нии такой информации; 3) предоставленная информация изолирована от БД локальных пользователей; 4) ответы внешних пользователей регистрируются в форме ответа на запрос и заносятся в БД только в виде добавлений, после предварительного тестирования и проверки.

Этот способ реализован в виде отдельной подсистемы External Security System (ESS). Детальное описание ESS рассмотрено в настоящем сборнике[6].

3. Результат синтеза целевой системы защиты.

Ввиду ограничений, рассмотренных в [1], в качестве оболочки реализации целесообразно выбрать открытую среду, поддерживающую архитектуру клиент–сервер, активное управление защитой, модели составных и распределенных составных объектов, открытый интерфейс с базой данных. Перечисленным требованиям удовлетворяет среда программирования MS Visual Basic [7, 8]. Поэтому результат синтеза целевой системы разграничения доступа выражается в виде двух частей: системы управления защитой и тестами защиты в виде ActiveX компонента, реализованного в среде MS Visual Basic, и системы генерации защиты на сервере БД в виде SQL скрипта. В результате запуска сгенерированного SQL скрипта создается специализированная БД защиты и API запомненных на сервере процедур.

Синтез вышеуказанных частей происходит на базе протестированных структур и алгоритмов путем компоновки модулей из соответствующих библиотек.

Средства Online и Offline функционального тестирования системы защиты также представлены в виде SQL скриптов.

4. Математическая модель.

Построенная модель защиты естественным образом отображается на соответствующую алгебру цепочек дескрипторов. Задачи построения полной системы эквивалентных преобразований для цепочек дескрипторов, их оптимизации по критериям объема и времени выполнения будут рассмотрены в наших последующих публикациях.

Математическое обоснование для тестов проверки системы защиты можно проводить в рамках модели, аналогичной рассмотренной в работе [9]. Пример построения тестов типа online, вытекающий из результатов построенных для этой модели, также будет рассмотрен в нашей последующей публикации.

5. Заключение.

Предложенная методика может быть использована как основа для построения интегрированной инструментальной среды для проектирования и тестирования систем распределения доступа. Конкретные примеры ее использования рассмотрены в настоящем сборнике [4, 6].

Существующие CASE системы разработки распределенных приложений могут быть дополнены средствами проектирования и тестирования защиты, позволяющими одновременно с проектированием приложения выполнять формализованное проектирование и тестирование СРД для данного приложения. Совмещение средств проектирования и тестирования СРД позволит уменьшить стоимость создаваемых приложений.

ЛИТЕРАТУРА

[1] *Shoukourian S.K., Vasilyan A.M., Shukurian A.K.* Draft on a virtual operating room for prediction of hearing function changes and operative interventions under some pathologies of middle ear, Proceedings of the International Conference on Critical Technologies, Yerevan, 1995.

[2] *Shoukourian S.K., Shukurian A.K., Avagyan A.A., Vasilyan A.M.* Designing a virtual operating room for prediction of operative interventions under some pathologies of middle ear. A case of the database, accepted for presentation and publication in Proceedings of 9th World Congress on Medical Informatics, Medinfo '98, Seoul, Korea, August, 1998.

[3] *Baker B.* Patient data and security: an overview. – International Journal of Medical Informatics, v. 49, March, 1998, p. 19-30.

[4] *Василян А.М.* Система разграничения доступа для пользователей некоторых распределенных приложений, функционирующих в рамках локальной сети. – См. настоящий сборник.

[5] *Norifusa M.* Internet security: difficulties and solutions. – International Journal of Medical Informatics, v. 49, March, 1998, p. 69-74.

[6] *Аваkян А.А.* Система разграничения доступа для пользователей некоторых распределенных приложений, подключенных через Internet. – См. настоящий сборник.

[7] Visual Basic® 5.0 Programmer's Guide. Microsoft® Press 1997.

[8] *William R.* Vaughn, Hitchhiker's Guide to Visual Basic®.& SQL Server™. Microsoft® Press 1997.

[9] *Shoukourian S.K.* A Unified designing methodology for offline and online testing, IEEE design and test of computers, April-June, 1998, p.72-78.

СИСТЕМА РАЗГРАНИЧЕНИЯ ДОСТУПА В РАМКАХ ЛОКАЛЬНОЙ СЕТИ ДЛЯ ПОЛЬЗОВАТЕЛЕЙ РАСПРЕДЕЛЕННЫХ ПРИЛОЖЕНИЙ

Василян А.

