s(x)$), либо $(x, downgrade\_admin\_role, read_a) \in AA$], [$(x, entities\_admin\_role, read_a) \in AA$ или $(x, downgrade\_admin\_role, read_a) \in AA$], [если $i_e(y) = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$]
Результаты применения: $CCR'(y) = ccr$, $CCRI'(y) = ccri$, $shared\_container'(y) = t$
Правило: *set_entity_labels*(x, x', y, yc, yi)
Условия применения: $x, x' \in S$, $y \in E$, $y \notin UE \cup RE$, $(x, entities\_admin\_role, read_a)$, $(x, downgrade\_admin\_role, read_a) \in AA$, $\{(s, y, \alpha_a) : s \in S, \alpha_a \in R_a\} \cap A = \varnothing$, $\max(f_e(y), yc) \leqslant f_s(x)$, $\max(i_e(y), yi) \leqslant i_s(x)$, [$yc \leqslant f_e(z)$, $yi \leqslant i_e(z)$ для всех $z \in E$, таких, что $y \in H_E(z)$], [$f_e(z) \leqslant yc$, $i_e(z) \leqslant yi$ для всех $z \in E$, таких, что $z \in H_E(y)$], [если $i_e(y) = i\_high$ или $yi = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$]
Результаты применения: $f'_e(y) = yc$, $i'_e(y) = yi$
Правило: *set_role_labels*(x, x', r, rc, re)
Условия применения: $x, x' \in S$, $r \in (R \cup AR) \setminus (\{u\_admin\_li : u \in U, li \leqslant i_u(u)\} \cup \{u\_c\_l\_li : u \in U, l \leqslant f_u(u), li \leqslant i_u(u)\} \cup ADMIN\_ROLES)$, $(x, downgrade\_admin\_role, read_a) \in AA$, [если $r \in R$, то $(x, roles\_admin\_role, read_a) \in AA$, если $r \in AR$, то $(x, admin\_roles\_admin\_role, read_a) \in AA$], $\max(f_r(r), rc) \in f_s(x)$, $\{(s, r, \alpha_a) : s \in S, \alpha_a \in R_a\} \cap AA = \varnothing$, [$rc \leqslant f_r(rz)$ для всех $rz \in R \cup AR$, таких, что $r \in H_R(rz)$], [$f_r(rz) \leqslant rc$ для всех $rz \in R \cup AR$, таких, что $rz \in H_R(r)$], $re \subset RE$, [для $e \in re$ выполняется $f_e(e) = rc$, $i_e(e) = i_r(r)$], [если $i_r(r) = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$]
Результаты применения: $f'_r(r) = rc$, $]r[= re$

Рассмотрим условия и результаты применения де-юре правил преобразования состояний.

Де-юре правила $create\_user(x, x', u, uc, ui, ue)$, $set\_user\_labels(x, x', u, uc, ui, ue)$, $delete\_user(x, x', u)$ и $get\_user\_attr(x, u, z)$ позволяют субъект-сессии x создать, изменить уровни конфиденциальности и целостности, удалить или получить параметры учётной записи пользователя u с уровнем доступа, равным текущему уровню доступа субъект-сессии x (за исключением случая, когда субъект-сессия x обладает административным доступом на чтение к специальной административной роли $downgrade\_admin\_role$), и уровнем целостности, не превосходящим текущего уровня целостности субъект-сессии x (кроме случая получения параметров). Во всех случаях, кроме получения параметров, требуется наличие у субъект-сессии x доступа на чтение к специальной административной роли $users\_admin\_role$, а также в последующем состоянии системы — выполнение ограничений на значения множеств административных прав доступа административных ролей $Constraint_{APA}$ (с учётом создания или удаления индивидуальных административных ролей и индивидуальных ролей учётной записи пользователя u), а если осуществляются действия над учётной записью пользователя u с высоким уровнем целостности, то необходимо обладание некоторой кооперирующей с x (либо самой x) субъект-сессией x' доступом на запись к сущности $f_s(x)\_i\_entity$, что моделирует ситуацию подачи сигнала системе (нажатия соответствующей кнопки), подтверждающего переход на высокий уровень целостности. При этом в последующем состоянии иерархия ролей модифицируется (создаются или удаляются индивидуальные роли или индивидуальные административные роли, назначаются или удаляются административные права доступа к ним).

