DOI: 10.18372/2410-7840.19.11442
УДК 004.657

АУДИТ НАСТРОЕК БЕЗОПАСНОСТИ MS SQL SERVER

Анастасия Мазуренко

Статья посвящена актуальной проблеме защите информации в базах данных. Корректно организованная политика безопасности позволяет обеспечить защиту конфиденциальной информации в базе данных. Политики безопасности гарантируют, что просмотр и/или изменения такой информации осуществляется только пользователями, имеющими на это право. Это минимизирует риски атак на информацию, а также увеличивает гарантии того, что система управления безопасностью охватила конфиденциальную информацию из всех баз данных организации. Важнейшим аспектом поддержания корректной политики безопасности является контроль настроек безопасности. Из множества доступных параметров, автор выделяет основные, сгруппировав их в пять категорий, которые должны контролироваться администратором в обязательном порядке: контроль доступа к SQL Server с административными правами; аудит разрешений CONTROL и IMPERSONATE; контроль владельцев баз данных; права доступа к объектам базы данных; контроль неудачных попыток входа. Для контроля настроек безопасности по каждой из этих категорий, автор предлагает набор процедур в виде SQL-запросов.

***Ключевые слова**: база данных, защита объектов БД, аудит настроек безопасности, информационная система.*

ЦЕЛЬ И ПОСТАНОВКА ЗАДАЧИ АУДИТА НАСТРОЕК БЕЗОПАСНОСТИ MS SQL SERVER

Цель статьи. Согласно различным исследования в области безопасности баз данных, наиболее популярные уязвимости связаны с некорректностью настроек безопасности. К таким относятся, например, первые три важнейших уязвимости [1, 2]:

– предоставление излишних полномочий пользователям;

– злоупотребление имеющимися полномочиями;

– повышение полномочий пользователями.

MS SQL Server предоставляет мощные средства для аудита событий и настроек [3,4]. Это и традиционные системные хранимые процедуры, и доступные в последних версиях, системные представления, а также другие средства. При наличии большого количества объектов защиты, возникает необходимость автоматизировать процесс аудита. Известные руководства по настройке безопасности СУБД предоставляют настолько обширный материал, что администратору в некоторых случаях не просто определиться, какие параметры SQL Server необходимо контролировать в первую очередь, и как сделать это оптимальным образом. Проверка корректности настроек безопасности относится к этому случаю. Целью работы является разработка и систематизация процедур контроля настроек безопасности, позволяющих снизить риски возникновения наиболее популярных уязвимостей.

Постановка задачи. Из множества доступных настроек безопасности SQL Server необходимо определить такие, которые должны контролироваться на регулярной основе. Для каждой из таких настроек необходимо разработать SQL-скрипт проверки ее состояния, позволяющий автоматизировать процесс аудита.

ВЫБОР ОСНОВНЫХ НАПРАВЛЕНИЙ АУДИТА И ОПРЕДЕЛЕНИЕ ПРОЦЕДУР КОНТРОЛЯ ОБЪЕКТОВ

Автор выделяет пять категорий настроек, которые должны контролироваться обязательно и регулярно:

1. Контроль доступа к SQL Server с административными правами.
2. Аудит разрешений CONTROL и IMPERSONATE.
3. Контроль владельцев баз данных.
4. Контроль права доступа к объектам базы данных.
5. Контроль неудачных попыток входа.

Рассмотрим подробнее особенности реализации процедур аудита в каждой из указанных категорий.

Контроль доступа к SQL SERVER с административными правами. Традиционно, объектами контроля являются:

– Учетная запись администратора sa.

– Члены серверной роли sysadmin.

Автор считает, что так же необходимо контролировать членов роли securityadmin.

Контроль учетной записи sa. Необходимо придерживаться правила – учетная запись sa не должна использоваться. По аналогии с учетной

записью Administrator в ОС Windows, учетную запись sa необходимо переименовать либо заблокировать ее соединение с SQL Server.

В SQL Server, учетная запись sa имеет ID равным 1. Это свойство можно использовать, для определения того, переименована или заблокирована учетная запись.

```
SELECT name, is_disabled
FROM sys.sql_logins
WHERE principal_id = 1; .
```

Контроль системных ролей sysadmin и securityadmin. Контроль членов роли sysadmin должен выполняться обязательно, поскольку участники роли имеют полный набор прав в SQL Server. Контроль членов роли securityadmin так же необходимо контролировать, поскольку они могут создавать имена входа (login), с правами, эквивалентными набору прав роли sysadmin. Начиная с SQL Server 2005 роль securityadmin дает возможность предоставить право доступа CONTROL к SQL Server, что эквивалентно членству в роли sysadmin.

