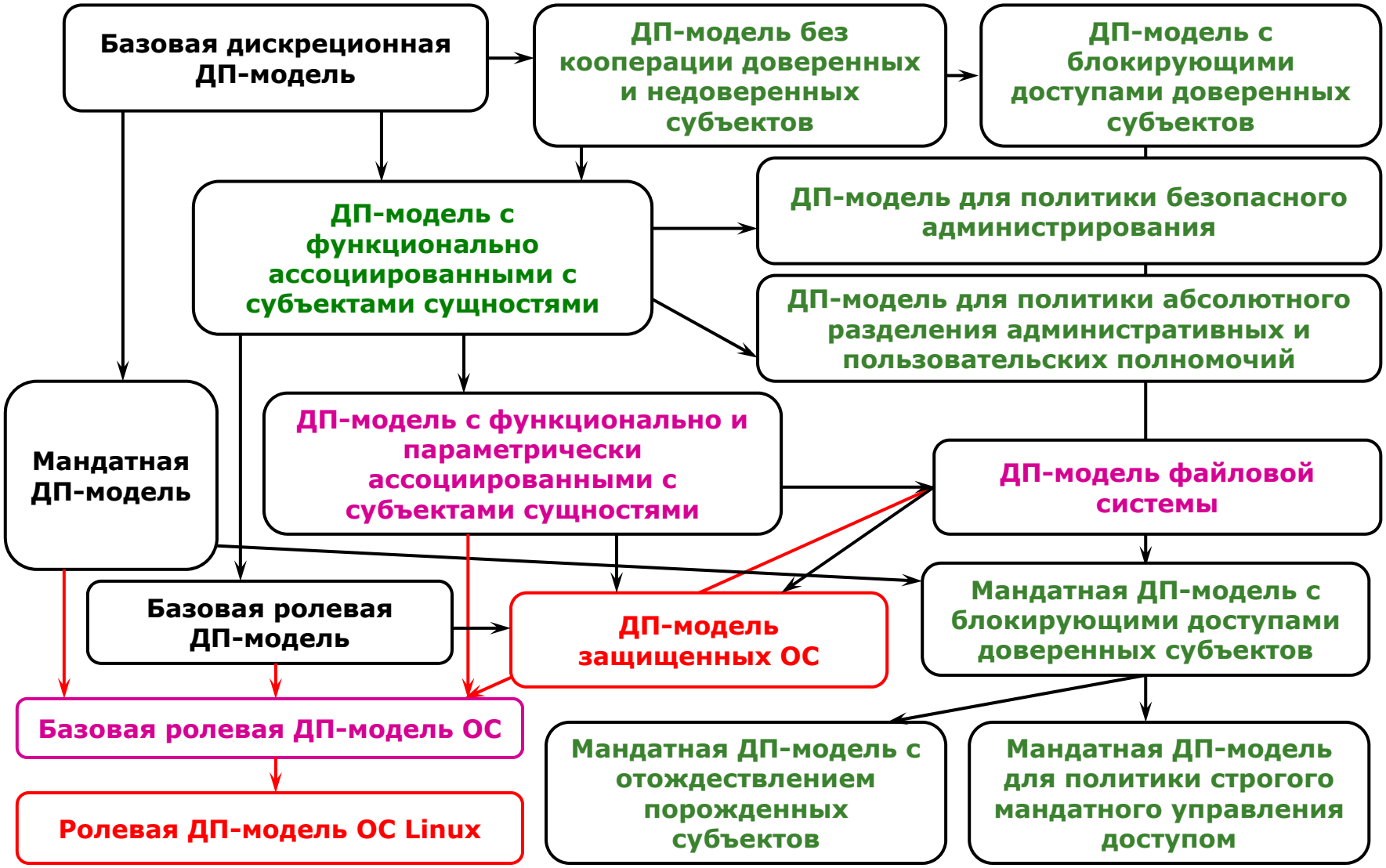
Подходы к разработке формальных моделей управления доступом в защищенных операционных системах семейства *Linux*

д.т.н., доцент Девянин П.Н.
peter_devyanin@hotmail.com
к.т.н., доцент Проскурин В.Г.
vadim_proskurin@hotmail.com
ИКСИ, г. Москва

Семейство ДП-моделей



ЗОС ДП-модель. Основные элементы

$E = O \cup C$ — множество сущностей, где O — множество объектов, C — множество контейнеров;

U — множество учетных записей пользователей;

$]u[\subset E \setminus S$ — множество сущностей, параметрически ассоциированных с учетной записью пользователя $u \in U$;

L_U — множество учетных записей доверенных пользователей;

N_U — множество учетных записей недоверенных пользователей;

$S \subseteq E$ — множество субъектов, функционирующих от имени учетных записей пользователей;

L_S — множество доверенных субъектов;

N_S — множество недоверенных субъектов;

$R_r = \{read_r, write_r, execute_r, own_r, grant_r\}$ — множество видов прав доступа;

$R_a = \{read_a, write_a, own_a\}$ — множество видов доступа;

$R_f = \{write_m, write_t\}$ — множество видов информационных потоков;

$R \subseteq U \times (E \cup U) \times R_r$ — множество прав доступа учетных записей пользователей к сущностям или учетным записям;

$Share = \{read_a, write_a\} \subset R_a$ — множество видов совместных доступов к сущностям;

$A \subseteq S \times E \times R_a \times 2^{Share}$ — множество доступов субъектов к сущностям;

$sa: E \setminus S \to 2^{Share}$ — функция, задающая текущие доступы из множества $Share$ к сущностям;

$sm: E \setminus S \to 2^{Share}$ — функция, задающая разрешенные совместные доступы из множества $Share$ к сущностям

Основные элементы. Уровни целостности

$F \subseteq E \times E \times R_f$ — множество информационных потоков;

$](e, r)[\subset E \setminus S$ — множество сущностей-параметров права доступа $r \in R_r$ к сущности $e \in E$;

$user: S \to U$ — функция принадлежности субъекта учетной записи пользователя;

$[s] \subset E \cup U$ — множество сущностей, функционально ассоциированных с субъектом s;

$fa: U \times E \to 2^E \cup 2^U$ — функция сущностей, функционально ассоциированных с субъектом при его создании от имени учетной записи пользователя из сущности;