Следует заметить, что эти действия осуществляются без явного получения субъект-сессией x административного доступа на запись ко всем административным ролям, административные права доступа которых модифицируются, что сделано, во-первых, для удобства реализации правил в реальной защищённой ОС, во-вторых, для использования правил требуются специальные административные роли $roles\_admin\_role$ и $admin\_roles\_admin\_role$, доступные только доверенным субъект-сессиям, не участвующим в реализации запрещённых информационных потоков по времени. Для создания, удаления и изменения уровней конфиденциальности и целостности учётной записи пользователя требуется наличие у субъект-сессии x доступов на чтение и запись к специальным административным ролям $roles\_admin\_role$ и $admin\_roles\_admin\_role$, так как именно эти роли получают права доступа владения к создаваемым при применении правил индивидуальным административным ролям и индивидуальным ролям учётной записи пользователя u, при этом (в том числе при изменении параметров) задаются множества сущностей, параметрически ассоциированных с учётной записью пользователя u.

В случае изменения параметров учётной записи пользователя или её удаления требуется, чтобы в этот момент времени от её имени в системе не функционировала ни одна субъект-сессия. При получении параметров в сущность z, к которой субъект-сессия x должна иметь доступ на запись, всегда записываются уровни доступа и целостности учётной записи пользователя u (т. е. если в системе не предъявляются дополнительные требования к анонимности, то самые общие данные о зарегистрированных учётных записях пользователей должны предоставляться всем субъект-сессиям с соответствующим уровнем доступа), а также, когда x функционирует от имени u или обладает текущим административным доступом на чтение к роли $users\_admin\_role$, записываются роли (с учётом их уровней конфиденциальности и уровня доступа x) и права доступа к ним, которыми обладают индивидуальные административные роли u, и записываются субъект-сессии (в реальной защищённой ОС — идентификаторы

субъект-сессий с учётом их уровней доступа и уровня доступа x), функционирующие от имени u (в этом случае полные данные об учётной записи пользователя предоставляются либо субъект-сессии, функционирующей от её имени, либо субъект-сессии, которая может администрировать эту учётную запись).

Де-юре правила $grant\_rights(x, x', r, \{(y, \alpha_{r_j}) : 1 \leqslant j \leqslant k\})$ и $remove\_rights(x, x', r, \{(y, \alpha_{r_j}) : 1 \leqslant j \leqslant k\})$ позволяют субъект-сессии x добавить или удалить соответственно права доступа к сущности y, имеющей уровень целостности не выше, чем у x, из множества прав доступа (за исключением права доступа владения) роли или административной роли r. Возможность с использованием правил одновременного изменения нескольких прав доступа к одной сущности соответствует возможностям типовых для реальных защищённых ОС функций администрирования прав доступа (например, функции *chmod*). Для применения правил необходимо наличие у субъект-сессии x доступа на запись к роли r и наличие текущей роли, обладающей правом доступа владения к сущности y. Во всех случаях (за исключением наличия у субъект-сессии x текущей административной роли $downgrade\_admin\_role$) требуется, чтобы x могла получить доступ к сущности y с учётом прав доступа к сущностям-контейнерам, содержащим y, и чтобы уровень конфиденциальности y равнялся уровню доступа x. Кроме того, если осуществляется добавление права доступа на запись, то требуется, чтобы уровень целостности сущности y не превосходил уровня целостности роли r. Также необходимо выполнение ограничений на значения множеств прав доступа ролей или административных ролей $Constraint_{PA}$ в последующем состоянии системы (с учётом изменения у роли r прав доступа к y). При этом в случае, когда уровень целостности сущности y равняется i_high, требуется наличие у кооперирующей субъект-сессии x' доступа на запись к сущности $f_s(x)\_i\_entity$.

Де-юре правила $set\_entity\_owner(x, x', r, r', y)$ и $set\_subject\_owner(x, x', r, r', y)$ позволяют субъект-сессии x изменить единственную роль-«владельца» (имеющую право доступа владения) к сущности или субъект-сессии y соответственно с роли или административной роли r на роль или административную роль r'. Для этого субъект-сессии x необходимо иметь административные доступы на чтение и запись к r и на запись к r' (чтобы иметь возможность менять права доступа данных ролей), а также иметь административный доступ на чтение соответственно либо к административной роли $entities\_admin\_role$, либо к $subjects\_admin\_role$. При этом уровень целостности y не должен превосходить уровней целостности r' и x и в последующем состоянии системы должны выполняться ограничения на значения множеств прав доступа ролей или административных ролей $Constraint_{PA}$ (с учётом изменения прав доступа ролей r и r'). За исключением случая наличия у субъект-сессии x текущей административной роли $downgrade\_admin\_role$, когда изменяется роль-«владелец» сущности y, требуется, чтобы x могла получить доступ к сущности y с учётом прав доступа к сущностям-контейнерам, содержащим y, и чтобы уровень конфиденциальности y равнялся уровню доступа x; когда изменяется роль-«владелец» субъект-сессии y, требуется, чтобы x и y имели равные уровни доступа. При этом в случае, когда уровень целостности y равняется i_high, требуется наличие у кооперирующей субъект-сессии x' доступа на запись к сущности $f_s(x)\_i\_entity$.