Аудит права доступа CONTROL можно вести с помощью системных представлений sys.server_principals и sys.server_role_members, выполняя следующий запрос:

```
SELECT R.name AS 'Role', L.name AS 'Login'
FROM sys.server_principals AS L
JOIN sys.server_role_members AS RM ON L.principal_id = RM.member_principal_id
JOIN sys.server_principals AS R ON R.principal_id = RM.role_principal_id
WHERE R.name IN ('sysadmin', 'securityadmin')
ORDER BY R.name, L.name; .
```

Аудит состояния полномочий и их имперсонации. В SQL Server предоставление полномочий выполняется не только через механизм ролей. Права доступа можно назначать отдельным объектам защиты (securables). Такими объектами могут являться экземпляр сервер в целом, база данных, имена входа и пользователи либо отдельные объекты.

Аудит полномочия CONTROL. Для безопасности экземпляра SQL Server и базы данных (БД) в целом, очень важно, кто из пользователей имеет полномочие CONTROL. Это полномочие позволяет полностью управлять объектом защиты. Поскольку объекты сервера являются контейнерами, содержащими другие объекты защиты, то указанное полномочие позволяет получить доступ ко всей иерархии объектов контейнера. Например, SQL Server как объект защиты, является контейнером, содержащим базы данных, имена входа и другие объекты. База данных является контейнером для пользователей, схем и других объектов уровня базы данных. В свою очередь контейнер Схема содержит традиционные для БД объекты (таблицы и прочее).

Полномочия, предоставленные на уровне контейнера, наследуются объектами в него входящими. Если учетная запись имеет полномочия CONTROL к базе данных, то такие же полномочия она имеет и для объектов БД. Поэтому аудит полномочия CONTROL важен для безопасности. Проверка таких полномочий на уровне сервера можно выполнить запросом:

```
SELECT L.name, P.state_desc, P.permission_name
FROM sys.server_permissions AS P
JOIN sys.server_principals AS L ON P.grantee_principal_id = L.principal_id
WHERE P.class = 100 AND P.type = 'CL'
ORDER BY L.name; .
```

В этом и следующих запросах используется отбор по значению атрибута class системного представления sys.server_permissions. Согласно [5], значения атрибута определяют уровень объекта, для которого установлены полномочия. Например:

0 = Database.
1 = Object or Column.
3 = Schema.
4 = Database Principal.

На уровне отдельной БД проверить наличие полномочия CONTROL можно таким запросом:

```
SELECT U.name, P.state_desc, P.permission_name
FROM sys.database_permissions AS P
JOIN sys.database_principals AS U ON P.grantee_principal_id = U.principal_id
WHERE P.class = 0 AND P.type = 'CL'; .
```

Имперсонация для повышения уровня полномочий. Смена контекста выполнения команд в SQL Server возможна, если для учетной записи установлено полномочие IMPERSONATE. Эта возможность может привести к необоснованному повышению уровня полномочий. Поэтому важно вести аудит учетных записей с полномочием IMPERSONATE и контролировать адекватность полномочий политике безопасности. На уровне сервера сделать это можно с помощью запроса:

```
SELECT L.name, P.state_desc, P.permission_name, I.name
FROM sys.server_permissions AS P
JOIN sys.server_principals AS L
ON P.grantee_principal_id = L.principal_id
JOIN sys.server_principals AS I ON P.major_id = I.principal_id
```

```
WHERE P.class = 101 AND P.type = 'IM'
ORDER BY L.name, I.name; .
```

И на уровне базы данных:

```
SELECT U.name, P.state_desc, P.permission_name, I.name
FROM sys.database_permissions AS P
JOIN sys.database_principals AS U ON P.grantee_principal_id = U.principal_id
JOIN sys.database_principals AS I ON P.major_id = I.principal_id
WHERE P.class = 4 AND P.type = 'IM'
ORDER BY U.name, I.name; .
```

Аудит владельцев БД и членов роли db_owner. Если учетная запись является владельцем БД, она автоматически является членом роли sysadmin и соответствует пользователю dbo, который имеет неограниченные права в БД. Вывести список баз и их владельцев можно запросом:

```
SELECT D.name AS 'Database', L.name AS 'Owner'
FROM sys.databases AS D
LEFT JOIN sys.server_principals AS L ON D.owner_sid = L.sid
ORDER BY D.name; .
```

Возможна ситуация, когда в колонке 'Owner' указано NULL. Это значит, что учетная запись владельца БД была удалена, а новый владелец назначен не был. Такую ситуацию необходимо срочно исправлять.