$]s[\subset E \setminus S$ — множество сущностей, параметрически ассоциированных с субъектом $s \in S$;

$fp: U \times E \to 2^E$ — функция сущностей, параметрически ассоциированных с субъектом при его создании из сущности от имени учетной записи пользователя;

$H_E: E \to 2^E$ — функция иерархии сущностей;

$(LI, \leq)$ — линейная шкала двух уровней целостности данных, где $LI = \{i\_low, i\_high\}$, $i\_low < i\_high$;

$i_u: U \to LI$ — функция уровней целостности учетных записей пользователей;

$i_e: E \setminus S \to LI$ — функция уровней целостности сущностей;

$i_s: S \to LI$ — функция уровней целостности субъектов;

$G = (U, S, E, user, (i_u, i_e, i_s), R, A, F, H_E, L_U, L_S)$ — состояние системы;

$G = (S, E, user, R, A, F, H_E)$ — сокращенное обозначение для состояния системы;

$\Sigma(G^*, OP)$ — система, при этом G^* — множество всех возможных состояний, OP — множество правил преобразования состояний, $G \vdash_{op} G'$ — переход системы из состояния G в состояние G' с использованием правила преобразования состояний;

Требования к управлению доступом и мандатному контролю целостности

Требования к распределению прав доступа и получению доступов:

- для учетных записей пользователей $u, u' \in U$, если $(u, u', \alpha_r) \in R$, то $\alpha_r \in \{own_r, grant_r\}$;
- для учетной записи пользователя $u \in U$ и субъекта $s \in S$, если $(u, s, \alpha_r) \in R$, то $\alpha_r = own_r$;
- для субъектов $s, s' \in S$, если $(s, s', \alpha_a, \gamma) \in A$, то $\alpha_a = own_a$, $\gamma = \varnothing$;
- для учетной записи пользователя $u \in U$ и сущности $e \in E \setminus S$, если $(u, e, \alpha_r) \in R$, то $\alpha_r \in \{read_r, write_r, execute_r, own_r\}$.

Требования к мандатному контролю целостности:

- для сущностей $e, e' \in E \setminus S$, если $e \le e'$, то $i_e(e) \le i_e(e')$;
- для субъектов $s, s' \in S$, если $s \le s'$, то $i_s(s) \le i_s(s')$;
- для каждой сущности $e \in]u[$, где $u \in U$, справедливо равенство $i_e(e) = i_u(u)$;
- для субъекта $s \in S$ верно неравенство $i_s(s) \le i_u(user(s))$;
- для учетной записи доверенного пользователя $u \in L_U$ выполняется $i_u(u) = i\_high$;
- для учетной записи недоверенного пользователя $u \in N_U$ выполняется $i_u(u) = i\_low$;
- верно равенство $i_e(i\_entity) = i\_high$.

Контроль над субъектами

Если субъект s реализовал информационный поток по памяти от себя к сущности, функционально ассоциированной с субъектом s', или реализовал информационный поток по памяти к себе от всех сущностей, параметрически ассоциированных с субъектом s', то субъект s получает доступ владения own_a к субъекту s'. Кроме того, субъект s получает:

- возможность использовать права доступа учетной записи пользователя субъекта s';
- возможность изменять множество прав доступа учетной записи пользователя субъекта s';
- возможность использовать текущий уровень целостности субъекта s';
- возможность использовать доступы субъекта s';
- возможность получать доступ владения own_a к субъектам, доступом владения к которым обладает субъект s';
- возможность использовать информационные потоки, в реализации которых участвует субъект s'.

$de\_facto\_rights: S \to 2^{E \times Rr}$ — функция фактических текущих прав доступа субъектов, при этом по определению в каждом состоянии системы $G = (S, E, user, R, A, F, H_E)$ для каждого субъекта $s \in S$ по определению верно равенство:

$de\_facto\_rights(s) = \{(e, r) \in E \times R_r: (user(s), e, r) \in R\} \cup \{(e, r) \in E \times R_r:$ существует $s' \in S$ такой, что $(s, s', own_a, \varnothing) \in A$ и $(user(s'), e, r) \in R\}$;

$de\_facto\_accesses: S \to 2^A$ — функция фактических доступов субъектов, при этом по определению в каждом состоянии системы $G = (S, E, user, R, A, F, H_E)$ для каждого субъекта $s \in S$ по определению верно равенство:

$de\_facto\_accesses(s) = \{(s, e, \alpha_a, \varnothing): (s, e, \alpha_a, \varnothing) \in A\} \cup \{(s', e, \alpha_a, \varnothing): (s, s', own_a, \varnothing), (s', e, \alpha_a, \varnothing) \in A\}$.

Правила преобразования состояний

Определение. Монотонное правило преобразования состояний — правило преобразования состояний из множества *OP*, применение которого не приводит к удалению из состояний:

- прав доступа учетных записей пользователей к субъектам или к сущностям;
- субъектов или сущностей;
- доступов субъектов к сущностям;
- информационных потоков.