Де-юре правила $grant\_admin\_rights(x, x', ar, \{(r, \alpha_{r_j}) : 1 \leqslant j \leqslant k\})$ и $remove\_admin\_rights(x, x', ar, \{(r, \alpha_{r_j}) : 1 \leqslant j \leqslant k\})$ позволяют субъект-сессии x добавить или удалить соответственно права доступа на чтение или запись к роли или административной роли r, имеющей уровень целостности не выше, чем у x, из множества прав доступа административной роли ar, к которой x должна иметь административный до-

ступ на запись. Для изменения прав доступа к роли r требуется наличие у x текущей административной роли $roles\_admin\_role$, а если r является административной ролью, то к роли $admin\_roles\_admin\_role$. Во всех случаях (за исключением наличия у субъект-сессии x текущей административной роли $downgrade\_admin\_role$) требуется, чтобы уровень конфиденциальности r равнялся уровню доступа x. Также необходимо выполнение ограничений на значения множеств административных прав доступа административных ролей $Constraint_{APA}$ в последующем состоянии системы (с учётом изменения у административной роли ar прав доступа к r). При добавлении прав доступа к роли r её уровень целостности должен быть не выше уровня целостности административной роли ar. В случае, когда уровень целостности роли или административной роли r равняется i_high, требуется наличие у кооперирующей субъект-сессии x' доступа на запись к сущности $f_s(x)\_i\_entity$.

Де-юре правила $create\_role(x, x', r, rc, ri, re, name, rz)$, $create\_hard\_link\_role(x, x', r, name, rz)$, $delete\_role(x, x', r, rz)$ и $delete\_hard\_link\_role(x, x', r, rz)$ позволяют субъект-сессии x создать или удалить роль или административную роль r или «жёсткую» ссылку на неё (изменить иерархию ролей), входящую в состав роли или административной роли rz, к которой субъект-сессия x должна иметь доступ на запись. При этом нельзя удалять или создавать роли или административные роли, «жёсткие» ссылки на них, являющиеся индивидуальными ролями и индивидуальными административными ролями учётных записей пользователей, а также специальными административными ролями из множества $ADMIN\_ROLES$. Для применения правил необходимо наличие у субъект-сессии x административных доступов на чтение и запись к административным ролям $roles\_admin\_role$ или $admin\_roles\_admin\_role$ для действий ролями или административными ролями соответственно. При этом доступ на запись к этим административным ролям следует требовать, так как создание, удаление ролей или «жёстких» ссылок на них являются существенными преобразованиями параметров безопасности системы. При реализации правил в последующем состоянии иерархия ролей модифицируется (назначаются или удаляются административные права доступа), при этом должны выполняться ограничения на значения множеств административных прав доступа административных ролей $Constraint_{APA}$, а при удалении ролей или административных ролей — на значение множества административных доступов $Constraint_{AA}$.

Аналогично правилам администрирования учётных записей пользователей, при реализации правил создания или удаления ролей права доступа к ним могут изменяться у административных ролей без явного получения к ним субъект-сессией x административного доступа на запись. Так же, как при удалении сущностей или «жёстких» ссылок на них, при удалении роли или административной роли проверяется, что ей в иерархии не подчинены другие роли, а при удалении «жёсткой» ссылки на роль — что эта ссылка не последняя. Однако при создании «жёсткой» ссылки на роль (так как, в отличие от сущностей, могут создаваться «жёсткие» ссылки на роли-контейнеры, содержащие подчинённые роли) проверяется, что это не приведёт к возникновению в иерархии циклов, т. е. нарушению отношения частичного порядка. При применении правил уровень конфиденциальности роли или административной роли r должен равняться уровню доступа субъект-сессии x (за исключением случая, когда x имеет административный доступ на чтение к административной роли $downgrade\_admin\_role$), уровень целостности r должен быть не выше уровня целостности x, а при создании роли или «жёсткой» ссылки на неё уровень конфиденциальности роли r должен равняться уровню конфиденциальности роли-контейнера rz (за исключением случая, когда x имеет ад-

министративный доступ на чтение к административной роли $downgrade\_admin\_role$), в котором она создаётся, и уровень целостности r не должен превосходить уровня целостности rz. При создании роли или административной роли уровни конфиденциальности и целостности сущностей, параметрически ассоциированных с нею, устанавливаются равными её уровням конфиденциальности и целостности соответственно. Если осуществляются действия над ролями или административными ролями с высоким уровнем целостности i_high, то требуется наличие у кооперирующей субъект-сессии x' доступа на запись к сущности $f_s(x)\_i\_entity$.

