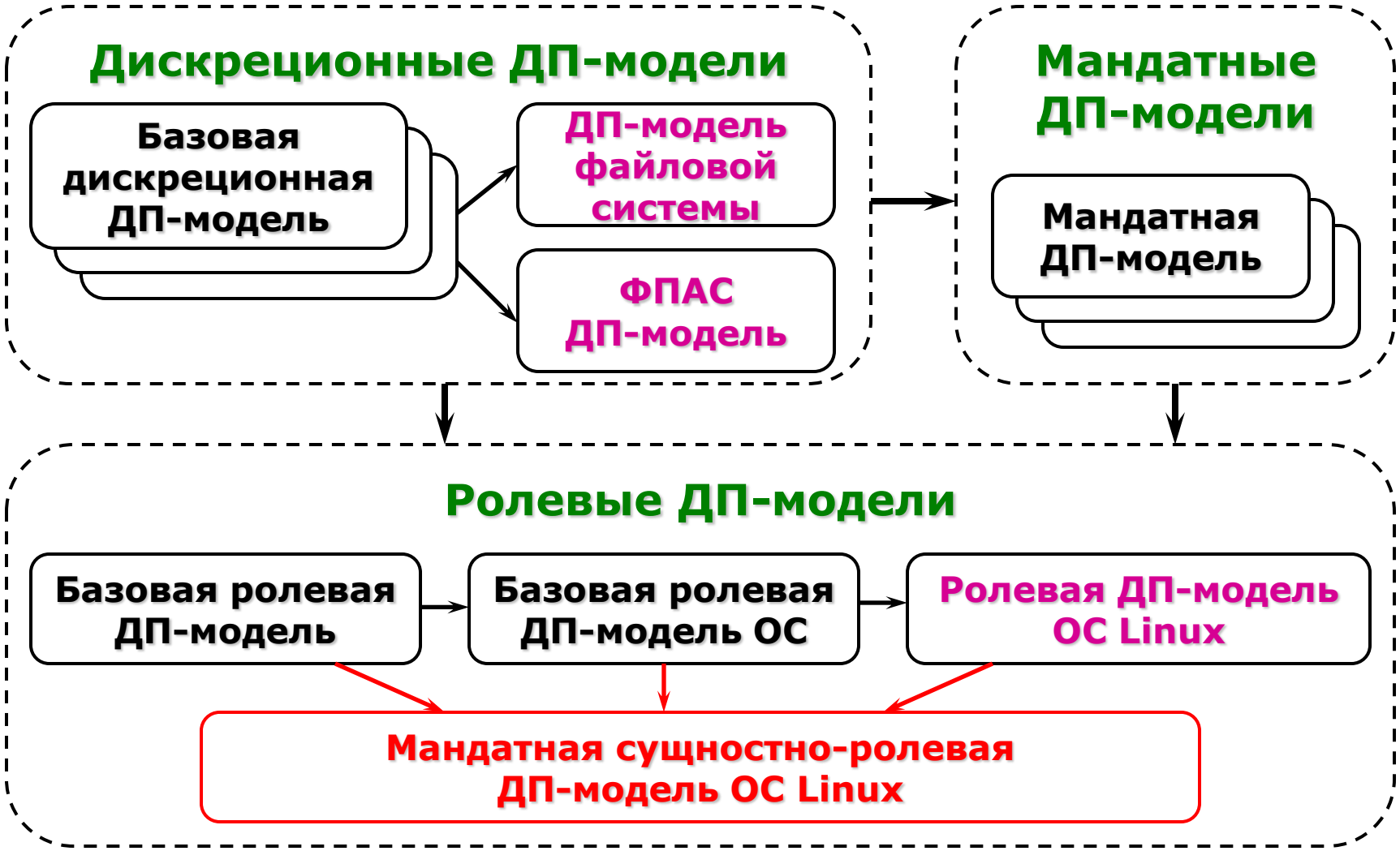
Адаптация мандатной сущностно-ролевой ДП-модели к условиям функционирования ОС семейства *Linux*

