УДК 004.94

РАЗРАБОТКА РОЛЕВОЙ МОДЕЛИ БЕЗОПАСНОСТИ УПРАВЛЕНИЯ ДОСТУПОМ И ИНФОРМАЦИОННЫМИ ПОТОКАМИ КОМПЬЮТЕРНОЙ СИСТЕМЫ SELinux[1]

М. А. Качанов

В настоящее время наиболее рапространенным методом реализации безопасного управления доступом и информационными потоками в операционных системах (ОС) семейства $GNU/Linux$ является применение программного средства $SELinux$, в связи с чем возникает задача разработки формальной модели его безопасности. В данной работе решается задача анализа безопасности управления доступом и информационными потоками в компьютерной системе (КС) $SELinux$, предлагаются ролевая модель безопасности данной КС, алгоритм проверки возможности получения права доступа и реализации информационного потока, а также метод применения данной модели на практике для анализа безопасности рассматриваемой КС.

Будем использовать основные понятия и обозначения теории ДП-моделей, считая, что моделируемая КС представляется системой, каждое состояние в которой задаётся набором объектов, а каждый переход из состояния в состояние осуществляется в результате применения одного из правил преобразования состояний. Определим новые элементы ДП-модели, необходимые для адекватного анализа безопасности КС $SELinux$:

T — множество типов сущностей;

M — множество известных классов сущностей;

R_M — множество прав доступа, допустимых для известных классов сущностей;

$label : E \to U \times R \times T$ — функция, сопоставляющая сущности контекст безопасности (метку);

$user : E \to U$, $role : E \to R$, $type : E \to T$ — такие функции, что если $e \in E$ и $label(e) = (u, r, t)$, то $user(e) = u$, $role(e) = r$, $type(e) = t$;

$class : E \to M$ — функция, сопоставляющая каждой сущности известный класс;

$allow\_role : R \to 2^R$ — функция, сопоставляющая каждой роли множество ролей, которые она может занять;

$role\_types : R \to 2^T$ — функция, сопоставляющая каждой роли множество типов субъектов, к которым ей разрешено получать доступ;

$user\_roles : U \to 2^R$ — функция, сопоставляющая каждому пользователю множество ролей, на которые он может быть авторизован;

$class\_perms : M \to 2^{R_M}$ — функция, сопоставляющая каждому известному классу сущностей набор прав доступа, к нему применимый;

$R_r = \{(c, p) : c \in M, p \in class\_perms(c)\}$ — множество видов прав доступа;

$P \subseteq S \times E \times (R_r \cup \{own_r\})$ — множество текущих прав доступа субъектов к сущностям;

функции $fa : U \times E \to 2^E$, $fp : U \times E \to 2^E$, $ft : S \times E \times R_r \to 2^E$, $type\_rights : T \times T \to 2^{R_r}$, $role\_transition : R \times T \to R$, $type\_transition : T \times T \times M \to 2^T$, $login : R \times T \to 2^{R \times T}$, $constrain : R_r \times (U \times R \times T)^2 \to \{0, 1\}$, $validatetrans : M \times (U \times R \times T)^3 \to \{0, 1\}$.

[1]Работа выполнена в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг. (гос. контракт № П1010).

Определение 1. Иерархия известных классов сущностей — это заданное на множестве M отношение частичного порядка $\leqslant_M$, такое, что

$$\forall m_1, m_2 \in M \ (m_1 \leqslant_M m_2 \Leftrightarrow class\_perms(m_1) \supseteq class\_perms(m_2)).$$

Определение 2. Совокупность объектов $G = (U, U_L, R, E, S, L_S, T, F, M, R_M, P, label, allow\_role, role\_types, user\_roles, class\_perms, type\_rights, role\_transition, type\_transition, H_E, login, constrain, validatetrans, \leqslant_M)$ будем называть состоянием системы.

В модели определены 19 правил преобразования состояний, образующих множество OP.

Используем следующие обозначения:

$\Sigma(G^*, OP)$ — система, где 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 $op \in OP$.

Предположение 1. В рамках модели КС *SELinux* на траекториях функционирования системы доверенные пользователи не создают новых субъектов, доверенные субъекты не участвуют в передаче прав доступа, не реализуют информационных потоков по времени, не создают субъектов, не используют права доступа владения к другим субъектам.

Предположение 2. В рамках модели КС *SELinux* на траекториях функционирования системы не изменяются множества U, U_L, L_S, R, T, M, R_M, отношение $\leqslant_M$ на множестве M, функции $allow\_role$, $role\_types$, $user\_roles$, $class\_perms$, $type\_rights$, $role\_transition$, $type\_transition$, $login$, предикаты $constrain$, $validatetrans$.

