2009 Математические основы компьютерной безопасности №1(3)
DOI 10.17223/20710410/3/8
УДК 681.322

АНАЛИЗ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ПОТОКОВ ПО ПАМЯТИ В КОМПЬЮТЕРНЫХ СИСТЕМАХ С ФУНКЦИОНАЛЬНО И ПАРАМЕТРИЧЕСКИ АССОЦИИРОВАННЫМИ СУЩНОСТЯМИ

Д. Н. Колегов

Томский государственный университет, г. Томск

E-mail: d.n.kolegov@gmail.com

В статье вводится определение сущностей, параметрически-ассоциированных с субъектами компьютерных систем. Строится расширение ДП-модели, охватывающее такие сущности.

Ключевые слова: *компьютерная безопасность, математические модели безопасности, дискреционные модели, анализ безопасности, права доступа, информационные потоки.*

Введение

Одной из современных моделей анализа безопасности компьютерных систем (КС) с дискреционным управлением доступа является ДП-модель с ее расширениями [1]. Дальнейшее изложение будет вестись на основе работы [1] с учетом всех определений, обозначений и теорем в ней. Сущность называется функционально-ассоциированной с субъектом, если она определяет вид преобразования данных, выполняемого этим субъектом. В ДП-моделях с функционально-ассоциированными с субъектами сущностями (ФАС ДП-моделях) анализируется ситуация, когда реализация информационного потока по памяти к сущности, функционально-ассоциированной с субъектом, приводит к изменению вида преобразования данных, реализуемого этим субъектом.

В то же время в современных КС возможна реализация информационного потока по памяти от сущности, позволяющая получить права доступа различных субъектов, в том числе и доверенных. Такие сущности являются параметрически-ассоциироваными с субъектами КС. Например, получение субъектом-нарушителем доступа на чтение к конфигурационному файлу или реестру, в котором хранится пароль или хэш-значение пароля субъекта КС, позволяет субъекту-нарушителю получить право доступа владения к последнему субъекту. Кроме того, в настоящее время дополнительно к классическим угрозам нарушения конфиденциальности, целостности и доступности информации рассматривают угрозу раскрытия параметров КС — возможность идентификации параметров, функций безопасности и свойств КС, знание которых позволяет реализовать нарушение безопасности [2]. Например, чтение сообщения, выдаваемого субъектом-процессом при подключении к нему, позволяет нарушителю идентифицировать программное обеспечение (ПО), реализующее данный субъект-процесс КС, и получить права доступа последнего, используя известные уязвимости в ПО.

Таким образом, для обеспечения возможности анализа получения субъектом права доступа владения к другому субъекту с использованием информационного потока от сущности, параметрически-ассоциированной с последним, следует построить расширение ФАС ДП-модели, охватывающее информационные потоки указанного вида и отражающее возможность получения субъектом права доступа владения к другому субъекту в современных КС. Решение этой задачи и является целью данной работы.

Для этого вводится определение параметрически-ассоциированных сущностей с субъектами в КС, на их основе строится ДП-модель с функционально- и параметрически-ассоциированными с субъектами сущностями, которая является требуемым расширением ФАС ДП-модели, и в рамках этой модели формулируются и обосновываются необходимые и достаточные условия получения недоверенным субъектом права доступа владения к другому субъекту без кооперации с ним.

1. ДП-модель с функционально- и параметрически-ассоциированными с субъектами сущностями

Определение 1. Пусть $G = (S, E, R \cup A \cup F, H)$ — состояние КС $\Sigma(G^*, OP)$. Сущность $e \in E$ будем называть *параметрически-ассоциированной* с субъектом $s \in S$ в состоянии G, если чтение данных в сущности e субъектом $z \in S$ позволяет ему получить право владения к субъекту s в этом или последующих состояниях КС.

Замечание 1. Сущность $e \in E$, параметрически-ассоциированная с субъектом $s \in S$, содержит аргументы операций преобразования данных, выполняемого субъектом s в этом или последующих состояниях КС.

Замечание 2. В множество сущностей, параметрически-ассоциированных с субъектом $s \in S$, могут входить сущности, на которые субъект s не имеет прав доступа. Например, сущность-пароль, параметрически-ассоциированная с недоверенным субъектом-пользователем в ОС семейства UNIX, хранится в файле /etc/shadow, правами доступа к которому могут обладать только доверенные субъекты-процессы данной ОС [3]. Кроме того, возможно создание субъектов ОС, которые препятствовали бы доступу субъектов к некоторым критичным сущностям КС, несмотря на наличие необходимых прав доступа данных субъектов к этим сущностям [4].