конференция
РусКрипто'2013

http://www.ruscrypto.ru/conference/

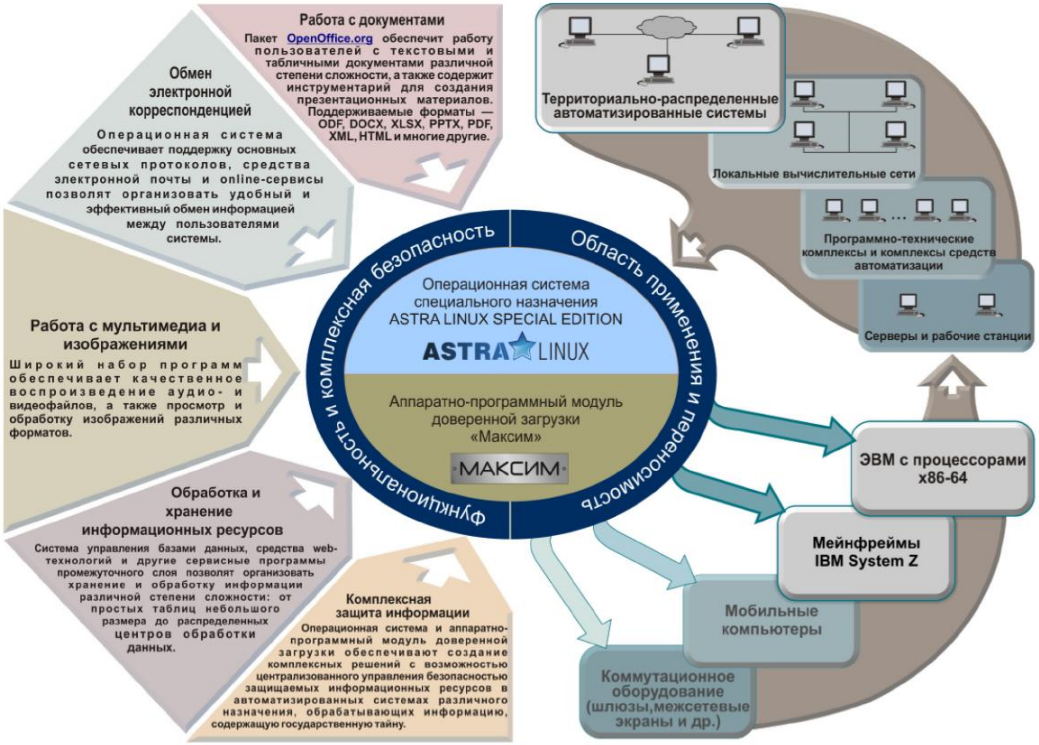
д.т.н., доцент Девянин П.Н.
УМО ИБ, г. Москва
peter_devyanin@hotmail.com

Основа — семейство ДП-моделей



Внедрение — реальная защищенная ОС

Внедрение модели — механизм управления доступом в защищенной ОС *Astra Linux Special Edition*



Основа модели — приоритетное выполнение требований мандатного управления доступом при реализации ролевого управления доступом в сочетании с мандатным контролем целостности

1. Роль — аналог сущности-контейнера

Модели семейства *RBAC*:

Роль — самостоятельная структура;

Иерархия ролей — самостоятельная структура;

Авторизованные роли пользователя (*UA*);

Текущие роли сессий (*roles*);

Администрирование прав доступа ролей — отдельный механизм (реализация функций *can-assign*, *can-revoke* и т.д.);

Администрирование иерархии ролей (реализация функции *can-modify*);

Информационные потоки по времени — не анализируются.

МРОСЛ ДП-модель:

Роль — аналог сущности-контейнера;

Иерархия ролей — аналог иерархии сущностей (виртуальная файловая система);

Права доступа ($read_r$, $write_r$, $execute_r$, own_r) индивидуальных административных ролей учетной записи пользователя к ролям;

Административные доступы субъект-сессий на чтение $read_a$ к ролям;

Административные доступы субъект-сессий на запись $write_a$ к ролям;

Изменение иерархии ролей аналогично изменению иерархии сущностей;

Информационные потоки по времени анализируются (в том числе, между сущностями и ролями).