Ввиду предположения 2 состояние системы будем записывать как $G = (S, E, P, F, H_E, class)$.

Определение 3. Система $\Sigma(G^*, OP)$ с множеством правил преобразования состояний OP называется ДП-моделью КС *SELinux*, если её состояния подчиняются определению 2 и удовлетворяют предположениям 1 и 2.

Для формализации возможности получения права доступа и реализации информационного потока сформулируем определения предикатов безопасности ДП-модели КС *SELinux*.

Определение 4. Пусть $G_0 = (S_0, E_0, P_0, F_0, H_{E0}, class_0)$ — состояние системы $\Sigma(G^*, OP)$, в котором существуют пользователь $u \in U$, право доступа $\alpha_r \in R_r$ и сущность $e \in E_0$. Определим предикат $can\_share(u, e, \alpha_r, G_0)$, который будет истинным тогда и только тогда, когда существуют состояния $G_1, \ldots, G_N = (S_N, E_N, P_N, F_N, H_{EN}, class_N)$ и правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ и существует субъект $x \in S_N$, такой, что $user(x) = u$, и $(x, e, \alpha_r) \in P_N$, где $N \geqslant 0$.

Определение 5. Пусть $G_0 = (S_0, E_0, P_0, F_0, H_{E0}, class_0)$ — состояние системы $\Sigma(G^*, OP)$, в котором существуют сущности $x, y \in E_0$. Определим предикат $can\_write\_memory(x, y, G_0)$, который будет истинным тогда и только тогда, когда существуют состояния $G_1, \ldots, G_N = (S_N, E_N, P_N, F_N, H_{EN}, class_N)$ и правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ и $(x, y, write_m) \in F_N$, где $N \geqslant 0$.

Определение 6. Пусть $G_0 = (S_0, E_0, P_0, F_0, H_{E0}, class_0)$ — состояние системы $\Sigma(G^*, OP)$, в котором существуют сущности $x, y \in E_0$. Определим предикат $can\_write\_time(x, y, G_0)$, который будет истинным тогда и только тогда, когда существуют состояния $G_1, \ldots, G_N = (S_N, E_N, P_N, F_N, H_{EN}, class_N)$ и правила преобразования состояний $op_1, \ldots, op_N$, такие, что $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ и $(x, y, write_t) \in F_N$, где $N \geqslant 0$.

Замечание 1. Заметим, что для проверки истинности предикатов $can\_share(u, e, \alpha_r, G_0)$, $can\_write\_memory(x, y, G_0)$ и $can\_write\_time(x, y, G_0)$ в соответствии с определением требуется учесть все траектории функционирования КС, что не осуществимо на практике.

В связи с замечанием 1 представляется целесообразным сформулировать и обосновать алгоритмы проверки истинности предикатов can_share, can_write_memory и can_write_time. Такие алгоритмы реализуют преобразование начального состояния КС в его замыкание и позволяют проверить истинность предикатов для всех пользователей, сущностей и прав доступа одновременно. В работе вводятся определения замыканий ДП-модели КС $SELinux$ ($time$-замыкания и $memory$-замыкания), предлагаются и обосновываются алгоритмы их построения.

Предлагается также метод применения построенной модели на практике для проверки возможности получения права доступа и реализации информационного потока в КС $SELinux$. Метод состоит из двух основных этапов. Первый этап — это построение начального состояния предложенной ДП-модели КС $SELinux$ по набору конфигурационных файлов КС. Второй этап — это построение $time$-замыкания полученного состояния и интерпретация полученных результатов с точки зрения КС $SELinux$. На входе метод имеет весь набор необходимых конфигурационных файлов КС $SELinux$, а на выходе — ответ на вопрос, возможны ли получение заданного права доступа или реализация заданного информационного потока.

УДК 004.94

ОСОБЕННОСТИ РАЗРАБОТКИ ДП-МОДЕЛЕЙ СЕТЕВОГО УПРАВЛЕНИЯ ДОСТУПОМ[1]

Д. Н. Колегов

Работа посвящена особенностям построения ДП-моделей компьютерных систем (КС), реализующих сетевое управление доступом. Такие модели будем называть сокращенно СУД ДП-моделями, а при их описании используем основные определения и обозначения из [1].

Механизмы сетевого управления доступом в современных КС, как правило, обладают следующими свойствами, затрудняющими применение элементов и средств существующих ДП-моделей для их описания и исследования:

— распределенностью компонентов управления доступом и их сетевым взаимодействием;
— динамическим управлением доступом субъектов к сущностям на основе правил доступа и, как следствие, предоставлением субъектам различных прав доступа в зависимости от истинности условий того или иного правила доступа;

[1]Работа выполнена в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг. (гос. контракт № П1010).