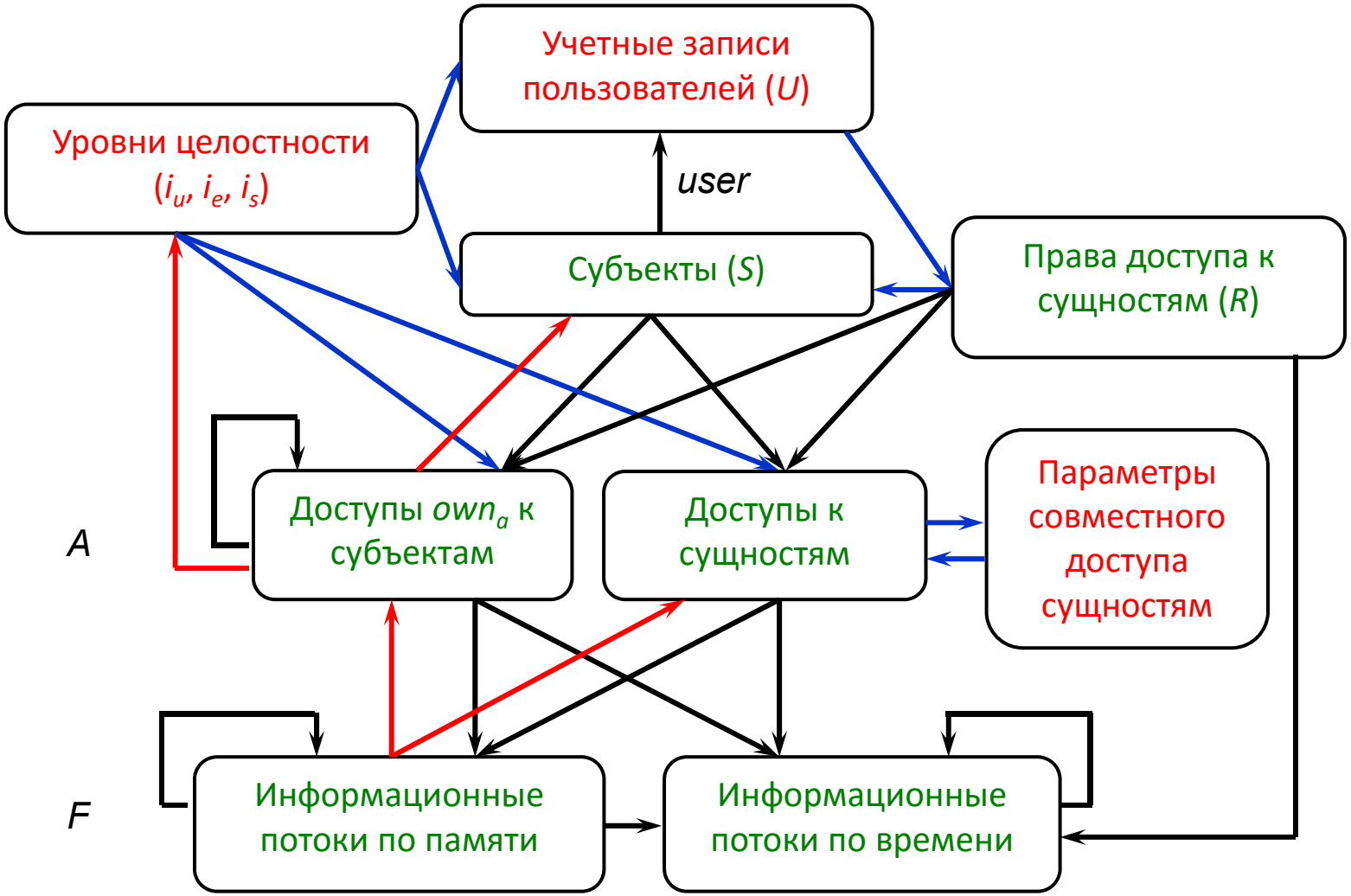
Монотонные правила: *grant_right*(α_r, *x*, *u*, *w*, *z*), *own_take*(α_r, *x*, *w*, *z*), *create_entity*(*x*, *w*, *y*, *z*, *zi*), *rename_entity*(*x*, *w*, *y*, *z*), *create_first_subject*(*x*, *u*, *w*, *y*, *z*, *zi*), *create_subject*(*x*, *w*, *y*, *z*, *zi*), *control*(*x*, *y*, *z*), *know*(*x*, *y*), *take_access_own*(*x*, *y*, *z*), *access_own*(*x*, *w*, *y*), *access_read*(*x*, *w*, *y*, γ), *access_write*(*x*, *w*, *y*, γ), *flow*(*x*, *y*, *y'*, *z*), *find*(*x*, *y*, *z*), *post*(*x*, *y*, *z*), *pass*(*x*, *y*, *z*), *take_flow*(*x*, *y*).

Немонотонные правила: *remove_right*(α_r, *x*, *u*, *w*, *z*), *delete_entity*(*x*, *w*, *y*, *z*), *delete_subject*(*x*, *w*, *z*), *delete_access*(*x*, *w*, *y*, α_a).

Примеры задания правил преобразования состояний

Правило	Исходное состояние $G = (S, E, user, R, A, F, H_E)$	Результирующее состояние $G' = (S', E', user', R', A', F', H_E')$
$grant\_right(\alpha_r, x, u, w, z)$	$x, w \in S$, $u \in U$, $z \in E$, $(w, z, own_a, \varnothing) \in A$, выполняется $(user(w), u, \beta_r) \in R$, где $\beta_r \in \{own_r, grant_r\}$, выполняется или $x = w$, или $(x, w, own_a, \varnothing) \in A$, если $z \in S$, то $\alpha_r = own_r$ и $i_s(z) \le i_u(u)$, если $z \in E \setminus S$, то $\alpha_r \in \{read_r, write_r, execute_r, own_r\}$, при этом, если $\alpha_r \in \{ write_r, own_r\}$, то $i_e(z) \le i_u(u)$, для $e \in](z, \alpha_r)[$ выполняется условие $(e, x, write_m) \in F$, если $i_e(z) = i\_high$, то $(x, i\_entity, write_m) \in F$	$S' = S, E' = E, user' = user, A' = A, H_E' = H_E$, $R' = R \cup \{(u, z, \alpha_r)\}$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, s, write_t)$: $s \in (N_S \cup NF_S) \cap S, x \ne s$ и или $s' = s$, или $(s, s', own_a, \varnothing) \in A$, где или $user(s') = u$, или $(s', z, own_a, \varnothing) \in A\}$, если $x \in LF_S \cap S$, то $F' = F$
$access\_write(x, w, y, \gamma)$	$x, w \in S$, $y \in E \setminus S$, $(user(w), y, write_r) \in R$, $write_a \in sm(y)$, $\gamma \subset sa(y)$, выполняется или $x = w$, или $(x, w, own_a, \varnothing) \in A$, $i_e(y) \le i_s(w)$, если $i_e(y) = i\_high$, то $(x, i\_entity, write_m) \in F$	$S' = S, E' = E, user' = user, R' = R, H_E' = H_E, A' = A \cup \{(w, y, write_a, \gamma)\}$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(w, y, write_m)\} \cup \{(x, e, write_t)$: $e \in E, x \ne e$ и $y \le e\}$, если $x \in LF_S \cap S$, то $F' = F \cup \{(w, y, write_m)\}$
$control(x, y, z)$	$x, y \in S$, $x \ne y$, $z \in [y]$ и или $x = z$, или $(x, z, write_m) \in F$, или $z \in S$ и $(x, z, own_a, \varnothing) \in A$	$S' = S, E' = E, user' = user, R' = R, H_E' = H_E$, $A' = A \cup \{(x, y, own_a, \varnothing)\}$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t)$: $e \in E, x \ne e$ и $y \le e\}$, если $x \in LF_S \cap S$, то $F' = F$
$find(x, y, z)$	$x, y \in S$, $z \in E$, $x \ne z$, $(x, y, \alpha) \in F$, где $\alpha \in \{write_m, write_t\}$, и или $(z, \beta) \in de\_facto\_accesses(y)$, где $\beta = write_a$, или $(y, z, \beta) \in F$, где $\beta \in \{write_m, write_t\}$	$S' = S, E' = E, user' = user, R' = R, A' = A, H_E' = H_E$, если $write_t \notin \{\alpha, \beta\}$, то $F' = F \cup \{(x, z, write_m)\}$, если $write_t \in \{\alpha, \beta\}$ и $x, y \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, z, write_t)\}$, если $write_t \in \{\alpha, \beta\}$ и $\{x, y\} \cap (LF_S \cap S) \ne \varnothing$, то $F' = F$