2. Разделение на де-юре и де-факто

de_facto_own: $S \to S$ — функция фактического владения субъект-сессий субъект-сессиями.

de_facto_accesses: $S \to 2^{(E \cup R \cup AR) \times Ra}$ — функция де-факто доступов субъект-сессий, при этом по определению в каждом состоянии системы $G = (PA, user, A, AA, F, H_E)$ для каждой субъект-сессии $s \in S$ верно равенство:

de_facto_accesses$(s) = \{(e, \alpha_a)$: существует $s' \in$ *de_facto_own*(s) и [либо $e \in E$ и $(s', e, \alpha_a) \in A$], [либо $e \in R \cup AR$ и $(s', e, \alpha_a) \in AA$]$\}$.

Де-юре правила

Правило	Исходное состояние $G = (PA, user, A, AA, F, H_E)$	Результирующее состояние $G' = (PA', user', A', AA', F', H_E')$
access_write(x, x', y)	$x, x' \in S$, $y \in (E \setminus S) \cup R \cup AR$, существует $r \in R \cup AR$: $(x, r, read_a) \in AA$, [если $y \in E$, то $(y, write_a) \in APA(r)$], [если $y \in E \setminus S$, то $i_s(y) \le i_s(x)$ и либо (*execute_container*$(x, y) = true$ и, если $y \in E\_HOLE$, то $f_s(x) \le f_s(y)$, иначе $f_s(y) = f_s(x)$), либо $(x,$ *downgrade_admin_role*, $read_a) \in AA$], [если $y \in R \cup AR$, то $i_s(y) \le i_s(x)$, $Constraint_{ca}(AA') = true$, (для $e \in [y'$, [либо $(x, e, read_a) \in A$, либо $(x, e, write_a) \in A$), (либо $f_s(y) = f_s(x)$, либо $(x,$ *downgrade_admin_role*, $read_a) \in AA$)], [если $(y \in E \setminus S$ и $i_s(y) = i\_high)$ или $(y \in R \cup AR$ и $i_s(y) = i\_high)$, то $(x', f_s(x)\_i\_entity, write_a) \in A$]	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $H_E' = H_E$, $F' = F$, если $y \in E \setminus S$, то $A' = A \cup \{(x, y, write_a)\}$, $AA' = AA$, если $y \in R \cup AR$, то $AA' = AA \cup \{(x, y, write_a)\}$, $A' = A$
create_container$(x, x', y, yc, yi,$ *name*, $z)$	$x, x' \in S$, $y \notin E$, $z \in C \setminus S$, $(x, z, write_a) \in A$, *name* $\in$ *NAME* $\setminus \{$"", [либо $(yc = f_s(z) = f_s(x)$ и, если $CCR(z) =$ *false* или $CCR(x) =$ *false*, то $(x,$ *ccr_admin_role*, $read_a) \in AA)$], либо $(yc \le f_s(x)$ и $(x,$ *downgrade_admin_role*, $read_a) \in AA)$], $yi = i_s(x)$, $yi \le i_s(z)$, $Constraint_{ca}(PA') = true$, [если $i_s(z) = i\_high$, то $(x', f_s(x)\_i\_entity, write_a) \in A$]	$S' = S$, $E' = E \cup \{y\}$ ($C' = C \cup \{y\}$, $O' = O$), при этом $y \notin UE \cup RE$, $AA' = AA$, $user' = user$, $F' = F$, $A' = A \cup \{(x, y,$ *own*$_a)\}$, $f_s'(y) = yc$, $CCR'(y) = true$, $i_s'(y) = yi$, $CCR'(y) = true$, *entity_name*$(z, y) =$ *name*, *shared_container*$'(y) =$ *false*, $PA'(user(x)\_c\_f_s(x)\_i_s(x)) = PA(user(x)\_c\_f_s(x)\_i_s(x)) \cup \{(y,$ *own*$_a)\}$, и для $r' \in R \setminus (user(x)\_c\_f_s(x)\_i_s(x))$ выполняется равенство $PA'(r') = PA(r')$, $H_E'(z) = H_E(z) \cup \{y\}$, $H_E'(y) = \varnothing$, для $e \in E \setminus \{z\}$ выполняется равенство $H_E'(e) = H_E(e)$
