УДК 004.94

# О ДОСТАТОЧНЫХ УСЛОВИЯХ ПОХИЩЕНИЯ ПРАВ ДОСТУПА В СУБД ДП-МОДЕЛИ

В. Ю. Смольянинов

Для обеспечения возможности теоретического анализа безопасности СУБД на основе РОСЛ ДП-модели [1] построена формальная модель (кратко, СУБД ДП-модель), а также сформулированы достаточные условия похищения прав доступа.

По сравнению с РОСЛ ДП-моделью, в модель внесены следущие изменения.

Множество объектов делится на множество сущностей-данных $O_d$, соответствующих записям таблиц, множество сущностей-процедур $O_p$ и множество сущностей-триггерров $O_t$, активизирующихся при выполнении операций над связанными с ними таблицами. Сущности-триггеры и сущности-процедуры содержат $SQL$-код, выполнение которого субъект-сессиями приводит к реализации преобразований системы, задаваемыми де-юре правилами модели.

На множестве сущностей-контейнеров $C$, соответствующих схемам и таблицам СУБД, задается функция иерархии сущностей $H$. В множество таблиц $T$ включаются те сущности-контейнеры, которые могут содержать только записи таблиц и сущности-триггеры.

В отличие от РОСЛ ДП-модели, элементы множества субъект-сессий $S$ не считаются сущностями. При этом считается, что субъект-сущности соответствуют сессиям пользователей с СУБД. С помощью функции $user$ задается соответствие между субъект-сессиями и элементами множества учётных записей пользователей $U$. Следует отметить, что при переходе системы из состояния в состояние допустимо изменение функции $user$. Это связано с особенностями выполнения кода хранимых процедур и триггеров в реальных СУБД, где код может выполняться с правами, отличными от прав активизировавшего их пользователя.

В модели считается, что множество функционально-ассоциированных с субъектом-сессией $s$ сущностей $[s]$ в каждый момент содержит ровно одну сущность-триггер или сущность-процедуру, т. е. $[s] \in O_p \cup O_t$.

Для учёта специфики реальных СУБД в модель введены следующие элементы:

$owner\colon E \to U$ — функция, задающая для каждой сущности учётную запись её владельца и удовлетворяющая условию $\forall t \in T\, \forall o \in H(t)\ (owner(o) = owner(t))$;

$execute\_as\colon O_p \cup O_t \to \{as\_caller, as\_owner\}$ — функция, задающая режим выполнения кода сущности-процедуры или сущности-триггера $o$ субъект-сессией $s$; если $execute\_as(o) = as\_caller$, то код сущности $o$ выполняется от имени учётной записи $user(s)$, иначе код сущности $o$ выполняется субъект-сессией $s$ от имени $owner(o)$;

$R_r = \{read, write, append, delete, alter, execute, create\}$ — множество видов прав доступа к сущностям-контейнерам и сущностям-процедурам, где $alter$ соответствует праву доступа, позволяющему задавать $SQL$-код сущностей-процедур и сущностей-триггеров, а $create$ — праву, позволяющему создавать дочерние контейнеры;

$R \subseteq U \times (C \cup O_p) \times R_r$ — множество прав учётных записей пользователей на контейнеры и сущности-процедуры, при этом де-юре правила модели обеспечивают выполнение следующего ограничения: если пользователь обладает правом на контейнер, то он обладает соответствующим правом доступа и на любой его дочерний элемент;

$Gr \subseteq U \times (C \cup O_p) \times R_r$ — множество, задающее учётные записи пользователей, а также права на контейнеры и сущности-процедуры, которые они могут предоставить на них другим учётным записям пользователей, при этом $Gr \subseteq R$, т. е. права

доступа могут быть предоставлены только в случае обладания ими учётной записью пользователя — инициатора предоставления прав;

$triggers\colon T \times \{write, append, delete\} \to 2^{O_t}$ — функция, задающая для таблицы множество сущностей-триггеров, активизирующихся при реализации к ней права доступа соответствующего вида;