Зависимость условий и результатов правил



РОСЛ ДП-модель. Предположения

Предположение 1. В рамках РОСЛ ДП-модели пользователям соответствуют их учетные записи, для каждой из которых задаются множества авторизованных ролей. Каждая учетная запись пользователя вне зависимости от имеющихся у нее авторизованных ролей является учетной записью либо доверенного, либо недоверенного пользователя. Каждая субъект-сессия является либо доверенной, либо недоверенной, и функционирует от имени учетной записи доверенного или недоверенного пользователя, соответственно. Доверенные субъект-сессии не инициируют создание субъект-сессий. Каждая недоверенная субъект-сессия может создать недоверенную субъект-сессию.

Предположение 2. Для каждой учетной записи пользователя задается множество сущностей, параметрически с ней ассоциированных, реализация от каждой из которых информационного потока по памяти к субъект-сессии позволяет ей создать субъект-сессию от имени данной учетной записи пользователя.

Предположение 3. Функционально ассоциированными с субъект-сессией являются сущности, от которых зависит вид преобразования данных, реализуемого субъект-сессией. Только информационный поток по памяти к сущности, функционально ассоциированной с субъект-сессией, приводит к изменению вида преобразования данных, реализуемого этой субъект-сессией.

Предположение 4. Параметрически ассоциированными сущностями с субъект-сессией являются сущности, которые содержат данные, позволяющие идентифицировать вид преобразования данных, реализуемого субъект-сессией.

Предположение 5. Для каждой роли или административной роли задается (возможно, пустое) множество сущностей, параметрически с ней ассоциированных. При этом для получения или удаления роли из множества текущих ролей субъект-сессии необходимо реализовать к себе информационные потоки по памяти от всех сущностей, параметрически ассоциированных с данной ролью.

Основные элементы

$E = O \cup C$ — множество сущностей, где O — множество объектов, C — множество контейнеров;

$U \subset 2^E$ — множество учетных записей пользователей;

$]u[\subset E \setminus S$ — множество сущностей, параметрически ассоциированных с учетной записью пользователя $u \in U$;

$UE = \{e \in]u[: u \in U\}$ — множество сущностей, каждая из которых параметрически ассоциирована хотя бы с одной учетной записью пользователя;

L_U — множество учетных записей доверенных пользователей;

N_U — множество учетных записей недоверенных пользователей;

$S \subseteq E$ — множество субъект-сессий учетных записей пользователей;

$L_S \subset S$ — множество доверенных субъект-сессий;

$N_S = S \setminus L_S$ — множество недоверенных субъект-сессий;

R — множество ролей;

AR — множество административных ролей ($AR \cap R = \varnothing$);

$]r[\subset E \setminus S$ — множество сущностей-параметров роли или административной роли $r \in R \cup AR$;

$RE = \{e \in]r[: r \in R \cup AR\}$ — множество сущностей-параметров ролей;

$R_r = \{read_r, write_r, append_r, execute_r, own_r\}$ — множество видов прав доступа;

$R_a = \{read_a, write_a, append_a, own_a\}$ — множество видов доступа;

$R_f = \{write_m, write_t\}$ — множество видов информационных потоков;

Основные элементы

$P \subset E \times R_r$ — множество прав доступа к сущностям;

$A \subseteq S \times E \times R_a$ — множество доступов субъект-сессий к сущностям;

$F \subseteq E \times E \times R_f$ — множество информационных потоков;

$UA: U \to 2^R$ — функция авторизованных ролей учетных записей пользователей;

$AUA: U \to 2^{AR}$ — функция авторизованных административных ролей учетных записей пользователей;

$PA: R \to 2^P \setminus \{\varnothing\}$ — функция прав доступа ролей;

$user: S \to U$ — функция принадлежности субъект-сессии учетной записи пользователя;

$roles: S \to 2^R \cup 2^{AR}$ — функция текущих ролей субъект-сессий;

$can\_manage\_rights: AR \to 2^R$ — функция администрирования прав доступа ролей;

$[s] \subset E \cup U$ — множество всех сущностей, функционально ассоциированных с субъект-сессией s;

$fa: U \times E \to 2^E \cup 2^U$ — функция, задающая множества сущностей, функционально ассоциированных с субъект-сессией при ее создании от имени учетной записи пользователя;

$]s[\subset E \setminus S$ — множество сущностей, параметрически ассоциированных с субъект-сессией;

$fp: U \times E \to 2^E$ — функция, задающая множества сущностей, параметрически ассоциированных с субъект-сессией при ее создании из сущности от имени учетной записи пользователя;

$H_R: R \to 2^R$ — функция иерархии ролей;

$H_{AR}: AR \to 2^{AR}$ — функция иерархии административных ролей.