read_container(x, y, z)	$x \in S$, $y \in (C \setminus S) \cup R \cup AR$, $z \in O$, $(x, z,$ *write*$_a) \in A$, [если $y \in C \setminus S$, то $(x, y, read_a) \in A$, иначе $(x, y, read_a) \in AA$]	$S' = S$, $E' = E$, $PA' = PA$, $A' = A$, $AA' = AA$, $user' = user$, $H_E' = H_E$, $F' = F$, если $y \in C \setminus S$, то $V'(z) =$ {*entity_name*(y, e): $e \in H_E(y)$ и выполняется условие $(f_s(e) \le f_s(x)$ или $CCR(e) =$ *false*$)$ и $(i_s(e) \le i_s(x)$ или $CCR(e) =$ *false*$)$}, если $y \in R \cup AR$, то $V'(z) = (role\_name(y, r)$: $r \in H_A(y))$
get_entity_attr(x, y, z)	$x \in S$, $y \in E$, $z \in O$, $(x, y,$ *own*$_a) \in A$, $(x, z,$ *write*$_a) \in A$	$S' = S$, $E' = E$, $PA' = PA$, $A' = A$, $AA' = AA$, $user' = user$, $H_E' = H_E$, $F' = F$, если $y \in S$, то $V'(z) = (user(s), f_s(y), i_s(y))$; если $y \in O \setminus S$, [если $(x,$ *hard_link_admin_role*, $read_a) \in AA$, то $V'(z) = (f_s(y),$ *CCR*$(y),$ $i_s(y),$ *CCR*$(y),$ $\{[e \in C$: $y \in H_E(e)]\})$, иначе $V'(z) = (f_s(y),$ *CCR*$(y),$ $i_s(y),$ *CCR*$(y)\})$], если $y \in C \setminus S$, то иначе $V'(z) = (f_s(y),$ *CCR*$(y),$ $i_s(y),$ *CCR*$(y)\})$, *shared_container*$(y))$
create_session(x, x', y, z)	$x, x' \in S$, $y \in E$, $z \notin E$, существует $r \in R \cup AR$ такая, что $(x, r,$ *read*$_a) \in AA$ и $(y,$ *execute*$_a) \in PA(r)$, *execute_container*$(x, y) = true$, $f_s(y) \le f_s(x)$	$S' = S \cup \{z\}$, $E' = E \cup \{z\}$, $A' = A$, $F' = F$, $f_s'(z) = f_s(x)$, $i_s'(z) = i_s(x)$, $user'(z) = user(x)$, для $s \in S$ выполняется $user'(s) = user(s)$, $[z] = fa(user(x), y)$, $[z] = fp(user(x), y)$; $AA' = AA \cup \{(z,$ *user*$(x)\_$*admin*$\_i_s(x),$ *read*$_a)$, $(z,$ *user*$(x)\_c\_f_s(x)\_i_s(x),$ *write*$_a)\} \cup \{(z,$ *user*$(x)\_c\_f_s$ $\setminus s\_i_s(x),$ *read*$_a)$, где $f \le f_s(x)$ и $f \le i_s(x)\}$, $PA'(user(x)\_c\_f_s(x)\_i_s(x)) = PA(user(x)\_c\_f_s(x)\_i_s(x)) \cup \{(z,$ *own*$_a)\}$, и для $r' \in R \setminus (user(x)\_c\_f_s(x)\_i_s(x))$ выполняется $PA'(r') = PA(r')$, $H_E'(x) = H_E(x) \cup \{z\}$, $H_E'(z) = \varnothing$, для $e \in E \setminus \{x\}$ выполняется равенство $H_E'(e) = H_E(e)$

