УДК 004.94

РЕЗУЛЬТАТЫ АНАЛИЗА УСЛОВИЙ РЕАЛИЗАЦИИ ИНФОРМАЦИОННЫХ ПОТОКОВ ПО ПАМЯТИ В РАМКАХ РОСЛ ДП-МОДЕЛИ

П. Н. Девянин

В докладе в рамках ролевой ДП-модели управления доступом и информационными потоками в операционных системах семейства Linux [1] (РОСЛ ДП-модели) рассматриваются достаточные условия реализации информационного потока по памяти. Дадим определения.

Определение 1. Назовем траекторию функционирования системы $\Sigma(G^*, OP)$ траекторией без информационных потоков по времени и кооперации доверенных и недоверенных субъект-сессий для передачи прав доступа, если при ее реализации используются только монотонные правила преобразования состояний и доверенные субъект-сессии не берут роли во множество де-факто текущих ролей; не дают другим ролям права доступа к сущностям; не изменяют атрибутов сущностей-контейнеров и не переименовывают сущности; не инициируют создание сущностей, субъект-сессий и жёстких ссылок на сущности, получение доступов к сущностям или де-факто владения к субъект-сессиями; не используют де-факто владение субъект-сессиями для выполнения от их имени правил преобразования состояний системы; не создают информационный поток по памяти к сущности i_entity в случае, когда правила преобразования состояний инициируются недоверенными субъект-сессиями; не реализуют информационные потоки по памяти к сущностям, функционально ассоциированным с доверенными субъектами, или от сущностей, параметрически ассоциированных с доверенными субъектами.

Определение 2. Пусть G_0 — состояние системы $\Sigma(G^*, OP)$, в котором существуют сущности $x, y \in E$, $x \neq y$. Определим предикат $can\_write\_memory(x, y, G_0)$, истинный тогда и только тогда, когда существуют состояния $G_1, \ldots, G_N$ и правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$, где $N \geqslant 0$, является траекторией без информационных потоков по времени и кооперации доверенных и недоверенных субъект-сессий для передачи прав доступа, и выполняется условие $(x, y, write_m) \in F_N$.

Определение 3. Пусть G — состояние системы $\Sigma(G^*, OP)$, в котором существуют сущности $x, y \in E$, $x \neq y$. Определим предикат $directly\_can\_write\_memory(x, y, G)$, истинный тогда и только тогда, когда существует последовательность сущностей $e_1, \ldots, e_m \in E$, где $e_1 = x$, $e_m = y$, $m \geqslant 2$, и либо $m = 2$ и $(x, y, write_m) \in F$, либо для каждого $i = 1, \ldots, m-1$ выполняется одно из условий 1–4:

1) $e_i \in L_S \cap S$ и либо $[(e_i, e_{i+1}, write_m) \in F]$, либо $[(e_i, e_{i+1}, write_a) \in A$ и не существует доверенной субъект-сессии $e'_{i+1} \in L_S \cap S$, такой, что $e_{i+1} \in [e'_{i+1}]$ и $(e_i, e_i) \in \in (e'_{i+1})_f(E)]$.

2) $e_i \in N_S \cap S$ и выполняется одно из условий:

— $(e_i, e_{i+1}, write_m) \in F$;
— существует $e'_i \in S$, такая, что $e'_i \in de\_facto\_own(e_i)$ и или $(e'_i, e_{i+1}, write_m) \in F$, или $e_{i+1} \in E \setminus S$ и $(e'_i, e_{i+1}, write_a) \in A$;
— $e_{i+1} \in E \setminus S$ и существует $e'_i \in S$, такая, что $e'_i \in de\_facto\_own(e_i)$, $execute\_container(e'_i, e_{i+1}) = \texttt{true}$, $i_e(e_{i+1}) \leqslant i_s(e'_i)$, [если $i_e(e_{i+1}) = i\_high$, то существует

$e''_i \in S$, такая, что $e''_i \in de\_facto\_own(e_i)$, $(e''_i, i\_entity, write_a) \in A$], и возможен один из двух случаев:

- $(e_{i+1}, write_r) \in PA(roles(e'_i))$;
- существует $r \in UA(user(e'_i))$, такая, что $(e_{i+1}, write_r) \in PA(r)$, [для $e \in]r[$ либо $(e'_i, e, read_a) \in A$, либо $(e'_i, e, write_a) \in A$], $i_r(r) \leqslant i_s(e'_i)$, [если $i_r(r) = i\_high$, то существует $(e''_i, i\_entity, write_a) \in A$], $Constraint_S(roles') = \texttt{true}$, где $[roles'(e'_i) = roles(e'_i) \cup \{r\}$ и для $e' \neq e'_i$ справедливо $roles'(e') = roles(e')]$;

