ЛИТЕРАТУРА

1. `https://tools.ietf.org/html/rfc6454` — The Web Origin Concept.
2. `http://code.google.com/p/rebind/` — DNS Rebinding Tool.

УДК 004.65, 004.056.52 DOI 10.17223/2226308X/8/35

АТРИБУТНОЕ УПРАВЛЕНИЕ ДОСТУПОМ К ХРАНИЛИЩУ ДАННЫХ ТИПА «КЛЮЧ — ЗНАЧЕНИЕ»

С. В. Овсянников, В. Н. Тренькаев

Предлагается способ разграничения доступа пользователей к хранилищу данных типа «ключ — значение», когда право на доступ вычисляется в зависимости от параметров запроса (тип операции, идентификатор данных, пароль). Данный способ апробирован при разработке NoSQL СУБД с сервером управления доступом и удалённым хранилищем данных.

Ключевые слова: *атрибутное управление доступом, хранилище данных типа «ключ — значение», NoSQL база данных.*

В современных СУБД нередко отсутствует реализация так называемого мелко гранулированного управления доступом к данным (fine-grained access control), когда требуется ограничить доступ пользователей к отдельным строкам таблицы (как в случае реляционной БД) или к отдельной паре «ключ — значение» (как в случае NoSQL-хранилища данных). В данной работе для этих целей предлагается использовать подход, который можно отнести к атрибутной модели управления доступом [1], когда субъект имеет право доступа к сущности, если истинен предикат, вычисленный от атрибутов субъекта и/или сущности. При этом рассмотрена ситуация, когда имеется возможность пользователям самим настраивать политику безопасности СУБД, определяя правила задания разграничительной политики доступа к ресурсам БД.

Далее будем иметь дело с хранилищем данных типа «ключ — значение» (key — value), т. е. когда база данных представляет собой набор записей, идентифицируемых по ключу. Формально ключ будем рассматривать как слово в заданном алфавите. В простейшем случае каждый ключ ставится в соответствие значению в виде произвольных данных, в усложнённом варианте значение связано с определённым типом данных (целые, строки, списки, множества). Хранилище пар «ключ — значение» отличается упрощённой моделью запросов, используется малый набор операций: установка (set), получение (get), удаление (delete) значений по ключу.

Предлагается задавать политику безопасности хранилища данных с помощью функции управления доступом $C : K \times O \times P \to \{Allow, Deny, Pass\}$, где K — множество префиксов ключей; $O = \{set, get, delete, access\}$ — множество операций, P — множество парольных слов. Значение функции C, равное $Allow$, изначально определяется не менее чем для одной тройки $(k, access, p) \in K \times O \times P$, и тот, кто знает пару (k, p), может условно считаться администратором хранилища данных. Кроме запросов к хранилищу на обработку данных по ключу $(set, get, delete)$, возможны также запросы на изменение политики безопасности $(access)$, т. е. на задание (изменение) значений функции C.

Пусть запрос к хранилищу данных имеет следующие параметры: key (идентификатор данных), $value$ (значение данных), op (операция), pas (пароль). С точки зрения управления доступом запрос обрабатывается, следуя двум правилам.

П р а в и л о 1. Инициатору запроса разрешено произвести операцию $op \in O$, если существует префикс pk слова key, такой, что $C(pk, op, pas) = Allow$, и при этом выполняется условие: если функция C определена для некоторого префикса $prefix$ слова pk, то $C(prefix, op, pas) = Pass$.

П р а в и л о 2. Инициатору запроса запрещено произвести операцию $op \in O$, если существует префикс pk слова k, такой, что $C(pk, op, pas) = Deny$, и при этом выполняется условие: если функция C определена для некоторого префикса $prefix$ слова pk, то $C(prefix, op, pas) = Pass$.

Считается, что префикс слова может совпадать с самим словом. Если условия правила 1 или правила 2 не выполняются, например, из-за того, что функция управления доступом не определена на параметрах запроса, то запрос не выполняется.

В таблице приведён простой пример задания функции управления доступом с использованием псевдокода. Для запросов с паролем «p1» разрешается выполнять любые операции над данными с ключами, которые начинаются с «a». Для запросов с ключом «ab» разрешается чтение данных всем, а запись только тем, кто знает пароль «p2».

Префиксы ключей	Псевдокод вычисления значения функции управления доступом
a	IF password = p1 return ALLOW; ELSE return PASS;
ab	IF operation = get return ALLOW; IF operation = set IF password = p2 return ALLOW; ELSE return DENY;

ЛИТЕРАТУРА

1. *Чернов Д. В.* О моделях логического управления доступом на основе атрибутов // Прикладная дискретная математика. Приложение. 2012. №. 5. С. 79–82.

УДК 004.62 DOI 10.17223/2226308X/8/36

THE CAPACITY OF A PACKET LENGTH COVERT CHANNEL

A. V. Epishkina, K. G. Kogos

Covert channels are used for information hiding and realize one of the most serious security threat. Widespread IP networks allow for designing such channels on the basis of special properties of packet data transfer. Packet length covert channels are resistant to traffic encryption, but some difficulties to detect them are known. It makes significant an investigation of capacity limitation methods. This work presents a technique to estimate and limit the capacity of the covert channels based on the packet length modulation by traffic padding.

Keywords: *covert channel, packet length, dummy packet, capacity limitation.*

A covert channel is a communication channel which is not intended for information transfer at all, such as the service program's effect on the system load [1]. At present the most popular covert channels are in packet networks because of some features available in the TCP/IP protocol suite. There is a number of undetectable packet length covert channels in IP networks that may be constructed even if an encryption is used at any OSI model level. This paper describes a technique to estimate and limit the capacity of such covert channels using dummy packets generation.

The design of the considered network covert channel and of a counteraction technique is as follows. Let the lengths of transferred packets have the natural values from l_{fix} to $l_{\text{fix}} + L$; $\{L_0, L_1\}$ is a partition of the set $N_{l_{\text{fix}}+L} \setminus N_L$ where $|L_0| = |L_1|$, N_a stands for the