Де-факто правила

Правило	Исходное состояние $G = (PA, user, A, AA, F, H_E)$	Результирующее состояние $G' = (PA', user', A', AA', F', H_E')$
de_facto_op$(s,$ *op*$(x, x',$ …$))$	$s \in N_S \cap S$, $x, x' \in$ *de_facto_own*(s), выполняются условия применения де-юре правила преобразования состояний *op*$(x, x',$ …$)$	Соответствуют результатам применения правила *op*$(x,$ $x',$ …$)$, кроме информационных потоков F
control(x, y, z)	$x \in N_S \cap S$, $y \in S$, $x \ne y$, $z \in [y]$ и или $x = z$, или $(x, z,$ *write*$_a) \in F$, или $z \in S$ и $z \in$ *de_facto_own*(x)	$S' = S$, $E' = E$, $PA' = PA$, $A' = A$, $AA' = AA$, $user' = user$, $H_E' = H_E$, *de_facto_own*$'(x) =$ *de_facto_own*$(x) \cup \{y\}$, $F' = F \cup \{(x, y,$ *write*$_a), (y, x,$ *write*$_a)\}$
know(x, y)	$x \in N_S \cap S$, $y \in S$, $x \ne y$, и для каждой $e \in [y[$, существует $(e, x,$ *write*$_a) \in F$	$S' = S$, $E' = E$, $PA' = PA$, $A' = A$, $AA' = AA$, $user' = user$, $H_E' = H_E$, *de_facto_own*$'(x) =$ *de_facto_own*$(x) \cup \{y\}$, $F' = F \cup \{(x, y,$ *write*$_a), (y, x,$ *write*$_a)\}$
take_access_own(x, y, z)	$x \in N_S \cap S$, $y, z \in S$, $y \in$ *de_facto_own*(x), $z \in$ *de_facto_own*(y)	$S' = S$, $E' = E$, $PA' = PA$, $A' = A$, $AA' = AA$, $user' = user$, $H_E' = H_E$, *de_facto_own*$'(x) =$ *de_facto_own*$(x) \cup \{z\}$, $F' = F \cup \{(x, z,$ *write*$_a), (z, x,$ *write*$_a)\}$
flow_memory_access(x, y, α_a)	$x \in S$, $y \in E \setminus E\_HOLE$, $(y, \alpha_a) \in$ *de_facto_accesses*(x), где $\alpha_a \in \{$*read*$_a,$ *write*$_a\}$	$S' = S$, $E' = E$, $PA' = PA$, $A' = A$, $AA' = AA$, $user' = user$, $H_E' = H_E$, если $\alpha_a =$ *read*$_a$, то $F' = F \cup \{(y, x,$ *write*$_m)\}$, если $\alpha_a =$ *write*$_a$, то $F' = F \cup \{(y, x,$ *write*$_m)\}$
flow_time_access(x, y)	$x \in (N_S \cup NF_S) \cap S$, или $[y \in E \cup R \cup AR$, $(y, \alpha_a) \in$ *de_facto_accesses*(x), где $\alpha_a \in R_a\}$, или $[y \in S$ и $y \in$ *de_facto_own*$(x)]$	$S' = S$, $E' = E$, $PA' = PA$, $A' = A$, $AA' = AA$, $user' = user$, $H_E' = H_E$, если $\alpha_a =$ *own*$_a$, то $F' = F \cup \{(y, x,$ *write*$_a)\} \cup \{(x, e,$ *write*$_a)$: $e \in E \cup R \cup AR, x \ne e$ и или $y \le e$, или $e \le y\}$; если $\alpha_a =$ *write*$_a$, или $y \in S$, то $F' = F \cup \{(y, x,$ *write*$_a)\} \cup \{(x, e,$ *write*$_a)$: $e \in E \cup R \cup AR, x \ne e$ и или $y \le e$, или $e \le y\}$; если $\alpha_a =$ *read*$_a$, то $F' = F \cup \{(y, x,$ *write*$_a)\}$
take_flow(x, y)	$x \in N_S \cap S$, $y \in S$, $x \ne y$, $y \in$ *de_facto_own*(x)	$S' = S$, $E' = E$, $PA' = PA$, $A' = A$, $AA' = AA$, $user' = user$, $H_E' = H_E$, $F' = F \cup \{(x, e, \alpha)$: $(y, e, \alpha) \in F$, $e \in E \cup R \cup AR$ и $\alpha \in \{$*write*$_m,$ *write*$_a\}\}$
find(x, y, z)	$x, y \in S$, $z \in E \cup R \cup AR$, $x \ne z$, $(x, y, \alpha_1), (y, z, \beta_1) \in F$, где $\alpha_1, \beta_1 \in \{$*write*$_m,$ *write*$_a\}$	$S' = S$, $E' = E$, $PA' = PA$, $A' = A$, $AA' = AA$, $user' = user$, $H_E' = H_E$, если *write*$_1 \in \{\alpha_1, \beta_1\}$ и $z \in E \setminus E\_HOLE$, то $F' = F \cup \{(x, z,$ *write*$_m)\}$, иначе если $x, y \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, z,$ *write*$_a)\}$, иначе $F' = F$
post(x, y, z)	$x, z \in S$, $y \in E \cup R \cup AR$, $x \ne z$, $(x, y, \alpha_1) \in F$, где $\alpha_1 \in \{$*write*$_m,$ *write*$_a\}$, $[(y, \beta_1) \in$ *de_facto_accesses*(z), где $\beta_1 \in R_a$, или $y \in S$ и $y \in$ *de_facto_own*$(z)]$	$S' = S$, $E' = E$, $PA' = PA$, $A' = A$, $AA' = AA$, $user' = user$, $H_E' = H_E$, если $\alpha_1 =$ *write*$_m$, $\beta_1 =$ *read*$_a$ и $y \in E \setminus E\_HOLE$, то $F' = F \cup \{(x, z,$ *write*$_m)\}$, иначе если $x, z \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, z,$ *write*$_a)\}$, иначе $F' = F$
pass(x, y, z)	$y \in S, x, z \in E \cup R \cup AR, x \ne z$, $[(x, \alpha_a) \in$ *de_facto_accesses*(y), где $\alpha_a \in \{$*read*$_a,$ *own*$_a\}$, или $x \in S$ и $x \in$ *de_facto_own*$(y)]$, $(y, z, \beta_1) \in F$, где $\beta_1 \in \{$*write*$_m,$ *write*$_a\}$	$S' = S$, $E' = E$, $PA' = PA$, $A' = A$, $AA' = AA$, $user' = user$, $H_E' = H_E$, если $\alpha_a =$ *read*$_a$, $\beta_1 =$ *write*$_m$ и $x, z \in E \setminus E\_HOLE$, то $F' = F \cup \{(x, z,$ *write*$_m)\}$, иначе если $x, z \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, z,$ *write*$_a)\}$, иначе $F' = F$