— $e_{i+1} \in E \setminus S$ и существуют $e'_i \in S$, $r \in can\_manage\_rights(roles(e'_i) \cap AR) \cap \cap roles(e'_i)$, такие, что $e'_i \in de\_facto\_own(e_i)$, $i_e(e_{i+1}) \leqslant i_r(r) \leqslant i_s(e'_i)$, [если $i_e(e_{i+1}) = i\_high$, то существует $e''_i \in S$, такая, что $(e''_i, i\_entity, write_a) \in A$], $Constraint_P(PA') = \texttt{true}$, где $[PA'(r) = PA(r) \cup \{(e_{i+1}, write_r)\}$ и для $r' \neq r$ справедливо $PA'(r') = PA(r')]$ и либо $[(e'_i, e_{i+1}, own_a) \in A]$, либо $[(e_{i+1}, own_r) \in \in PA(roles(e'_i))$ и $execute\_container(e'_i, e_{i+1}) = \texttt{true}]$.

3) $e_{i+1} \in L_S \cap S$, $(e_{i+1}, e_i, read_a) \in A$, и либо не существует доверенной субъект-сессии $e'_{i+1} \in L_S \cap S$, такой, что $e_{i+1} \in [e'_{i+1}]$ и $(e_{i+1}, e_i) \in (e'_{i+1})_f(E)$, либо не существует доверенной субъект-сессии $e'_i \in L_S \cap S$, такой, что $e_i \in]e'_i[$ и $(e_{i+1}, e_{i+1}) \in (e'_i)_p(E)$.

4) $e_{i+1} \in N_S \cap S$, $e_i \in E \setminus S$ и существует $e'_{i+1} \in S$, такая, что $e'_{i+1} \in de\_facto\_own(e_{i+1})$ и выполняется одно из следующих условий:

— $(e'_{i+1}, e_i, read_a) \in A$;

— $execute\_container(e'_{i+1}, e_i) = \texttt{true}$ и возможен один из двух случаев:

- $(e_i, read_r) \in PA(roles(e'_{i+1}))$;
- существует $r \in UA(user(e'_{i+1}))$, такая, что $(e_i, read_r) \in PA(r)$, [для $e \in]r[$ либо $(e'_{i+1}, e, read_a) \in A$, либо $(e'_{i+1}, e, write_a) \in A$], $i_r(r) \leqslant i_s(e'_{i+1})$, [если $i_r(r) = i\_high$, то существует $(e''_{i+1}, i\_entity, write_a) \in A$], $Constraint_S(roles') = \texttt{true}$, где $[roles'(e'_{i+1}) = roles(e'_{i+1}) \cup \{r\}$ и для $e' \neq e'_{i+1}$ справедливо $roles'(e') = roles(e')]$;

— существует $r \in can\_manage\_rights(roles(e'_{i+1}) \cap AR) \cap roles(e'_{i+1})$, такая, что $e'_{i+1} \in de\_facto\_own(e_{i+1})$, $i_e(e_i) \leqslant i_r(r) \leqslant i_s(e'_{i+1})$, если $i_e(e_i) = i\_high$, то существует $e''_{i+1} \in S$, такая, что $(e''_{i+1}, i\_entity, write_a) \in A$, $Constraint_P(PA') = \texttt{true}$, где $[PA'(r) = PA(r) \cup \{(e_i, read_r)\}$ и для $r' \neq r$ справедливо $PA'(r') = PA(r')]$, и либо $[(e'_{i+1}, e_i, own_a) \in A]$, либо $[(e_i, own_r) \in PA(roles(e'_{i+1}))$ и $execute\_container(e'_{i+1}, e_i) = \texttt{true}]$.

Справедливо следующее утверждение о достаточных условиях истинности предиката $can\_write\_memory(x, y, G_0)$.

Утверждение 1. Пусть G_0 — состояние системы $\Sigma(G^*, OP)$, в котором существуют сущности $x, y \in E_0$, $x \neq y$. Если истинен предикат $directly\_can\_write\_memory(x, y, G_0)$, то истинен предикат $can\_write\_memory(x, y, G_0)$.

В дальнейшем планируется развитие РОСЛ ДП-модели по следующим направлениям: расширение достаточных условий реализации информационных потоков по памяти, анализ условий получения недоверенной субъект-сессией контроля над доверенной субъект-сессией, а также включение в модель элементов, позволяющих задать в ней мандатное управление доступом.

ЛИТЕРАТУРА

1. *Девянин П. Н.* Ролевая ДП-модель управления доступом и информационными потоками в операционных системах семейства Linux // Прикладная дискретная математика. 2012. № 1(15). С. 69–90.