Замечание 3. Существуют сущности, параметрически-ассоциированные с субъектом, которые не являются функционально-ассоциированными с ним. Примером такой сущности является раздел реестра ОС семейства Windows, содержащий информацию об установленных обновлениях ОС узла. Также существуют сущности, параметрически-ассоциированные с субъектом, которые являются функционально-ассоциированными с ним. Так, зная идентификатор сессии пользователя web-приложения, возможно получить его права доступа; с другой стороны, удаление данного идентификатора приводит к закрытию сессии пользователя.

Наличие у субъекта данных о параметрах функционирования другого субъекта КС может позволить получить первому субъекту право доступа владения ко второму субъекту. При этом первому субъекту необходимо иметь возможность реализовать информационный поток по памяти к некоторому субъекту, позволяющему получить права доступа ко второму субъекту. Например, при получении субъектом-нарушителем пароля доверенного субъекта первому необходимо иметь возможность записи пароля в сущность-интерфейс КС. С учетом того, что доверенные субъекты не участвуют в реализации информационных потоков по времени, и того, что в современных КС, как правило, реализация информационного потока по времени от сущности, параметрически-ассоциированной с субъектом, не приводит к получению другим субъектом права доступа владения к первому, будем считать, что в КС выполняется следующее предположение.

Предположение 1. Информационный поток по времени от сущности, параметрически-ассоциированной с субъектом, не приводит к получению другим субъ-

ектом права доступа владения к первому субъекту. Если субъект реализовал информационный поток по памяти от сущности, параметрически-ассоциированной с другим субъектом, к себе, то первый субъект получает право доступа владения ко второму субъекту. Множество сущностей, параметрически-ассоциированных с субъектом, не изменяется в процессе функционирования КС.

Через $]s[\subset E$ обозначим множество всех сущностей, параметрически-ассоциированных с субъектом s. При этом будем считать, что $s \in]s[$.

В соответствии с данным предположением модифицируем определение ФАС ДП-модели для возможности анализа условий получения субъектом права доступа владения к другому субъекту с использованием реализации информационного потока по памяти от сущности, параметрически-ассоциированной с последним субъектом. Модифицированную ФАС ДП-модель будем называть *ДП-моделью с функционально- и параметрически-ассоциированными с субъектами сущностями*, или ФПАС ДП-моделью. Модификация состоит в добавлении к правилам преобразования состояний ФАС ДП-модели правила $know(x, y, z)$, определенного в таблице: аргументом операции является состояние G, значением — состояние G', параметрами — сущности x, y, z.

Условия и результаты применения правила *know(x, y, z)* ФПАС ДП-модели

Правило	Исходное состояние $G = (S, E, R \cup A \cup F, H)$	Результирующее состояние $G' = (S', E', R' \cup A' \cup F', H')$
$know(x, y, z)$	$x, y \in S, z \in E, z \in]y[$ и или $x = z$, или $(z, x, write_m) \in F$	$S' = S, E' = E, A' = A, H' = H, F' = F, R' = R \cup (x, y, ownr)$

Замечание 4. Правило $know(x, y, z)$ является монотонным, то есть применение данного правила не приводит к удалению ребер или вершин из графа доступа.

Замечание 5. Как и правило $control(x, y, z)$ ФАС ДП-модели, правило $know(x, y, z)$ отражает возможность одним субъектом получить право доступа владения к другому субъекту путем реализации информационного потока.

Рассмотрим условия получения недоверенным субъектом права доступа владения own_r к другому субъекту без кооперации с ним для случая, когда в КС $\Sigma(G^*, OP)$ используются правила преобразования состояний $know(x, y, z)$ и $control(x, y, z)$.

2. Анализ условий получения субъектом права доступа владения к другому субъекту

Определение 2. Траекторию функционирования КС $\Sigma(G^*, OP)$ будем называть *траекторией без кооперации доверенных и недоверенных субъектов для передачи прав доступа*, если при ее реализации используются монотонные правила преобразования состояний, и доверенные субъекты:

— не дают недоверенным субъектам права доступа к сущностям;
— не берут у недоверенных субъектов права доступа к сущностям;
— используя информационные потоки по памяти к сущностям, не получают право доступа владения к субъектам;
— используя информационные потоки по памяти от сущностей, не получают право доступа владения к субъектам.