$OP$ — множество де-юре правил преобразования состояний, включающее правила: создания новой субъект-сессии $create\_session(u, s)$; предоставления права доступа $grant\_right(s, u, e, \alpha_r)$; изменения и создания сущностей-процедур и сущностей-триггеров $alter\_procedure$ и $alter\_trigger$; выполнения указанной процедуры $execute\_procedure(s, o_p)$, а также для вставки $access\_insert(s, t, o_{d1}, o_{d2})$, обновления $access\_update(s, t, o_{d1}, o_{d2})$ и удаления записей $access\_delete(s, t, o_d)$;

$operations\colon O_p \cup O_t \to OP^*$ — функция, задающая для каждой сущности-процедуры и сущности-триггера реализуемые ими последовательности де-юре правил преобразования состояний.

**Определение 1.** Пусть $G_0$ — состояние системы $\Sigma(G^*, OP)$, $u \in U$, $e \in C \cup O_p$, $\alpha_r \in R_r$, $U' = \{u' \in U_0 : (u', e, \alpha_r) \in Gr_0\}$. Определим предикат $can\_steal\_right(u, e, \alpha_r, G_0)$, истинный тогда и только тогда, когда существуют состояния $G_1, \ldots, G_N$ и де-юре 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 $N \geqslant 0$; $(u, e, \alpha_r) \in R_N$; $op_i$ для всех $i = 1, \ldots, N$ не является правилом вида $grant\_right(s, u, e, \alpha_r)$, $alter\_procedure(s, \ldots)$, $alter\_trigger(s, \ldots)$, где $user_0(s) = owner_0([s]_0) \in U'$.

Справедливо следующее утверждение о достаточных условиях истинности предиката $can\_steal\_right(u, e, \alpha_r, G_0)$.

**Утверждение 1.** Пусть $G_0$ — состояние системы $\Sigma(G^*, OP)$, $u \in U$, $e \in C \cup O_p$, $\alpha_r \in R_r$. Если существует $u' \in U_0$, такой, что $(u', e, \alpha_r) \in Gr_0$ и выполняется одно из условий 1–2, то предикат $can\_steal\_right(u, e, \alpha_r, G_0)$ истинен.

У с л о в и е 1. $\exists u'' \in U_0\ \exists o_p \in O_{p_0}$ $(operations_0(o_p) = (op_1, \ldots, op_N)$ и $\exists i \in \{1, \ldots, N\}$ $\exists s \in S_0$ $(op_i = grant\_right(s, u, e, \alpha_r)$ и или $owner_0(o_p) = u'$, $(u'', o_p, execute) \in R_0$, $execute\_as_0(o_p) = as\_owner$, или $(u', o_p, execute) \in R_0$, $execute\_as_0(o_p) = as\_caller))$.

У с л о в и е 2. $\exists u'' \in U_0\ \exists t \in T\ \exists \alpha'_r \in \{append, write, delete\}\ \exists o_t \in triggers(t, \alpha'_r)$ $(operations_0(o_t) = (op_1, \ldots, op_N)$ и $\exists i \in \{1, \ldots, N\}\ \exists s \in S_0$ $(op_i = grant\_right(s, u, e, \alpha_r)$ и или $(u'', o_t, \alpha'_r) \in R_0$, $execute\_as_0(o_t) = as\_owner$, или $(u', o_t, \alpha'_r) \in R_0$, $execute\_as_0(o_t) = as\_caller))$.

**Следствие 1.** Пусть $G_0$ — состояние системы $\Sigma(G^*, OP)$, $u \in U$, $e \in C \cup O_p$, $\alpha_r \in R_r$. Если существуют $c' \in C$, $u' \notin \{u'' \in U_0 : (u'', e, \alpha_r) \in Gr_0\}$, такие, что $(u', c', alter) \in R$, то предикат $can\_steal\_right(u, e, \alpha_r, G_0)$ истинен.

В дальнейшем планируется развитие СУБД ДП-модели по следующим направлениям: расширение модели набором де-факто правил и определение необходимых и достаточных условий реализации информационных потоков в случаях наличия или отсутствия кооперации субъект-сессий.

## 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.