Проверку принадлежности пользователей к роли db_owner можно выполнить с помощью системной хранимой процедуры

```
EXEC sp_helprolemember 'db_owner'; .
```

Аудит прав доступа к базе данных. Для приложений, работающих с БД, особенно важно контролировать права доступа к объектам базы.

Для объектов, уровня базы данных, необходимо контролировать права доступа для всех учетных записей. Следующий запрос выведет список полномочий на уровне базы данных (полномочие Connect не рассматриваем):

```
SELECT U.name, P.state_desc, P.permission_name
FROM sys.database_permissions AS P
JOIN sys.database_principals AS U ON P.grantee_principal_id = U.principal_id
WHERE P.class = 0 AND NOT P.type = 'CO'
ORDER BY U.name, P.permission_name; .
```

На уровне схемы:

```
SELECT U.name AS 'User', P.state_desc, P.permission_name, S.name AS 'Schema'
FROM sys.database_permissions AS P
JOIN sys.database_principals AS U ON P.grantee_principal_id = U.principal_id
JOIN sys.schemas AS S ON P.major_id = S.schema_id
WHERE P.class = 3
ORDER BY U.name, S.name; .
```

И на уровне таблиц и представлений:

```
SELECT U.name AS 'User', P.state_desc, P.permission_name,
S.name + '.' + O.name AS 'Object'
FROM sys.database_permissions AS P
JOIN sys.database_principals AS U ON P.grantee_principal_id = U.principal_id
JOIN sys.objects AS O ON P.major_id = O.object_id
JOIN sys.schemas AS S ON O.schema_id = S.schema_id
WHERE P.class = 1
ORDER BY U.name, O.name, P.permission_name; .
```

Если же необходимо контролировать права доступа к отдельным колонкам таблицы, последний запрос нужно модифицировать, присоединив системное представление sys.columns. Колонка major_id представления определяет объект, колонка minor_id определяет нужную колонку.

Аудит неудачных попыток входа. Есть две причины для проведения аудита неудачных попыток входа:

– Обнаружение атак на SQL Server.

– Для определения причин, почему человек или приложение с законным основанием для доступа к SQL Server не может подключиться.

Как правило, основной является первая причина. Однако и вторая причина важна для анализа правильности настроек системы безопасности. Анализируя журнал, можно понять причину отказа в доступе, например, неправильный пароль (рис. 1).

Дата	Источник	Сообщение
25.12.2016 7:54:00	Logon	Login failed for user 'Test'. Причина: пароль не соответствует переданному имени входа. [КЛИЕНТ: 192.168.1.129]
25.12.2016 7:54:00	Logon	Ошибка: 18456, серьезность: 14, состояние: 8.

Рис. 1 Логи ошибок

Если неудачных попыток входа не зарегистрировано, а приложение не может соединиться с сервером, причину нужно искать в настройках брандмауэра, сетевых проблемах или некорректной строке соединения.

Включение режима аудита неудачных попыток доступа выполняется с помощью SQL Server

Management Studio (SSMS) в свойствах сервера БД. Этот режим устанавливается по умолчанию. Анализ событий выполняется так же с помощью SSMS. Поскольку SQL Server регистрирует события и в журнале регистрации ОС, поиск событий можно вести и там.

ЗАКЛЮЧЕНИЕ

В статье предложены разработанные автором систематизированные процедуры контроля настроек безопасности, позволяющие снизить риски возникновения наиболее популярных уязвимостей. Таким образом обеспечивается поддержка корректной политики безопасности баз данных. Набор SQL-запросов может быть оформлен в виде хранимых процедур и использоваться администраторами баз данных как средство автоматизации контроля настроек безопасности. Наличие такого средства позволяет упростить задачу администрирования базы данных, оперативно реагировать на изменение должностных обязанностей и прав доступа пользователей.