Таким образом, в КС $\Sigma(G^*, OP)$ на траекториях без кооперации доверенных и недоверенных субъектов для передачи прав доступа доверенные субъекты:

— не инициируют выполнения следующих правил преобразования состояний: $take\_right(\alpha_r, u, x, e)$, $grant\_right(\alpha_r, u, x, e)$, $control(u, y, z)$, $know(u, y, z)$;
— доверенные субъекты могут выполнять монотонные правила преобразования состояний: $own\_take(\alpha_r, u, e)$, $create\_entity(u, e, e')$, $create\_subject(u, e, e')$, $rename\_entity(u, e, e')$, $access\_read(u, e)$, $access\_write(u, e)$, $access\_append(u, e)$, $find(u, e, e')$, $post(u, e, e')$, $pass(u, e, e')$ с условиями и результатами применения в соответствии с правилами преобразования БК ДП-модели,

где $u \in L_S$ — доверенный субъект, $x \in N_S$ — недоверенный субъект, y — субъект, что $z \in]y[$ или $z \in [y]$, z, e, e' — сущности, $\alpha_r \in R_r$ — право доступа.

Определение 3. Траекторию функционирования КС $\Sigma(G^*, OP)$ будем называть *траекторией без кооперации доверенных и недоверенных субъектов для передачи прав доступа и реализации информационных потоков*, если она является траекторией без кооперации доверенных и недоверенных субъектов для передачи прав доступа, и при ее реализации используются правила преобразования состояний:

— $take\_right(\alpha_r, x, y, z)$, $grant\_right(\alpha_r, x, y, z)$, $own\_take(\alpha_r, x, y)$ с условиями и результатами применения в соответствии с правилами преобразования базовой ДП-модели;
— $create\_entity(x, y, z)$, $create\_subject(x, y, z)$, $rename\_entity(x, y, z)$, $flow(x, y, y', z)$, $find(x, y, z)$, $post(x, y, z)$, $pass(x, y, z)$, $access\_read(x, y)$, $access\_write(x, y)$, $access\_append(x, y)$ с условиями и результатами применения в соответствии с правилами преобразования БК ДП-модели;
— $control(x, y, z)$, $know(x, y, z)$ с условиями и результатами применения в соответствии с правилами преобразования ФАС ДП-модели и таблицей.

Определение 4. Пусть имеются $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — состояние КС $\Sigma(G^*, OP)$, недоверенный субъект $x \in N_S \cap S_0$ и $y \in S_0$, где $x \neq y$. Определим предикат $can\_steal\_own(x, y, G_0, L_S)$, который будет истинным тогда и только тогда, когда существуют состояния $G_1, \ldots, G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний $op_1, \ldots, op_N$ такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N$ является траекторией без кооперации доверенных и недоверенных субъектов для передачи прав доступа и реализации информационных потоков и $(x, y, own_r) \in R_N$, где $N \geqslant 0$. При этом в последовательности правил $op_1, \ldots, op_N$ отсутствуют правила вида $grant\_right(\alpha_r, y, s, e)$, $take\_right(\alpha_r, y, s, e)$, $control(y, s, e')$, $know(y, s, e')$, где $\alpha_r \in R_r$, $s \in N_S \cap S_0$, $e, e' \in E_0$ и $e' \in [s]$ или $e' \in]s[$ (субъект y не передает другим субъектам любые имеющиеся у него права доступа к любым сущностям и не получает с использованием правил $control(x, y, z)$ и $know(x, y, z)$ право доступа владения own_r к другим субъектам).

Определение 5. Пусть имеются $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — состояние КС $\Sigma(G^*, OP)$, недоверенный субъект $x \in N_S \cap S_0$ и $y \in S_0$, где $x \neq y$. Определим предикат $directly\_can\_share\_own(x, y, G_0, L_S)$, который будет истинным тогда и только тогда, когда существует последовательность субъектов $s_1, \ldots, s_m \in S_0$, где $s_1 = x$, $s_m =$ y и $m \geqslant 2$, таких, что для каждого $i = 1, ..., m-1$ выполняется одно из условий:

1) $s_i \in N_S \cap S_0$, $s_i \in [s_{i+1}]$ или $s_i \in]s_{i+1}[$;
2) истинен предикат $can\_share(own_r, s_i, s_{i+1}, G_0, L_S)$;
3) существует сущность $e \in [s_{i+1}]$, что предикат $can\_write\_memory(s_i, e, G_0, L_S)$ является истинным;

4) существует сущность $e \in]s_{i+1}[$, что предикат $can\_write\_memory(e, s_i, G_0, L_S)$ является истинным.