Ереванский государственный университет

Приведен пример применения методики проектирования и тестирования систем разграничения доступа (СРД) для распределенных приложений, использующих базы данных. Пример приведен для двухуровневой архитектуры клиент–сервер, функционирующей в рамках локальной сети, где сервер представлен СУБД MS SQL Server 6.5. Описана реализация механизма распределения доступа и рассмотрены примеры ее использования клиентом, реализованным в среде MS Visual Basic 5.0.

***Վասիլյան Ա.*, Հասանելիության տարբերակման համակարգ լոկալ համակարգչային ցանցում գործող, տեղաբաշխված կիրառական ծրագրերի համար.** Հոդվածում դիտարկված է տվյալների հենքերի վրա հիմնված, տեղաբաշխված կիրառական ծրագրերի համար հասանելիության տարբերակման համակարգի

(ՀՏՀ) նախագծման և տեստավորման մեթոդոլոգիայի մանրամասն օրինակ: Օրինակը ներկայացված է լոկալ համակարգչային ցանցում գործող երկմակարդակ կլիենտ–սերվեր ճարտարապետության համար, երբ սերվեր է հանդիսանում ՏՀՂՀ MS SQL Server 6.5: Դիտարկված է հասանելիության տարբերակման համակարգի իրականացման մեխանիզմը և MS Visual Basic 5.0 միջավայրում կլիենտի իրականացման օրինակը:

Vasilyan A. **An Access Differentiation System in Distributed Applications for Users in the Local Area Network.** The paper suggests a detailed example of applicaion of design and testing methodology for access differentiation systems (ADS) in distributed applications which use databases. An application example of the suggested technique for two-level client-server architecture for execution in the local area network is adduced. The server is implemented using DBMS MS SQL Server 6.5 and tools for a use by a client are implemented using MS Visual Basic 5.0 environment.

ВВЕДЕНИЕ. В работе [1] настоящего сборника описана методика проектирования и тестирования систем разграничения доступа (СРД) для приложений, основанных на базах данных. Методика основана на комбинации в рамках единой оболочки механизмов и инструментов разграничения доступа в СУБД и ОС, современного инструментария прикладного программирования и формальной верификации.

Проектирование системы защиты заключается в определении информационных структур СРД, на основе которых осуществляется защита и алгоритмов защиты, определенных на этих структурах. Одновременно с проектированием защиты на основе спецификации происходит проектирование структур и алгоритмов для тестов СРД.

Результатом проектирования является инструментальная среда [2], поддерживающая выбранную архитектуру, модели составных и распределенных составных объектов, интерфейс с базой данных. В рамках системы обеспечиваются средства Online и Offline тестирования системы. Синтез целевой СРД происходит на базе протестированных структур и алгоритмов путем компоновки модулей из соответствующих библиотек.

В настоящей статье рассмотрено применение методики проектирования и тестирования систем разграничения доступа для приложений, функционирующих в рамках локальной сети. Пример построения внутренней системы защиты (Internal Security System – ISS) приведен для двухуровневой архитектуры клиент–сервер, где сервер представлен СУБД MS SQL Server 6.5. Описана реализация механизма распределения доступа и рассмотрены примеры ее использования клиентом, реализованным в среде MS Visual Basic 5.0.

1. Ключевые объекты защиты.

В работе [1] приведено разделение объектов защиты на три класса: функциональные ресурсы, ресурсы информационного поля, ресурсы времени и рабочих станций.

При таком рассмотрении синтез системы можно проводить независимо от отдельных классов и рассматривать только взаимосвязь данного класса и функций, общих для всей системы.

Как указано в [1], составными частями общей спецификации объектов защиты являются: 1) список используемых классов объектов защиты; 2) способ идентификации пользователей (например, через сервер БД или через сервис идентификации сетевой операционной системы); 3) дополнительная поддержка журнала действий пользователей; 4) ограничение на общее количество пользователей системы; 5) ограничение на общий объем дисковой памяти, запрашиваемой системой защиты; 6) ограничение на время ответа на запрос о проверке полномочий системой защиты.

Рассмотрим сначала пример проектирования и тестирования защиты для класса функциональных ресурсов системы.