Иерархия сущностей и механизм ограничений

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 условиям:

Условие 1. Если сущность $e \in H_E(c)$, то $e < c$, при этом, если $e \in C \cup E$, то не существует сущности-контейнера $d \in C$ такой, что $e < d$, $d < c$.

Условие 2. Для любых сущностей $e_1, e_2 \in E$, $e_1 \neq e_2$, по определению выполняются равенство $H_E(e_1) \cap H_E(e_2) \cap (C \cup E) = \varnothing$ и условия:

- если $o \in O$, то справедливо равенство $H_E(o) = \varnothing$;
- если $e_1 < e_2$, то или $e_1, e_2 \in E \setminus S$, или $e_1, e_2 \in S$;
- если $e \in E \setminus S$, то $H_E(e) \subset E \setminus S$;
- если $s \in S$, то $H_E(s) \subset S$.

C^U: $UA^* \to \{true, false\}$ — функция, задающая ограничение на значения множеств авторизованных ролей учетных записей пользователей, то есть по определению множества авторизованных ролей учетных записей пользователей, заданные функцией $UA \in UA^*$, удовлетворяют ограничению C^U, если выполняется равенство $C^U(UA) = true$;

C^P: $PA^* \to \{true, false\}$ — функция, задающая ограничение на значения множеств прав доступа ролей, то есть по определению множества прав доступа ролей, заданные функцией $PA \in PA^*$, удовлетворяют ограничению C^P, если выполняется равенство $C^P(PA) = true$;

C^S: $roles^* \to \{true, false\}$ — функция, задающая ограничение на значения множеств текущих ролей субъект-сессий, то есть по определению множества текущих ролей субъект-сессий, заданные функцией $roles \in roles^*$, удовлетворяют ограничению C^S, если выполняется равенство $C^S(roles) = true$.

Мандатный контроль целостности

$(LI, \leq)$ — линейная шкала двух уровней целостности данных, где $LI = \{i\_low, i\_high\}$, $i\_low < i\_high$;

$(i_u, i_e, i_r, i_s) \in I$ — четверка функций уровней целостности, при этом:

i_u: $U \to LI$ — функция уровней целостности субъект-сессий;

i_e: $E \setminus S \to LI$ — функция уровней целостности сущностей;

i_r: $R \cup AR \to LI$ — функция уровней целостности ролей;

i_s: $S \to LI$ — функция текущих уровней целостности субъект-сессий;

I — множества всех четверок функций заданного вида;

По предположениям выполняются условия:

- для ролей $r, r' \in R \cup AR$, если $r \leq r'$, то $i_r(r) \leq i_r(r')$;
- для сущностей $e, e' \in E \setminus S$, если $e \leq e'$, то $i_e(e) \leq i_e(e')$;
- для субъект-сессий $s, s' \in S$, если $s \leq s'$, то $i_s(s) \leq i_s(s')$;
- для каждой сущности $e \in u$, где $u \in U$, справедливо равенство $i_e(e) = i_u(u)$;
- для субъект-сессии $s \in S$ верно неравенство $i_s(s) \leq i_u(user(s))$;
- для учетной записи пользователя $u \in U$ и роли $r \in R$, если $r \in UA(u)$, то $i_r(r) \leq i_u(u)$;
- для субъект-сессии $s \in S$ и роли $r \in R$, если $r \in roles(s)$, то $i_r(r) \leq i_s(s)$;
- для права доступа к сущности $(e, \alpha) \in P$, где $\alpha \in \{own_r, write_r, append_r\}$, и роли $r \in R$, если $(e, \alpha) \in PA(r)$, то $i_e(e) \leq i_r(r)$;
- для учетной записи доверенного пользователя $u \in L_U$ справедливо равенство $i_u(u) = i\_high$;
- для учетной записи недоверенного пользователя $u \in N_U$ справедливо равенство $i_u(u) = i\_low$;
- верно равенство $i_e(i\_entity) = i\_high$.

Фактические роли, права доступа, доступы, возможные действия

Предположение 6. Если субъект-сессия *s* реализовала информационный поток по памяти от себя к сущности, функционально ассоциированной с другой субъект-сессией *s'*, или субъект-сессия *s* реализовала информационный поток по памяти к себе от всех сущностей, параметрически ассоциированных с другой субъект-сессией *s'*, то субъект-сессия *s* получает доступ владения own_a к субъект-сессии *s'*.

Предположение 7. Если субъект-сессия *s* имеет доступ владения own_a к субъект-сессии *s'*, то субъект-сессия *s* получает:

- возможность использовать роли из множества текущих ролей субъект-сессии *s'*;
- возможность изменять множество текущих ролей субъект-сессии *s'*;
- возможность использовать текущий уровень целостности субъект-сессии *s'*;
- возможность использовать доступы субъект-сессии *s'*;
- возможность получать доступ владения own_a к субъект-сессиям, доступом владения к которым обладает субъект-сессия *s'*;
- возможность использовать административные роли субъект-сессии *s'*;
- возможность использовать информационные потоки, которые реализует субъект-сессия *s'*.

Используем обозначения:

- $de\_facto\_roles: S \to 2^{R \cup AR}$ — фактические текущие роли субъект-сессий;
- $de\_facto\_rights: S \to 2^{P}$ — фактические текущие права доступа субъект-сессий;
- $de\_facto\_accesses: S \to 2^{A}$ — фактические доступы субъект-сессий;
- $de\_facto\_actions: S \to S \times U \times 2^{P} \times 2^{R}$ — фактические возможные действия субъект-сессий.

Правила преобразования состояний

$G = (UA, AUA, PA, user, roles, (i_u, i_e, i_r, i_s), A, F, H_R, H_{AR}, H_E, Constraint_U, Constraint_P, Constraint_S, L_U, L_S)$ — состояние системы ($G = (PA, user, roles, A, F, H_E)$ — сокращенное обозначение);

$\Sigma(G^*, OP)$ — система, при этом G^* — множество всех возможных состояний, OP — множество правил преобразования состояний, $G \vdash_{op} G'$ — переход системы из состояния G в состояние G' с использованием правила преобразования состояний;

$\Sigma(G^*, OP, G_0)$ — система $\Sigma(G^*, OP)$ с начальным состоянием G_0.