ЛИТЕРАТУРА

[1]. SANS Institute InfoSec Reading Room. Setting Up a Database Security Logging and Monitoring Program. Access mode: World Wide Web. – URL: https://www.sans.org/reading-room/whitepapers/application/setting-database-security-logging-monitoring-program-34222.

[2]. Top Ten Database Security Threats. Access mode: World Wide Web. – URL: https://www.imperva. com/ docs/ gated/ WP_TopTen_ Database_Threats. pdf.

[3]. Access mode: World Wide Web. – URL: http://ebook-dl.com/item/securing-sql-server-third-edition-protecting-your-database-from-attackers-3rd-edition-denny-cherry.

[4]. Chapter 18 - Securing Your Database Server. Access mode: World Wide Web. – URL: https:// msdn. microsoft. com / en-us / library / ff648664.aspx.

[5]. sys.database_permissions (Transact-SQL) Access mode: World Wide Web. – URL: https:// msdn.microsoft.com/en-us/library/ms188367.aspx.

REFERENCES

[1]. SANS Institute InfoSec Reading Room. Setting Up a Database Security Logging and Monitoring Program. Access mode: World Wide Web. – URL: https://www.sans.org/reading-room/whitepapers/application/setting-database-security-logging-monitoring-program-34222.

[2]. Top Ten Database Security Threats. Access mode: World Wide Web. – URL: https://www.imperva.com/docs/gated/WP_TopTen_ Database_Threats.pdf.

[3]. Access mode: World Wide Web. – URL: http://ebook-dl.com/item/securing-sql-server-third-edition-protecting-your-database-from-attackers-3rd-edition-denny-cherry.

[4]. Chapter 18 - Securing Your Database Server. Access mode: World Wide Web. – URL: https:// msdn. microsoft. com / en-us / library / ff648664.aspx.

[5]. sys.database_permissions (Transact-SQL) Access mode: World Wide Web. – URL: https:// msdn. microsoft.com/en-us/library/ms188367.aspx

АУДИТ НАЛАШТУВАНЬ БЕЗПЕКИ MS SQL SERVER

Стаття присвячена актуальній проблемі захисту інформації в базах даних. Коректно організована політика безпеки дозволяє забезпечити захист конфіденційної інформації в базі даних. Політики безпеки гарантують, що перегляд та/або зміни такої інформації здійснюється лише користувачами, які мають на це право. Це мінімізує ризики атак на інформацію, а також збільшує гарантії того, що система управління безпекою охопила конфіденційну інформацію з усіх баз даних організації. Найважливішим аспектом підтримки коректної політики безпеки є контроль параметрів безпеки. З безлічі доступних параметрів, автор виділяє основні, згрупувавши їх у п'ять категорій, які повинні контролюватися адміністратором в обов'язковому порядку: контроль доступу до SQL Server з адміністративними правами; аудит дозволів CONTROL і IMPERSONATE; контроль власників баз даних; права доступу до об'єктів бази даних; контроль невдалих спроб входу. Для контролю параметрів безпеки по кожній з цих категорій, автор пропонує набір процедур у вигляді SQL-запитів.

Ключові слова: база даних, захист об'єктів БД, аудит налаштувань безпеки, інформаційна система.

MS SQL SERVER SECURITY SETTINGS AUDIT

The article is devoted to the trending problem of the information protection in databases. Correctly organized security policy allows you to protect sensitive information in database. Security policies ensure that view and / or modification of such information should be performed only by users who have appropriate rights. This minimizes the risks of attacks on the information, and also increases the assurance that the security management system encompass all confidential information from the organization's database. The most important aspect of maintaining proper security policy is to control the security settings. From the set of available options, the author identifies the main, grouped into five categories, which should be controlled by the administrator on a mandatory basis: access control to SQL Server with administrative privileges; audit CONTROL and IMPERSONATE permissions; database owners control; access permissions to the database objects; failed login attempts control. In order to control security settings, the author proposes a set of procedures in the form of SQL-queries for each of these categories.

Keywords: database (DB), DB objects protection, security settings audit, information system.

Мазуренко Анастасія Євгенівна, студентка Фізико-технічного інституту НТУУ «КПІ».
E-mail: ks0610@mail.ru

Мазуренко Анастасия Евгениевна, студентка Физико-технического института НТУУ «КПИ».

Mazurenko Anastasiia, student of the Institute of Physics and Technologies of the NTUU "KPI".