Лемма 1. Пусть имеются $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — состояние КС $\Sigma(G^*, OP)$, недоверенный субъект $x \in N_S \cap S_0$ и $y \in S_0$, где $x \neq y$, и истинен предикат $directly\_can\_share\_own(x, y, G_0, L_S)$. Тогда истинен предикат $can\_share\_own(x, y, G_0, L_S)$.

Доказательство. Пусть истинен предикат $directly\_can\_share\_own(x, y, G_0, L_S)$, тогда по определению 5 существует последовательность субъектов $s_1, \ldots, s_m$ в S_0, где $s_1 = x$, $s_m = y$ и $m \geqslant 2$, таких, что для каждого $i = 1, ..., m-1$ выполняется одно из условий 1 – 4 определения 5. Докажем, что для такой последовательности $s_1, \ldots, s_m$ предикат $can\_share\_own(x, y, G_0, L_S)$ является истинным. Проведем доказательство этого утверждения индукцией по длине m.

Пусть $m = 2$, тогда возможно четыре случая.

Первый случай: $x \in N_S \cap S_0$, $x \in [y]$ или $x \in]y[$. Если $x \in [y]$, то пусть $op_1 = control(x, y, x)$. Тогда $G_0 \vdash_{op1} G_1$, $(x, y, own_r) \in R_1$ и предикат $can\_share\_own(x, y, G_0, L_S)$ истинен. Если $x \in]y[$, то пусть $op_1 = know(x, y, x)$. Тогда $G_0 \vdash_{op1} G_1$ и $(x, y, own_r) \in R_1$ и предикат $can\_share\_own(x, y, G_0, L_S)$ также истинен.

Во втором случае истинен предикат $can\_share(x, y, G_0, L_S)$. Следовательно, истинен предикат $can\_share\_own(x, y, G_0, L_S)$.

В третьем случае имеется $x \in N_S \cap S_0$, существует сущность $e \in [y]$ и истинен предикат $can\_write\_memory(x, e, G_0, L_S)$. Пусть $op_1 = control(x, y, e)$, тогда $G_0 \vdash_{op1} G_1$, $(x, y, own_r) \in R_1$ и предикат $can\_share\_own(x, y, G_0, L_S)$ истинен.

В четвертом случае имеется $x \in N_S \cap S_0$, существует сущность $e \in]y[$ и истинен предикат $can\_write\_memory(e, x, G_0, L_S)$. Пусть $op_1 = know(x, y, e)$, тогда $G_0 \vdash_{op1} G_1$, $(x, y, own_r) \in R_1$ и предикат $can\_share\_own(x, y, G_0, L_S)$ истинен.

Докажем индуктивный шаг. Пусть $m > 2$ и доказываемое утверждение верно для всех последовательностей субъектов длины $k < m$. Докажем, что оно верно и для всех таких последовательностей длины m.

Рассмотрим последовательность субъектов $s_1, \ldots, s_m$ в S_0, где $s_1 = x$, $s_2 = z$ и $s_m = y$. Пусть $z \in N_S \cap S_0$. Тогда по предположению индукции существуют правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N$, и верно, что (x, z, own_r), $(z, y, own_r) \in R_N$, где $N \geqslant 0$. Положим $op_{N+1} = take\_right(own_r, x, z, y)$. Тогда $G_N \vdash_{op(N+1)} G_{N+1} = (S_{N+1}, E_{N+1}, R_{N+1} \cup A_{N+1} \cup F_{N+1}, H_{N+1})$ и $(x, y, own_r) \in R_{N+1}$, следовательно, предикат $can\_share\_own(x, y, G_0, L_S)$ истинен.

Если $z \in L_S \cap S_0$, то $(z, y, own_r) \in R_0$, и по предположению индукции предикат $can\_share\_own(x, z, G_0, L_S)$ истинен. Следовательно, существуют правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N$, и $(x, z, own_r) \in R_N$, где $N \geqslant 0$. Аналогично получаем истинность предиката $can\_share\_own(x, y, G_0, L_S)$. Лемма доказана. ■

Определим и обоснуем алгоритмически проверяемые необходимые и достаточные условия истинности предиката $can\_steal\_own(x, y, G_0, L_S)$.

Теорема 1. Пусть имеется $G_0 = (S_0, E_0, R_0 \cup A_0 \cup F_0, H_0)$ — состояние КС $\Sigma(G^*, OP)$, и $(x, y, own_r) \in N_r$, где $x \in N_S \cap S_0$, $y \in S_0$ и $x \neq y$. Тогда предикат $can\_steal\_own(x, y, G_0, L_S)$ является истинным, если и только если существует по-

следовательность субъектов $s_1, \ldots, s_m$ в S_0, где $s_1 = x$, $s_m = y$ и $m \geqslant 2$, таких, что выполняется одно из условий.