Определение. Монотонное правило преобразования состояний — правило преобразования состояний из множества OP, применение которого не приводит к удалению из состояний:

- ролей из множества текущих ролей субъект-сессии;
- прав доступа ролей к сущностям;
- субъект-сессий, сущностей или «жестких» ссылок на сущности-объекты;
- доступов субъект-сессий к сущностям;
- информационных потоков.

Монотонные правила: *take_roles*(*x*, *x'*, *w*, $\{r_j: 1 \le j \le k\}$), *grant_rights*(*x*, *x'*, *r*, $\{(y_j, \alpha_{rj}): 1 \le j \le k\}$), *create_object*(*x*, *x'*, *r*, *y*, *yi*, *z*), *create_container*(*x*, *x'*, *r*, *y*, *yi*, *z*), *create_hard_link*(*x*, *x'*, *y*, *z*), *rename_entity*(*x*, *x'*, *y*, *z*), *create_first_session*(*x*, *x'*, *u*, *r*, *y*, *z*, *zi*), *create_session*(*x*, *x'*, *w*, *r*, *y*, *z*, *zi*), *control*(*x*, *y*, *z*), *know*(*x*, *y*), *take_access_own*(*x*, *y*, *z*), *access_own*(*x*, *x'*, *w*, *y*), *access_read*(*x*, *w*, *y*), *access_write*(*x*, *x'*, *w*, *y*), *flow_access*(*x*, *y*), *flow*(*x*, *y*, *y'*, *z*), *find*(*x*, *y*, *z*), *post*(*x*, *y*, *z*), *pass*(*x*, *y*, *z*) и *take_flow*(*x*, *y*).

Немонотонные правила: *remove_roles*(*x*, *x'*, *w*, $\{r_j: 1 \le j \le k\}$), *remove_rights*(*x*, *x'*, *r*, $\{(y_j, \alpha_{rj}): 1 \le j \le k\}$), *delete_entity*(*x*, *x'*, *y*, *z*), *delete_hard_link* (*x*, *x'*, *y*, *z*), *delete_session*(*x*, *x'*, *w*, *z*), *delete_access*(*x*, *x'*, *w*, *y*, α_a).