2. Проектирование и тестирование СРД для класса функциональных ресурсов.

2.1. ***Спецификация защиты класса.***

Спецификация защиты класса функциональных ресурсов системы содержит иерархическое описание функциональных ресурсов в виде списка, каждый элемент которого имеет следующий вид: 1) идентификатор функционального ресурса; 2) наименование функционального ресурса; 3) идентификатор функционального ресурса верхнего уровня; 4) максимальный уровень вложенности в иерархии функциональных ресурсов (FuncMaxLevel); 5) максимальное общее число функций (FuncMaxCount).

Кроме того, из общей спецификации защиты используется: 1) максимально возможное число пользователей системы (UsersMaxCount); 2) ограничение на объем дисковой памяти, запрашиваемой системой защиты (TotalDiskSpace).

2.2. ***Определение схем базовых структур защиты.***

Структурами, на которых основана защита класса функциональных ресурсов системы, являются: структура описания иерархии функциональных ресурсов (FTREE) и структура дескрипторов защиты функциональных ресурсов системы (FDESC).

Выбор длины идентификатора функционального ресурса F_id_Length происходит исходя из ограничения на общее число функций. Если число функций ≤ 255, то длина его дескриптора равна 1 байту, в противном случае выбирается длина, равная 2 байтам.

Схема структуры описания иерархии функциональных ресурсов системы (FTREE) приводится ниже.

Поле	Тип	Описание поля
F_id	Char (F_id_Length)	Идентификатор функционального ресурса
F_name	Varchar (32)	Наименование функционального ресурса
F_idX	Varchar (F_id_Length * FuncMaxLevel)	Идентификатор точного положения функции в иерархии функциональных ресурсов. Соответствует пути от корня дерева к функции.
F_level	Smallint	Уровень в иерархии функциональных ресурсов

Например, рассмотрим иерархию функций, приведенную на *рис. 1.*

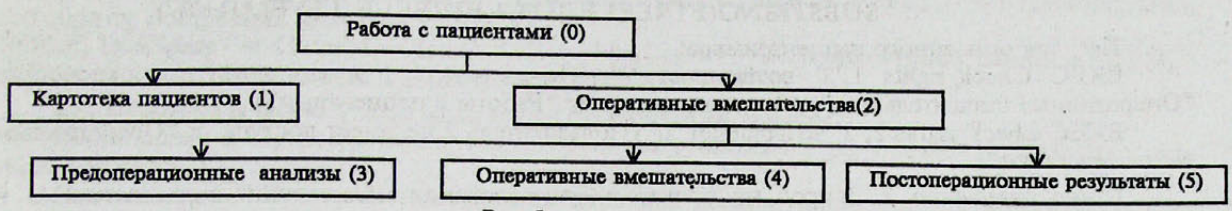

Рис. 1.

Структура, описывающая эту иерархию, выглядит следующим образом:

F_id	F_name	F_idX	F_level
0	Работа с пациентами	0	0
1	Картотека пациентов	01	1
2	Оперативные вмещательства	02	1
3	Предоперационные обследования	023	2
4	Оперативные вмешательства	024	2
5	Постоперационные результаты	025	2

Приведем также схему структуры дескрипторов защиты функциональных ресурсов системы (FDESC).

Поле	Тип	Описание поля
Uid	Smallint	Идентификатор пользователя или группы пользователей из системного каталога пользователей сервера БД (syslogins)
F_id	Char (F_id_Length)	Идентификатор функционального ресурса
Action	Char(1)	Право на доступ к функциональному ресурсу

По отношению к дескрипторам защиты функциональных ресурсов применяется техника наследования прав доступа сверху–вниз по иерархии. Следовательно, при отсутствии дескриптора доступа к данному ресурсу для пользователя за основу принимается ближайший дескриптор верхнего уровня.