3. Учет специфики ОС семейства *Linux*

- Виды прав доступа и доступов:

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

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

- «Жесткие» ссылки (*hard link*):

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

- Разделяемые контейнеры:

shared_container: $C \to \{true, false\}$ — функция разделяемых контейнеров;

- Сущности-«дырки» («*dev/null*» или «*dev/zero*»):

E_HOLE — множество сущностей-объектов, не являющихся субъект-сессиями, в которых записываемые данные «не сохраняются»;

- Механизмы защиты от использования буфера обмена *clipboard* для создания запрещенных информационных потоков по времени.

4. Атрибуты *CCR* и *CCRI*

CCR: $(E \setminus S) \cup R \cup AR \rightarrow \{true, false\}$ — функция, задающая способ доступа к сущностям, не являющимся субъект-сессиями, внутри контейнеров или ролям в иерархии ролей (с учетом их мандатных уровней конфиденциальности).

CCRI: $(E \setminus S) \cup R \cup AR \rightarrow \{true, false\}$ — функция, задающая способ доступа к сущностям, не являющимся субъект-сессиями, внутри контейнеров, ролям или административным ролям в иерархии ролей (с учетом их мандатных уровней целостности).

Примеры требований:

- Любой доступ субъект-сессии к сущности, существующей в данном состоянии, должен осуществляться при выполнении условий доступа внутрь сущностей-контейнеров с учетом мандатных атрибутов конфиденциальности *CCR* и мандатных атрибутов целостности *CCRI*.
- Сущности-контейнеры, не являющиеся субъект-сессиями, создаются с мандатным атрибутом конфиденциальности *CCR* равным *true* и помечаются как неразделяемые; для изменения мандатного атрибута конфиденциальности *CCR* или метки разделяемости сущности-контейнера субъект-сессии необходимо обладать к ней доступом владения, а также обладать доступом на чтение к административной роли *ccr_admin_role*, имеющей минимальный уровень конфиденциальности и высокий уровень целостности;
- Субъект-сессии при получении ею доступа внутрь сущности-контейнера, роли или административной роли предоставляются имена входящих в них сущностей, ролей или административных ролей, либо имеющих уровень конфиденциальности не выше текущего уровня доступа субъект-сессии, либо обладающих мандатным атрибутом конфиденциальности *CCR* равным *false*.