Условие 1. $m = 2$ и истинен предикат $directly\_can\_share\_own(x, y, G_0, L_S)$.

Условие 2. $m > 2$ и для каждого $i = 1, ..., m-2$ выполняется одно из условий:

— $s_i, s_{i+1} \in N_S \cap S_0$, $s_i \neq y$ и предикаты $directly\_can\_share\_own(s_i, s_{i+1}, G_0, L_S)$, $directly\_can\_share\_own(s_{i+1}, s_{i+2}, G_0, L_S)$ являются истинными;

— $i < m-2$, $s_i, s_{i+2} \in N_S \cap S_0$, $s_i, s_{i+2} \neq y$ и являются истинными предикаты $directly\_can\_share\_own(s_i, s_{i+1}, G_0, L_S)$ и $directly\_can\_share\_own(s_{i+2}, s_{i+1}, G_0, L_S)$;

— $i < m-2$, $s_{i+1}, s_{i+2} \in N_S \cap S_0$, $s_{i+1} \neq y$, $s_{i+2} \neq y$ и являются истинными предикаты $directly\_can\_share\_own(s_{i+1}, s_i, G_0, L_S)$, $directly\_can\_share\_own(s_{i+2}, s_{i+1}, G_0, L_S)$;

— $s_{i+1} \in N_S \cap S_0$, $s_{i+1} \neq y$ и являются истинными предикаты $directly\_can\_share\_own(s_{i+1}, s_i, G_0, L_S)$, $directly\_can\_share\_own(s_{i+1}, s_{i+2}, G_0, L_S)$.

Доказательство. Докажем достаточность выполнения условий теоремы для истинности предиката $can\_steal\_own(x, y, G_0, L_S)$. Если выполнено первое условие теоремы, то в соответствии с леммой истинен предикат $can\_share\_own(x, y, G_0, L_S)$. Следовательно, предикат $can\_steal\_own(x, y, G_0, L_S)$ также является истинным. Если выполнено второе условие теоремы, то тогда существует последовательность субъектов $s_1, ..., s_m$ в S_0, где $s_1 = x$, $s_m = y$, $x \neq y$ и $m > 2$.

Проведем доказательство индукцией по длине m последовательности субъектов. Пусть $m = 3$. Возможны два случая. Первый случай: $x, s_2 \in N_S \cap S_0$ и истинны предикаты $directly\_can\_share\_own(x, s_2, G_0, L_S)$, $directly\_can\_share\_own(s_2, y, G_0, L_S)$. Тогда в соответствии с леммой существуют состояния $G_1, \ldots, G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N$ и (x, s_2, own_r), $(s_2, y, own_r) \in R_N$, где $N \geqslant 0$. Пусть $op_{N+1} = take\_right(own_r, x, s_2, y)$ и $G_N \vdash_{op(N+1)} G_{N+1}$, где $G_{N+1} = (S_{N+1}, E_{N+1}, R_{N+1} \cup A_{N+1} \cup F_{N+1}, H_{N+1})$. Тогда $(x, y, own_r) \in R_{N+1}$ и предикат $can\_steal\_own(x, y, G_0, L_S)$ является истинным. Второй случай: $s_2 \in N_S \cap S_0$ и истинны предикаты $directly\_can\_share\_own(s_2, x, G_0, L_S)$, $directly\_can\_share\_own(s_2, y, G_0, L_S)$. Тогда в соответствии с леммой существуют состояния $G_1, \ldots, G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N$ и (s_2, x, own_r), $(s_2, y, own_r) \in R_N$, где $N \geqslant 0$. Пусть $op_{N+1} = grant\_right(own_r, s_2, x, y)$ и $G_N \vdash_{op(N+1)} G_{N+1}$, где $G_{N+1} = (S_{N+1}, E_{N+1}, R_{N+1} \cup A_{N+1} \cup F_{N+1}, H_{N+1})$. Тогда $(x, y, own_r) \in R_{N+1}$ и предикат $can\_steal\_own(x, y, G_0, L_S)$ является истинным.

Докажем индуктивный шаг. Пусть $m > 4$ и утверждение теоремы верно для всех последовательностей субъектов длины $k < m$. Докажем, что утверждение теоремы верно для всех таких последовательностей длины m. Пусть имеется последовательность субъектов $s_1, \ldots, s_m$ в S_0, где $s_1 = x$, $s_2 = z$, $s_3 = w$ и $s_m = y$. Возможно четыре случая.