Де-юре правило $set\_container\_attr(x, x', y, ccr, ccri, t)$ позволяет субъект-сессии x задать сущности-контейнеру y, обладающей уровнем целостности не выше, чем у x, либо её мандатный атрибут конфиденциальности CCR, либо уровень целостности $CCRI$, либо признак — является она разделяемой или нет. Для этого требуется наличие у субъект-сессии административного доступа на чтение к роли $entities\_admin\_role$ или $downgrade\_admin\_role$. При этом субъект-сессия x должна иметь либо административный доступ на чтение к роли-«владельцу» контейнера y, либо к административной роли $entities\_admin\_role$, а также требуется (за исключением случая, когда x имеет административный доступ на чтение к административной роли $downgrade\_admin\_role$), чтобы доступ к y мог быть предоставлен x с учётом её прав доступа к сущностям-контейнерам, содержащим y, и уровень доступа x равнялся уровню конфиденциальности y. Если сущность-контейнер обладает высоким уровнем целостности i_high, то требуется наличие у кооперирующей субъект-сессии x' доступа на запись к сущности $f_s(x)\_i\_entity$.

Де-юре правило $set\_entity\_labels(x, x', y, yc, yi)$ позволяет субъект-сессии x задать сущности y (к которой на момент применения правила ни одна субъект-сессия системы не имеет доступов) её новые уровни конфиденциальности и целостности, для чего требуется наличие у x административного доступа на чтение к административным ролям $entities\_admin\_role$ и $downgrade\_admin\_role$. При этом текущие и устанавливаемые уровни конфиденциальности и целостности сущности y не должны превосходить соответственно текущих уровней доступа и целостности субъект-сессии x. Кроме того, устанавливаемые уровни конфиденциальности и целостности сущности y не должны превосходить соответственно уровней конфиденциальности и целостности всех сущностей-контейнеров, в которых непосредственно находится y, и наоборот, должны быть не ниже уровней конфиденциальности и целостности соответственно всех сущностей, непосредственно входящих в y. Если либо текущий, либо устанавливаемый уровень целостности сущности y является высоким уровнем целостности i_high, то требуется наличие у кооперирующей субъект-сессии x' доступа на запись к сущности $f_s(x)\_i\_entity$.

Де-юре правило $set\_role\_labels(x, x', r, rc, re)$ позволяет субъект-сессии x задать роли или административной роли y (не являющейся индивидуальной ролью или индивидуальной административной ролью учётных записей пользователей, а также специальной административной ролью из множества $ADMIN\_ROLES$, и к которой на момент применения правила ни одна субъект-сессия системы не имеет доступов) её новый уровень конфиденциальности, для чего требуется наличие у x административного доступа на чтение к административным ролям $roles\_admin\_role$, $admin\_roles\_admin\_role$ соответственно и к административной роли $downgrade\_admin\_role$. При этом текущие и устанавливаемые уровни конфиденциальности роли r не должны превосходить соответственно текущего уровня доступа субъект-сессии x. Кроме того, устанавливаемый уровень конфиденциальности

роли r не должен превосходить уровня конфиденциальности всех ролей или административных ролей, в которых непосредственно находится r, и наоборот, должен быть не ниже уровней конфиденциальности всех ролей или административных ролей, непосредственно входящих в r. При применении правила уровни конфиденциальности и целостности сущностей, параметрически ассоциированных с ролью или административной ролью r, должны быть соответственно равными уровням конфиденциальности и целостности r. Если уровень целостности роли r является высоким уровнем целостности i_high, то требуется наличие у кооперирующей субъект-сессии x' доступа на запись к сущности $f_s(x)\_i\_entity$.

Заключение

Разработанные в рамках МРОСЛ ДП-модели де-юре правила администрирования системы позволяют задать детальные спецификации функций механизма управления доступом защищённой ОС *Astra Linux Special Edition*, что, в свою очередь, создаёт предпосылки для осуществления как минимум полуформальной верификации реализации модели непосредственно в программном коде ОС. Кроме того, описание рассмотренных правил позволило выявить ряд неточностей при определении элементов модели, что в целом повысило её качество.

ЛИТЕРАТУРА

1. *Девянин П. Н.* Модели безопасности компьютерных систем. Управление доступом и информационными потоками: учебн. пособие для вузов. 2-е изд., испр. и доп. М.: Горячая линия-Телеком, 2013. 338 с.
2. *Девянин П. Н.* Корректность правил преобразования состояний системы в рамках мандатной сущностно-ролевой ДП-модели ОС семейства Linux // Прикладная дискретная математика. Приложение. 2013. № 6. С. 58–59.
3. *Девянин П. Н.* Об опыте внедрения мандатной сущностно-ролевой ДП-модели управления доступом и информационными потоками в защищенную ОС Astra Linux Special Edition // Методы и технические средства обеспечения безопасности информации. Материалы 22-й науч.-технич. конф. 08–11 июля 2013 г. СПб.: Изд-во Политехн. ун-та, 2013. С. 78–80.
4. Операционные системы Astra Linux. `http://www.astra-linux.ru/`.