5. Мандатный контроль целостности

Основные требования:

- Учетные записи доверенных пользователей имеют высокий уровень целостности, а недоверенных пользователей — низкий; уровень целостности сущности, входящей в состав сущности-контейнера и не являющейся субъект-сессией, не превосходит уровня целостности сущности-контейнера; уровни целостности сущностей, параметрически ассоциированных с учетной записью пользователя, ролью или административной ролью, совпадают с их уровнями целостности, соответственно.
- Любой доступ субъект-сессии к сущности, существующей в данном состоянии, должен осуществляться при выполнении условий доступа внутрь сущностей-контейнеров с учетом мандатных атрибутов целостности *CCRI*; субъект-сессии может быть предоставлен доступ на запись или владение к сущности только в случае, когда ее уровень целостности не выше текущего уровня целостности субъект-сессии; субъект-сессии может быть предоставлен доступ к роли или административной роли только в случае, когда ее уровень целостности не превосходит текущего уровня целостности субъект-сессии.
- Во множестве сущностей для каждого уровня конфиденциальности имеются сущности, обладающие высоким уровнем целостности. Получение доступа на запись к такой сущности субъект-сессией (или кооперирующей с ней другой субъект-сессией) является необходимым, когда она осуществляет действия, влияющие на целостность системы .
- Роль или административная роль может содержать права доступа на владение или запись к сущности, роли или административной роли с уровнем целостности не выше, чем у первой роли или административной роли.
- При создании каждой субъект-сессии она получает доступ на чтение к индивидуальной административной роли ее учетной записи пользователя с уровнем целостности не выше уровня целостности этой учетной записи пользователя; индивидуальные роли учетных записей пользователей имеют уровень целостности, не превосходящий уровня целостности этой учетной записи; на множестве индивидуальных ролей учетной записи пользователя задается иерархия:
- Изменение мандатного атрибута целостности *CCRI* роли или административные роли со значения *false* запрещено; сущности-контейнеры, не являющиеся субъект-сессиями, создаются с мандатным атрибутом целостности *CCRI* равным *true*; для изменения мандатного атрибута целостности *CCRI* сущности-контейнера субъект-сессии необходимо обладать к ней доступом владения, а также обладать доступом на чтение к административной роли *ccr_admin_role*; для создания, переименования или удаления сущности-контейнера с мандатным атрибутом целостности *CCRI* равным *false* требуется наличие у субъект-сессии доступа на чтение к административной роли *ccr_admin_role*.
- Субъект-сессии при получении ею доступа внутрь сущности-контейнера, роли или административной роли предоставляются имена входящих в них сущностей, ролей или административных ролей, имеющих уровень целостности не выше текущего уровня целостности субъект-сессии, либо обладающих мандатным атрибутом целостности *CCRI* равным *false*.

6. Индивидуальные роли и административные роли

Для каждой учетной записи пользователя $u \in U$ в зависимости от ее уровня целостности задаются индивидуальные административные роли, на которые авторизована только эта учетная запись:

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

Для каждой учетной записи пользователя $u \in U$ и уровня конфиденциальности $l \le f_u(u)$ и целостности $li \le i_u(u)$ задается индивидуальная роль, имеющая уровень конфиденциальности не выше уровня доступа учетной записи пользователя, административным правом доступа на чтение, запись и выполнение к которым обладают только его индивидуальные административные роли; каждая такая индивидуальная роль может содержать права доступа к сущностям с уровнем конфиденциальности равным уровню конфиденциальности роли:

- $u\_c\_l\_li \in R$ такая, что $f_r(u\_c\_l\_li) = l$, $(u\_c\_l\_low, \alpha_r) \in APA(u\_i\_low)$ и в случае, когда $i_u(u) = i\_high$, тогда $(u\_c\_l\_li, \alpha_r) \in APA(u\_i\_high)$, где $\alpha_r \in \{read_r, write_r, execute_r\}$;
- если для сущности $e \in E \setminus S$ выполняется $(e, \alpha_r) \in PA(u\_c\_l\_li)$, где $\alpha_r \in R_r$, то $f_e(e) = l$;
- если для субъект-сессии $s \in S$ выполняется $(s, own_r) \in PA(u\_c\_l\_li)$, то $f_s(s) = l$.