Первый случай: $x, z \in N_S \cap S_0$ и истинны предикаты $directly\_can\_share\_own(x, z, G_0, L_S)$, $directly\_can\_share\_own(z, w, G_0, L_S)$. Тогда в соответствии с леммой существуют состояния $G_1, \ldots, G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N$ и $(x, z, own_r), (z, w, own_r) \in R_N$, где $N \geqslant 0$. По предположению индукции существуют состояния $G_{N+1}, \ldots, G_{N+K} = (S_{N+K}, E_{N+K}, R_{N+K} \cup A_{N+K} \cup F_{N+K}, H_{N+K})$ и правила преобразования состояний $op_{N+1}, \ldots, op_{N+K}$, такие, что $G_{N+1} \vdash_{op(N+1)} G_{N+2} \vdash_{op(N+2)}$

$\ldots \vdash_{op(N+K)} G_{N+K}$ и $(w, y, own_r) \in R_{N+K}$, где $K \geqslant 0$. Положим $op_{N+K+1} = take\_right(own_r, x, z, w)$, $op_{N+K+2} = take\_right(own_r, x, w, y)$, тогда $G_{N+K} \vdash_{op(N+K+1)} G_{N+K+1} \vdash_{op(N+K+2)} G_{N+K+2} = (S_{N+K+2}, E_{N+K+2}, R_{N+K+2} \cup A_{N+K+2} \cup F_{N+K+2}, H_{N+K+2})$ и $(x, y, own_r) \in R_{N+K+2}$. Следовательно, предикат $can\_share\_own(x, y, G_0, L_S)$ истинен.

Второй случай: $x, w \in N_S \cap S_0$ и истинны предикаты $directly\_can\_share\_own(x, z, G_0, L_S)$, $directly\_can\_share\_own(w, z, G_0, L_S)$. Тогда в соответствии с леммой существуют состояния $G_1, \ldots, G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N$ и $(x, z, own_r), (w, z, own_r) \in R_N$, где $N \geqslant 0$. По предположению индукции существуют состояния $G_{N+1}, \ldots, G_{N+K} = (S_{N+K}, E_{N+K}, R_{N+K} \cup A_{N+K} \cup F_{N+K}, H_{N+K})$ и правила преобразования состояний $op_{N+1}, \ldots, op_{N+K}$, такие, что $G_{N+1} \vdash_{op(N+1)} G_{N+2} \vdash_{op(N+2)} \ldots \vdash_{op(N+K)} G_{N+K}$ и $(w, y, own_r) \in R_{N+K}$, где $K \geqslant 0$. Положим $op_{N+K+1} = grant\_right(own_r, w, z, y)$, $op_{N+K+2} = take\_right(own_r, x, z, y)$, тогда $G_{N+K} \vdash_{op(N+K+1)} G_{N+K+1} \vdash_{op(N+K+2)} G_{N+K+2} = (S_{N+K+2}, E_{N+K+2}, R_{N+K+2} \cup A_{N+K+2} \cup F_{N+K+2}, H_{N+K+2})$ и $(x, y, own_r) \in R_{N+K+2}$. Следовательно, предикат $can\_share\_own(x, y, G_0, L_S)$ истинен.

Третий случай: $w, z \in N_S \cap S_0$, $z \neq y$ и истинны предикаты $directly\_can\_share\_own(z, x, G_0, L_S)$, $directly\_can\_share\_own(w, z, G_0, L_S)$. Тогда в соответствии с леммой существуют состояния $G_1, \ldots, G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N$ и $(z, x, own_r), (w, z, own_r) \in R_N$, где $N \geqslant 0$. По предположению индукции существуют состояния $G_{N+1}, \ldots, G_{N+K} = (S_{N+K}, E_{N+K}, R_{N+K} \cup A_{N+K} \cup F_{N+K}, H_{N+K})$ и правила преобразования состояний $op_{N+1}, \ldots, op_{N+K}$, такие, что $G_{N+1} \vdash_{op(N+1)} G_{N+2} \vdash_{op(N+2)} \ldots \vdash_{op(N+K)} G_{N+K}$ и $(w, y, own_r) \in R_{N+K}$, где $K \geqslant 0$. Положим $op_{N+K+1} = grant\_right(own_r, w, z, y)$, $op_{N+K+2} = grant\_right(own_r, z, x, y)$, тогда $G_{N+K} \vdash_{op(N+K+1)} G_{N+K+1} \vdash_{op(N+K+2)} G_{N+K+2} = (S_{N+K+2}, E_{N+K+2}, R_{N+K+2} \cup A_{N+K+2} \cup F_{N+K+2}, H_{N+K+2})$ и $(x, y, own_r) \in R_{N+K+2}$. Следовательно, предикат $can\_share\_own(x, y, G_0, L_S)$ истинен.