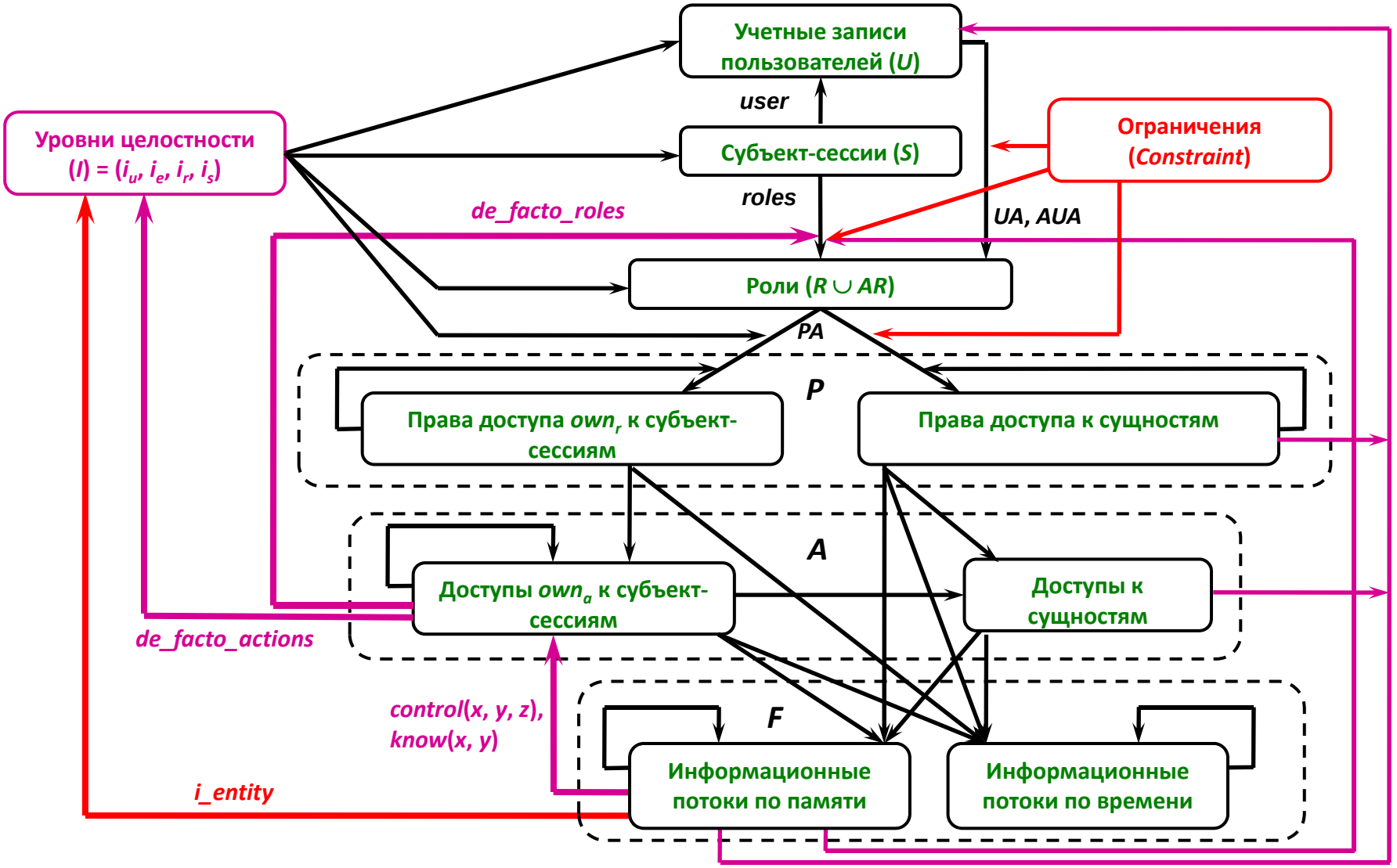
Примеры правил преобразования состояний

Правило	Исходное состояние $G = (PA, user, roles, A, F, H_E)$	Результирующее состояние $G' = (PA', user', roles', A', F', H_E')$
$take\_roles(x, x', w, \{r_j: 1 \le j \le k\})$	$x, x', w \in S$, $(w, user(w), \varnothing, \varnothing) \in de\_facto\_actions(x)$, $r_j \in UA(user(w)) \cup AUA(user(w))$, для $e \in]r_j[$ выполняется условие $(e, x, write_m) \in F$, $i_r(r_j) \le i_s(w)$, $Constraint_S(roles') = true$, если $i_r(r_j) = i\_high$, то $(x', i\_entity, write_m) \in F$, где $1 \le j \le k$	$S' = S, E' = E, PA' = PA, user' = user, A' = A, F' = F, H_E' = H_E$, $roles'(w) = roles(w) \cup \{r_j: 1 \le j \le k\}$ и для $s \in S \setminus \{w\}$ выполняется равенство $roles'(s) = roles(s)$, если $x \in (N_S \cup NF_S) \cap S$ и $\{r_j: 1 \le j \le k\} \setminus roles(w) \ne \varnothing$, то $F' = F \cup \{(x, s, write_t): s \in (N_S \cup NF_S) \cap S, x \ne s$ и или $s = w$, или $(w, own_a) \in de\_facto\_accesses(s)\}$, если $x \in LF_S \cap S$ или $\{r_j: 1 \le j \le k\} \subset roles(w)$, то $F' = F$
$grant\_rights(x, x', r, \{(y_j, \alpha_\eta): 1 \le j \le k\})$	$x, x' \in S$, $y_j \in E$, $(y_j, \alpha_\eta) \in P$, $(w, \varnothing, (y_j, own_r), r) \in de\_facto\_actions(x)$, $(w, y_j, own_a) \in A$, $i_r(r) \le i_s(w)$, [если $y_j \in S$, то $\alpha_\eta = own_r$ и $i_s(y_j) \le i_r(r)$], [если $y_j \in E \setminus S$ и $\alpha_\eta \in \{own_r, write_r\}$, то $i_e(y_j) \le i_r(r)$], $Constraint_P(PA') = true$, если $i_e(y_j) = i\_high$, то $(x', i\_entity, write_m) \in F$, где $1 \le j \le k$,	$S' = S, E' = E, user' = user, roles' = roles$, $A' = A, H_E' = H_E, PA'(r) = PA(r) \cup \{(y_j, \alpha_\eta): 1 \le j \le k\}$, и для $r' \in R \setminus \{r\}$ выполняется равенство $PA'(r') = PA(r')$, если $x \in (N_S \cup NF_S) \cap S$ и $\{(y_j, \alpha_\eta): 1 \le j \le k\} \setminus PA(r) \ne \varnothing$, то $F' = F \cup \{(x, s, write_t): s \in (N_S \cup NF_S) \cap S, x \ne s$ и $r \in de\_facto\_roles(s)\}$, если $x \in LF_S \cap S$ или $\{(y_j, \alpha_\eta): 1 \le j \le k\} \subset PA(r)$, то $F' = F$
$create\_hard\_link(x, x', y, z)$	$x, x' \in S$, $y \in O \setminus S$, $z \in C \setminus S$, $y \notin UE \cup RE$, $(w, \varnothing, (z, write_r), r) \in de\_facto\_actions(x)$, $i_e(y) \le i_r(r) \le i_s(w)$, $i_e(y) \le i_e(z) \le i_s(w)$, если $i_e(z) = i\_high$, то $(x', i\_entity, write_m) \in F$	$S' = S, E' = E, PA' = PA, user' = user, roles' = roles, A' = A$, $H_E'(z) = H_E(z) \cup \{y\}$, для $e \in E \setminus \{z\}$ выполняется равенство $H_E'(e) = H_E(e)$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t): e \in E$ и или $y \le e\}$, если $x \in LF_S \cap S$, то $F' = F$
$create\_first\_session(x, x', u, r, y, z, zi)$	$x, x' \in S, u \in U, y \in E, z \notin E$, $(y, execute_r) \in PA(UA(u))$ и $r \in can\_manage\_rights(AUA(u))$, $zi \le i_u(u)$, $zi \le i_r(r)$, $\{(e, x, write_m): e \in]u[\} \subset F$, $Constraint_P(PA') = true$, $Constraint_S(roles') = true$, если $zi = i\_high$, то $(x', i\_entity, write_m) \in F$	$S' = S \cup \{z\}, E' = E \cup \{z\}, A' = A, i_s'(z) = zi, user'(z) = u$, для $s \in S$ выполняется равенство $user'(s) = user(s), roles'(z) = \varnothing$, для $s \in S$ выполняется равенство $roles'(s) = roles(s), [z] = fa(u, y),]z[= fp(u, y)$, $PA'(r) = PA(r) \cup \{(z, own_r)\}$, и $r' \in R \setminus \{r\}$ выполняется $PA'(r') = PA(r')$, $H_E'(z) = \varnothing$, для $e \in E$ выполняется равенство $H_E'(e) = H_E(e)$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(z, x, write_t), (x, z, write_t)\} \cup \{(x, e, write_t): e \in E$ и $y \le e\} \cup \{(x, s, write_t): s \in (N_S \cup NF_S) \cap S, x \ne s$ и $r \in de\_facto\_roles(s)\}$, если $x \in LF_S \cap S$, то $F' = F$