7. Администрирование системы

Правила преобразования состояний, реализующие администрирование:

- учетных записей пользователей (создание, удаление);
- ролей или административных ролей (создание, удаление, переименование);
- параметров мандатного управления доступом или мандатного контроля целостности (изменение уровней конфиденциальности, уровней доступа или уровней целостности);

Правило	Исходное состояние $G = (PA, user, A, AA, F, H_E)$	Результирующее состояние $G' = (PA', user', A', AA', F', H_E')$
create_user(x, x', u, uc, ui, ue)	$x, x' \in S$, $u \notin U$, $(x, create\_user\_admin\_role, read_a) \in AA$, $(x, roles\_admin\_role, \alpha_a) \in AA$, $(x, admin\_roles\_admin\_role, \alpha_a) \in AA$, где $\alpha_a \in \{read_a, write_a\}$, $ue \subset UE$, $[uc \le f_a(x), ui \le i_a(x)]$, [для $e \in ue$ выполняется $f_e(e) = uc$, $i_e(e) = ui$], $Constraint_{APA}(APA') = true$, [если $ui = i\_high$, то $(x', f_a(x)\_i\_entity, write_a) \in A$]	$S' = S, E' = E, A' = A, AA' = AA, F' = F, H_E' = H_E, U' = U \cup \{u\}$, $]u[= ue$, $f_u'(u) = uc$, $i_u'(u) = ui$, для $u' \in U$ выполняется $f_u'(u') = f_u(u')$, $i_u'(u') = i_u(u')$, $AR' = AR \cup \{u\_admin\_li: li \le ui\}$, $f_r'(u\_admin\_li) = \otimes LC$, $i_r'(u\_admin\_li) = li$, $shared\_container'(u\_admin\_li) = true$, $CCR'(u\_admin\_li) = CCRl'(u\_admin\_li) = false$, $role\_name'(u\_admin\_li) =$ "u_admin_li", где $li \le ui$, $H_R'(u\_admin\_i\_low) = \varnothing$, если $ui = i\_high$, то $H_R'(u\_admin\_i\_high) = \{u\_admin\_i\_low\}$, $R' = R \cup \{u\_c\_l\_li: l \le uc, li \le ui\}$, $f_r'(u\_c\_l\_li) = l$, $i_r'(u\_c\_l\_li) = li$, $shared\_container'(u\_c\_l\_li) = true$, $CCR'(u\_c\_l\_li) = CCRl'(u\_c\_l\_li) = false$, $role\_name'(u\_c\_l\_li) =$ "$u\_c\_l\_li$", где $l \le uc$, $li \le ui$, $APA'(admin\_roles\_admin\_role) = APA(admin\_roles\_admin\_role) \cup \{(u\_admin\_li, own_r): li \le ui\}$, $APA'(roles\_admin\_role) = APA(roles\_admin\_role) \cup \{(u\_c\_l\_li, own_r): l \le uc, li \le ui\}$, для $ar \in AR$ выполняется $APA'(ar) = APA(ar) \cup \{(u\_admin\_li, execute_r): li \le ui\} \cup \{(u\_c\_l\_li, execute_r): l \le uc, li \le ui\}$, $APA'(u\_admin\_i\_low) = \{(u\_admin\_i\_low, \alpha_r): \alpha_r \in \{read_r, write_r, execute_r\}\} \cup \{(u\_c\_l\_(i\_low), \alpha_r): l \le uc, \alpha_r \in \{read_r, write_r, execute_r\}\}$, если $ui = i\_high$, то $APA'(u\_admin\_i\_high) = \{(u\_admin\_i\_high, \alpha_r): \alpha_r \in \{read_r, write_r, execute_r\}\} \cup \{(u\_c\_l\_(i\_high), \alpha_r): l \le uc, \alpha_r \in \{read_r, write_r, execute_r\}\}$, $H_R'(u\_c\_l\_li) = \{u\_c\_l'\_li': l' < l, li' < li$ и не существуют $l'' \in LC: l' < l'' < l$ или $li'' \in LI: li' < li'' < li\}$, $PA'(u\_c\_l\_li) = \varnothing$, где $l \le uc, li \le ui$

Конкретизация правил, использующих права доступа владения own_r или доступ владения own_a.

Спасибо за внимание!