Четвертый случай: $z \in N_S \cap S_0$, $z \neq y$ и истинны предикаты $directly\_can\_share\_own(z, x, G_0, L_S)$, $directly\_can\_share\_own(z, w, G_0, L_S)$. Тогда в соответствии с леммой существуют состояния $G_1, \ldots, G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N$ и $(z, x, own_r), (z, w, own_r) \in R_N$, где $N \geqslant 0$. По предположению индукции существуют состояния $G_{N+1}, \ldots, G_{N+K} = (S_{N+K}, E_{N+K}, R_{N+K} \cup A_{N+K} \cup F_{N+K}, H_{N+K})$ и правила преобразования состояний $op_{N+1}, \ldots, op_{N+K}$, такие, что $G_{N+1} \vdash_{op(N+1)} G_{N+2} \vdash_{op(N+2)} \ldots \vdash_{op(N+K)} G_{N+K}$ и $(w, y, own_r) \in R_{N+K}$, где $K \geqslant 0$. Положим $op_{N+K+1} = take\_right(own_r, z, w, y)$, $op_{N+K+2} = grant\_right(own_r, z, x, y)$, тогда $G_{N+K} \vdash_{op(N+K+1)} G_{N+K+1} \vdash_{op(N+K+2)} G_{N+K+2} = (S_{N+K+2}, E_{N+K+2}, R_{N+K+2} \cup A_{N+K+2} \cup F_{N+K+2}, H_{N+K+2})$ и $(x, y, own_r) \in R_{N+K+2}$. Следовательно, предикат $can\_share\_own(x, y, G_0, L_S)$ истинен.

Индуктивный шаг доказан. Доказательство достаточности выполнения условий теоремы для истинности предиката $can\_steal\_own(x, y, G_0, L_S)$ закончено.

Докажем необходимость выполнения условий теоремы для истинности предиката $can\_steal\_own(x, y, G_0, L_S)$.

Пусть истинен предикат $can\_steal\_own(x, y, G_0, L_S)$. Тогда по определению существуют состояния $G_1, \ldots, G_N = (S_N, E_N, R_N \cup A_N \cup F_N, H_N)$ и правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N$, и $(x, y, own_r) \in R_N$,

где $N \geqslant 0$. Выберем среди последовательностей правил преобразований ту, у которой длина N является минимальной. Следовательно, $(x, y, own_r) \notin R_{N-1}$. При этом в последовательности правил $op_1, \ldots, op_N$ отсутствуют правила вида $grant\_right(\alpha, y, s, e)$, $take\_right(\alpha, y, s, e)$, $control(y, s, e')$, $know(y, s, e')$, где $alpha \in R_r$, $s \in N_S \cap S_0$, $e, e' \in E_0$ и $e' \in [s]$ или $e' \in]s[$. Проведем доказательство индукцией по числу N.

Пусть $N = 0$, тогда $(x, y, own_r) \in R_0$ и условие 1 теоремы выполнено. Пусть $N = 1$, тогда $x \in N_S \cap S_0$, $y \in S0$, $(x, y, own_r) \notin R_0$ и существует правило преобразования состояний op_1, такое, что $G_0 \vdash_{op1} G_1$ и $(x, y, own_r) \in R_1$. Из определения правил преобразования состояний следует, что возможны шесть случаев:

— $x \in [y]$ и $op_1 = control(x, y, x)$;
— $x \in]y[$ и $op_1 = know(x, y, x)$;
— существует сущность $e \in [y]$, такая, что $(x, e, write_m) \in F_0$ и $op_1 = control(x, y, e)$;
— существует сущность $e \in]y[$, такая, что $(e, x, write_m) \in F_0$ и $op_1 = know(x, y, e)$;
— существует субъект $z \in N_S \cap S_0$, такой, что (x, z, own_r), $(x, z, own_r) \in R_0$ и $op_1 = take\_right(own_r, x, z, y)$;
— существует субъект $z \in N_S \cap S_0$, такой, что $(z, x, own_r), (x, z, own_r) \in R_0$ и $op_1 = grant\_right(own_r, z, x, y)$.

Все шесть случаев соответствуют условиям 1 и 2 теоремы.