Примеры правил преобразования состояний

Правило	Исходное состояние $G = (PA, user, roles, A, F, H_E)$	Результирующее состояние $G' = (PA', user', roles', A', F', H_E')$
$know(x, y)$	$x, y \in S$, $x \neq y$, и для каждой $e \in]y[$, существует $(e, x, write_m) \in F$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $H_E' = H_E$, $A' = A \cup \{(x, y, own_a)\}$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t): e \in E, x \neq e$ и $y \leq e\}$, если $x \in LF_S \cap S$, то $F' = F$
$access\_own(x, x', w, y)$	$x, x', w \in S$, $y \in E$, $w \neq y$, $(w, \varnothing, (y, own_r), \varnothing) \in de\_facto\_actions(x)$, [если $y \in S$, то $i_s(y) \leq i_s(w)$], [если $y \in E \setminus S$, то $i_e(y) \leq i_s(w)$], если ($y \in S$ и $i_s(y) = i\_high$) или ($y \in E \setminus S$ и $i_e(y) = i\_high$), то $(x', i\_entity, write_m) \in F$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $H_E' = H_E$, $A' = A \cup \{(w, y, own_a)\}$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t): e \in E, x \neq e$ и $y \leq e\} \cup \{(x, s, write_t): s \in (N_S \cup NF_S) \cap S, x \neq s$ и $(y, own_a) \in de\_facto\_accesses(s)\}$, если $x \in LF_S \cap S$, то $F' = F$
$flow\_access(x, y)$	$x \in S$, $y \in E$, $(y, \alpha_a) \in de\_facto\_accesses(x)$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $H_E' = H_E$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t): e \in E, x \neq e$ и $y \leq e\}$, если $x \in LF_S \cap S$, то $F' = F$
$find(x, y, z)$	$x, y \in S$, $z \in E$, $x \neq z$, $(x, y, \alpha) \in F$, где $\alpha \in \{write_m, write_t\}$, и или $(z, \beta) \in de\_facto\_accesses(y)$, где $\beta = write_a$, или $(y, z, \beta) \in F$, где $\beta \in \{write_m, write_t\}$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $H_E' = H_E$, если $write_t \notin \{\alpha, \beta\}$, то $F' = F \cup \{(x, z, write_m)\}$, если $write_t \in \{\alpha, \beta\}$ и $x, y \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, z, write_t)\}$, если $write_t \in \{\alpha, \beta\}$ и $\{x, y\} \cap (LF_S \cap S) \neq \varnothing$, то $F' = F$
$take\_flow(x, y)$	$x, y \in S$, $x \neq y$, $(x, y, own_a) \in A$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $H_E' = H_E$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, \alpha): (y, e, \alpha) \in F, e \in E, \alpha \in \{write_m, write_t\}\}$, если $x \in LF_S \cap S$, то $F' = F \cup \{(x, e, write_m): (y, e, write_m) \in F, e \in E\}$

Зависимость условий и результатов правил



Инвариантные относительно немонотонных правил ограничения

Определение. Ограничение инвариантно относительно немонотонных правил преобразования состояний в системе $\Sigma(G^*, OP)$, когда при условии, что в системе задано только данное ограничение, для любых состояния системы G_0, немонотонного правила преобразования состояний op_1, правил преобразования состояний $op_2, \ldots, op_N$, где $N > 1$, справедливо следующие: если $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{op(N-1)} G_{N-1}$, $G_0 \vdash_{op2} G'_2 \vdash_{op3} \ldots \vdash_{op(N-1)} G'_{N-2}$, и в состоянии G_{N-1} выполнены ограничения, заданные в условиях применения правила op_N, то эти ограничения выполнены в состоянии G'_{N-2}. Ограничения, заданные в системе $\Sigma(G^*, OP)$, по определению инвариантны относительно немонотонных правил преобразования состояний, когда каждое из ограничений в отдельности инвариантно относительно немонотонных правил преобразования состояний.

Утверждение. Пусть $G_0 = (PA_0, user_0, roles_0, A_0, F_0, H_{E0})$ — начальное состояние системы $\Sigma(G^*, OP, G_0)$, в котором все ограничения инвариантны относительно немонотонных правил преобразования состояний и функции $(i_u, i_e, i_r, i_s)_0$ удовлетворяют условиям предположений. Пусть также существуют состояния системы $G_1, \ldots, G_N = (PA_N, user_N, roles_N, A_N, F_N, H_{EN})$ и правила преобразования состояний $op_1, \ldots, op_N$ такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N$, где $N \geq 0$. Тогда существуют состояния $G'_1, \ldots, G'_M = (PA'_M, user'_M, roles'_M, A'_M, F'_M, H'_{EM})$, где $M \geq 0$, и монотонные правила преобразования состояний $op'_1, \ldots, op'_M$ такие, что $G_0 \vdash_{op'1} G'_1 \vdash_{op'2} \ldots \vdash_{op'M} G'_M$ и выполняются следующие условия.

Условие 1. Верно включение $S_N \subset S'_M$, и для каждой субъект-сессии $s \in S_N$ выполняются условия: $user_N(s) = user'_M(s)$, $roles_N(s) \subset roles'_M(s)$.

Условие 2. Верно включение $E_N \subset E'_M$, для каждой сущности $e \in E_N \setminus S_N$, не являющейся субъектом, выполняется условие $H_{EN}(e) \subset H'_{EM}(e)$, и для любых сущностей $e, e' \in E_N$, если в состоянии G_N выполняется условие $e < e'$, то данное условие выполняется в состоянии G'_M.

Условие 3. Для каждой роли $r \in R$ выполняется условие $PA_N(r) \subset PA'_M(r)$.

Условие 4. Верно включение $A_N \subset A'_M$.

Условие 5. Верно включение $F_N \subset F'_M$.

Условие 6. Функции $(i_u, i_e, i_r, i_s)'_M$ удовлетворяют условиям предположений.

Развитие ДП-моделей операционных систем

- Построение ДП-моделей существующих или перспективных компьютерных систем (особенно операционных систем семейства *Linux*) для формального анализа их безопасности;
- Макетирование ролевого управления доступом на основе ОС *AltLinux* и корректировка формальных моделей;
- Исследование возможности реализации мандатного ролевого управления доступом с использованием динамических ограничений, в том числе, не позволяющих порождать запрещенные информационные потоки по времени, применительно к защищенным операционным системам;
- Разработка алгоритма построения замыкания графа прав доступа, доступов и информационных потоков для проверки корректности теоретических результатов.

Литература по теме доклада

1. Bishop M. Computer Security: art and science. — ISBN 0-201-44099-7, 2002. — 1084 p.
2. Девянин П.Н. Модели безопасности компьютерных систем. Управление доступом и информационными потоками. Учебное пособие для вузов. — М.: Горячая линия — Телеком, 2011. — 320 с.
3. Проскурин В.Г. Защита программ и данных. Учебное пособие для вузов. — М.: Издательский центр «Академия», 2011. — 300 с. (план издательства)
4. Прикладная дискретная математика. — Томск: ТГУ.