Пример. Дескрипторы защиты для двух пользователей, где первый имеет доступ ко всем функциональным ресурсам (Uid=1), а второй (Uid=2) только к "Картотеке пациентов", для описанной иерархии имеют вид:

Uid	F_id	action
1	0	1
2	0	0
2	1	1

2.3. *Определение базовых схем алгоритмов над структурами защиты.*

Схема алгоритма 1. Запрос на проверку полномочий пользователя U на выполнение функции F.

```
CREATE PROCEDURE CHECK_RIGHTS
 (@U SMALLINT , @F CHAR (F_id_Length), @ACTION CHAR(1)=NULL OUTPUT)
/*      Параметры  процедуры
        @U              – идентификатор пользователя
        @F              – идентификатор функционального ресурса
        @ACTION         – значение, возвращаемое процедурой
*/
AS
/* Определение локальных переменных */
DECLARE @F_IDX VARCHAR(255)
/* Присвоить @F_IdX  путь к функции F по дереву FTREE */
 SELECT @F_IDX= FTREE.F_IDX
 FROM FTREE  WHERE FTREE.F_ID=@F
/* Вернуть значение поля action для дескриптора с наибольшим значением поля F_level, путь к
F_id которого является префиксом  @F_IdX */
SELECT @ACTION=FDESC.ACTION
```

```
    FROM FDESC, FTREE
    WHERE FDESC.SUID=@U AND FDESC.F_ID=FTREE.F_ID AND
          @F_IDX LIKE SUBSTRING(FTREE.F_IDX,1,FTREE.F_LEVEL*1)+'%'
              AND FTREE. F_LEVEL =
              ( SELECT MAX(FTREE1.F_LEVEL)
                FROM FDESC FDESC1, FTREE FTREE1
                WHERE FDESC1.SUID=@U AND
                    FTREE1.F_ID=FDESC1.F_ID AND
                    @F_IDX LIKE
                        SUBSTRING(FTREE1.F_IDX,1,FTREE1.F_LEVEL*1)+'%')
```

Так, для описанного выше примера:

EXEC Check_rights 1,'2' возвращает '1' (Пользователь 1 имеет доступ к подсистеме "Оперативные вмешательства", унаследованный от "Работы с пациентами")

EXEC Check_rights 2,'2' возвращает '0' (Пользователь 2 не имеет доступа к "Оперативным вмешательствам")

Схема алгоритма 2. Запрос на проверку существования полномочий пользователя U на выполнение подфункции функции F.

```
CREATE PROCEDURE CHECK_SUB_RIGHTS
 (@U SMALLINT , @F CHAR(F_id_Length), @ACTION CHAR(1)=NULL OUTPUT)
/* Параметры  процедуры
       @U          – идентификатор пользователя
       @F          – идентификатор функционального ресурса
       @ACTION     – значение, возвращаемое процедурой
*/
AS
/* Определение локальных перменных */
DECLARE @F_IDX VARCHAR(255)
DECLARE @REZULTS CHAR(1)
/ Присвоить @Rezults значение доступа пользователя @U к функции @F */
EXEC CHECK_RIGHTS @U,@F,@REZULTS OUTPUT
IF @REZULTS = '1'
  /* Если пользователь  @U имеет доступ к функции @F, то вернуть '1' */
  SELECT  @ACTION ='1'
ELSE BEGIN
 /* Присвоить @F_IdX  путь к функции F по дереву FTREE */
 SELECT @F_IDX= SUBSTRING(FTREE.F_IDX,1,FTREE.F_LEVEL*1)+'%'
  FROM FTREE  WHERE FTREE.F_ID=@F
/* Если существует функция, к которой пользователь  @U имеет доступ, и эта функция является
потомком функции @F по дереву FTREE, то вернуть '1', иначе вернуть '0' */
IF EXISTS(SELECT *
        FROM FDESC, FTREE
        WHERE FDESC.SUID=@U AND
            FDESC.ACTION='1' AND
            FTREE.F_ID=FDESC.F_ID AND
            FTREE.F_IDX LIKE @F_IDX)
 SELECT  @ACTION ='1'
 ELSE
  SELECT  @ACTION ='0'
 END
```

Так, для описанного выше примера:

EXEC Check_sub _rights 1,'0' возвращает '1' (Пользователь 1 имеет доступ непосредственно к "Работе с пациентами")

EXEC Check_sub _rights 2,'0' возвращает '1' (Пользователь 2 имеет доступ к подфункции "Картотека пациентов" функции "Работа с пациентами")

Схема алгоритма 3. Запрос на первичную оптимизацию дескрипторов подфункций функции F пользователя U.

Схема алгоритма 4. Запрос на предоставление или отмену доступа пользователю U к функции F.