Пусть $N > 2$ и утверждение теоремы верно для всех последовательностей преобразований состояний длины $l < N$. Тогда $x \in N_S \cap S_0$, $y \in S_0$, $(x, y, own_r) \notin R_{N-1}$ и существует правило преобразования состояний op_N, такое, что $G_{N-1} \vdash_{opN} G_N$ и $(x, y, own_r) \in R_N$.

Из определения правил преобразования состояний и минимальности N следует, что выполняется одно из условий:

— существует сущность $e \in [y]$, такая, что $(x, e, write_m) \in F_{N-1}$ и $op_N = control(x, y, e)$;
— существует сущность $e \in]y[$, такая, что $(e, x, write_m) \in F_{N-1}$ и $op_N = know(x, y, e)$;
— существует субъект $z \in N_S \cap S_0$, такой, что $(x, z, own_r), (z, y, own_r) \in R_{N-1}$ и $op_N = take\_right(own_r, x, z, y)$;
— существует субъект $z \in N_S \cap S_0$, такой, что $(z, x, own_r), (z, y, own_r) \in R_{N-1}$ и $op_N = grant\_right(own_r, z, x, y)$.

Если выполнено первое или второе условие, то выполнено первое условие теоремы. Рассмотрим случай выполнения третьего условия, когда $(x, z, own_r), (z, y, own_r) \in R_{N-1}$ и $op_N = take\_right(\alpha, x, z, y)$. Доказательство для случая выполнения четвертого условия проводится аналогично доказательству для случая выполнения третьего условия.

Так как длина N минимальна, то в последовательности преобразований состояний не использовались правила вида $create\_entity(x, y, z)$ и $create\_subject(x, y, z)$. Следовательно, $z \in S_0$ и истинны предикаты $can\_steal\_own(x, z, G_0, L_S)$ и $can\_steal\_own(z, y, G_0, L_S)$ с длинами последовательностей преобразований состояний меньше N. По предположению индукции возможны четыре случая. Первый случай: истинны предикаты $directly\_can\_share\_own(x, z, G_0, L_S)$ и $directly\_can\_share\_own(z, y, G_0, L_S)$. Следовательно, второе условие теоремы выполнено.

Второй случай: истинен предикат $directly\_can\_share\_own(x, z, G_0, L_S)$ и существует последовательность субъектов $s_1, \ldots, s_m$ в S_0, где $s_1 = z, s_m = y$ и $m \geqslant 2$, таких, что выполняется условие 2. Следовательно, существует последовательность субъектов $s_1, \ldots, s_m, s_{m+1}$ в S_0, где $s_1 = x$, $s_2 = z$, $s_{m+1} = y$, таких, что выполняется условие 2 теоремы.

Третий случай: истинен предикат $directly\_can\_share\_own(z, y, G_0, L_S)$ и существует последовательность субъектов $s_1, \ldots, s_m$ в S_0, где $s_1 = x, s_m = z$ и $m \geqslant 2$, таких, что выполняется условие 2. Следовательно, существует последовательность субъектов $s_1, \ldots, s_m, s_{m+1}$ в S_0, где $s_1 = x$, $s_m = z$, $s_{m+1} = y$, таких, что выполняется условие 2 теоремы.

Четвертый случай: существуют последовательности субъектов $s_1, \ldots, s_m$ и $s'_1, \ldots, s'_n$ в S_0, где $s_1 = x, s_m = s'_1 = z$, $s'_n = y$ и $m, n \geqslant 2$, для которых выполняется условие 2. Следовательно, существует последовательность субъектов $s''_1, \ldots, s''_{m+n-1}$ в S_0, где $s''_1 = x$, $s''_{m+n-1} = y$, таких, что выполняется условие 2 теоремы.

Индуктивный шаг доказан. Доказательство необходимости выполнения условия теоремы для истинности предиката $can\_steal\_own(x, y, G_0, L_S)$ закончено. Теорема доказана. ■

ЛИТЕРАТУРА

1. *Девянин П. Н.* Анализ безопасности управления доступом и информационными потоками в компьютерных системах. М.: Радио и связь, 2006. 176 с.
2. *Девянин П. Н.* Модели безопасности компьютерных систем: Учеб. пособие для студ. высш. учеб. заведений. М.: Издательский центр «Академия», 2005. 144 с.
3. *Робачевский А.* Операционная система UNIX. СПб.: БХВ-Петербург, 2000. 528 с.
4. *Качанов М. А., Колегов Д. Н.* Расширение функциональности системы безопасности ядра Linux на основе подмены системных вызовов // Прикладная дискретная математика. 2008. № 2. С. 76 – 80.