Схема алгоритма 5. Запрос на предоставление или отмену доступа пользователю U ко всем подфункциям функции F.

Схема алгоритма 6. Запрос на список функций, к которым имеет (не имеет) доступ пользователь U.

Схема алгоритма 7. Запрос на список пользователей, которые имеют доступ к функциональному ресурсу F.

Схема алгоритма 8. Добавление пользователя U в систему. Добавить дескриптор FDESC.uid=U, FDESC.F_id=0, ESC.action='0' (пользователь с нулевыми правами).

Схема алгоритма 9. Удаление пользователя U из системы.

2.4. *Тестирование выполнения ограничений, задаваемых в спецификации.*

Тестирование ограничений сводится к проверке следующих соотношений:
<TotalDiskSpace> = <FTREE DiskSpace> + <FDESC DiskSpace>.
FTREE DiskSpace> = <FuncMaxCount> * (<Размер строки таблицы FTREE> + <Пространство, необходимое построения индексов FTREE>).
<FDESC DiskSpace> = <FuncMaxCount> * <FuncMaxCount> * (<Размер строки таблицы FDESC> + <Пространство, необходимое для построения индексов FDESC.>).

2.5. *Определение базовых схем тестов над структурами защиты.*

Схема теста 1. Проверка целостности структуры описания иерархии функциональных ресурсов (FTREE).

Схема теста 2. Проверка целостности структуры дескрипторов защиты функциональных ресурсов (FDESC).

2.6. *Синтез результирующей системы защиты.* Результирующая система состоит из системы защиты и тестов на сервере БД и ActiveX элемента управления защитой и тестами.

Результат генерации системы защиты и тестов на сервере БД это SQL скрипт [3], включающий: 1) создание структур FTREE и FDESC, на основе выбранных схем базовых структур защиты; 2) перекачку описания иерархии функциональных ресурсов в FTREE; 3) создание и настройку схем алгоритмов 1-7 в виде хранимых на сервере БД SQL процедур, схем алгоритмов 8 и 9 в виде триггеров добавления и удаления syslogins; 4) генерацию первичных дескрипторов защиты функциональных ресурсов системы: администратору – разрешающий дескриптор на корень, остальным пользователям – дескриптор нулевых полномочий; 5) создание и настройку схем тестов 1 и 2 в виде хранимых на сервере БД SQL процедур.

ActiveX элемент управления [5, 6] защитой и тестами обеспечивает графический интерфейс пользователя для динамического управления доступом на основе вызовов алгоритмов 1-6 и для запуска тестов 1 и 2. Позволяет администратору производить сеанс редактирования полномочий пользователей системы.

4. Проектирование и тестирование СРД для класса ресурсов информационного поля системы.

Техника проектирования и тестирования защиты для класса ресурсов информационного поля аналогична технике проектировании СРД для класса функциональных ресурсов системы. Особенностью класса ресурсов информационного поля является их многочисленность и динамизм. Поэтому естественной является трехуровневая иерархия – таблица БД/группа записей таблицы БД/запись в таблице БД. Введение иерархии с большим числом уровней приводит к аномалиям включения и изменения.

Многочисленность ресурсов третьего уровня иерархии диктует использование дескрипторов защиты ресурсов информационного поля только для двух верхних уровней иерархии.

ЛИТЕРАТУРА

[1] *Авакян А.А., Василян А.М.* Методика проектирования и тестирования систем разграничения доступа для распределенных приложений, основанных на базах данных. – Настоящий сборник.

[2] *Shoukourian S.K., Shukurian A.K., Avagyan A.A., Vasilyan A.M.* Designing a virtual operating room for prediction of operative interventions under some pathologies of middle ear. – A case of the database, accepted for presentation and publication in Proceedings of 9th World Congress on Medical Informatics, Medinfo '98, Seoul, Korea, August 1998.

[3] *Baker B.* Patient data and security: an overview. International Journal of Medical Informatics, v. 49, March, 1998, p. 19-31.

[4] *Vaughn W.R.* Hitchhiker's Guide to Visual Basic® & SQL Server™, Microsoft® Press 1997.

[5] *Chappell D.* Understanding ActiveX™ and OLE, Microsoft® Press 1996.

[6] Visual Basic® 5.0 Programmer's Guide. Microsoft® Press 1997.