Труды МАИ. 2022. № 127
Trudy MAI, 2022, no. 127

Научная статья
УДК 004.658.2
DOI: 10.34759/trd-2022-127-13

КИБЕРБЕЗОПАСНОСТЬ ФУНКЦИОНИРОВАНИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ С УЧАСТИЕМ SQL SERVER

Ольга Витальевна Вакульчик
Московский авиационный институт (национальный исследовательский университет), МАИ, Москва, Россия
olga.vakulchik@mail.ru

Аннотация. Одним из основных компонентов информационно-управляющей системы (ИУС) авиационно-космического комплекса является база данных (БД). Данная статья посвящена обеспечению кибербезопасности ИУС путём разработки метода эффективной оценки качества функционирования ИУС с участием SQL Server, реализуемого как часть высокоуровневого протокола защищенной клиент-серверной системы дистанционного мониторинга состояния ИУС. Разработанный метод обеспечивает не только информационную полноту мониторинга, но и безопасность передачи ценной информации в открытой среде передачи, благодаря выбранному инструменту мониторинга и оптимизации созданных административных SQL-запросов.

Ключевые слова: кибербезопасность, информационно-управляющие системы, дистанционный мониторинг состояния, удаленное администрирование, система управления базами данных, SQL Server, SQL-запрос.

Для цитирования: Вакульчик О.В. Кибербезопасность функционирования информационно-управляющей системы с участием SQL server // Труды МАИ. 2022. № 127. DOI: 10.34759/trd-2022-127-13

Original article

CYBERSECURITY OF THE FUNCTIONING OF THE INFORMATION MANAGEMENT SYSTEM WITH THE SQL SERVER

Olga V. Vakulchik
Moscow Aviation Institute (National Research University),
Moscow, Russia
olga.vakulchik@mail.ru

Abstract. This article regards the issue of cyber-security of the information management system of the aerospace system (IMS). Database is the basic IMS component. The SQL server was employed as the database system. Monitoring execution is necessary for the Database information processes performance and security support. The article describes the development of the remote monitoring effective method. This method solves two basic problems, namely information transmission security in the open environment, and effective assessment of the information environment state.

Analysis revealed that effective remote monitoring ensuring requires tool selection. The selected tool should wield enhanced security and employ SQL language for the Database performance evaluation. The tool tackled with in this article ensures information encoding in the open transmission environment. It acts as an administrative panel as well.

After the tool selection, the SQL language objects analysis was performed.

Selection criteria of the necessary objects are as follows: the provided information completeness, request preparation complexity, and the internal functionality enhancing capability. Analysis revealed that storable system procedures are the most informative.

The next stage of development consisted in the stored procedures selection according to the “integral informativity” criterion. The sp_whoisactive stored procedure is of specially great capabilities. However, it has been found that these procedures were redundant. That is why optimized requests for the remote monitoring effectiveness improving were developed. The optimized requests reduce the status information volume, ensuring security, and provide herewith the remote monitoring completeness.

Keywords: cybersecurity, the information management system, remote status monitoring, remote administration, database management system, SQL Server, SQL query

For citation: Vakulchik O.V. Cybersecurity of the functioning of the information management system with the SQL Server. *Trudy MAI,* 2022, no. 127. DOI: 10.34759/trd-2022-127-13

Введение. Проблемы и задачи дистанционного мониторинга состояния

С появлением возможности автоматизации трудоёмкого процесса обработки данных вручную использование электронных баз данных во многих сферах жизнедеятельности стало обыденным [1], так, все информационные системы современных бортовых систем технического обслуживания и систем самолетовождения содержат в себе систему управления базами данных (СУБД) [2].

Необходимость поддержки производительности СУБД для автоматизированных систем управления стратегического звена управления Воздушно-Космическими силами описано в [3].

Повышение требований к защищенности ИС в авиационной отрасли приводит к необходимости создания и модернизации существующих систем защиты информации [4, 5], включая такие аспекты, как мониторинг состояния ИУС.

Дистанционный мониторинг состояния информационной среды удаленной информационной системы с участием SQL Server позволяет поддерживать устойчивость и безопасность информационных процессов [6]. Для оценки работоспособности сервера необязательно находиться рядом с сервером - лучше всего подходят приложения, позволяющие удаленно управлять базой данных и выполнять мониторинг состояния ИУС. К выбору инструментов для дистанционного мониторинга стоит подходить ответственно, поскольку при получении злоумышленником доступа к данным, тот сможет нарушить их целостность и, таким образом, привести к потере ценных данных [7]. Следовательно, обеспечение кибербезопасности путем дистанционного мониторинга является актуальной задачей.

Анализ средств и методов дистанционного мониторинга состояния SQL Server

Важным этапом при разработке метода мониторинга SQL Server является выбор инструмента, посредством которого будет производиться отслеживание ресурсов. Требуется выбрать такой инструмент, который будет отвечать требованиям безопасности. Злоумышленник извне не должен получить данные о состоянии SQL Server и тем более функции администрирования [8, 9, 10]. Инструмент должен обладать гибкостью настройки и содержать в себе возможность отслеживать любые ресурсы сервера, которые могут приводить к снижению производительности, в том числе и передаваемые инструкции пользователей. Результаты анализа сведены в Таблицу 1 по описанным критериям.

Инструмент	Гибкость	Безопасность	Информативность
Zabbix	+	+-	-
PRTG: WMI Sensor	+-	-	-
SQLNetRemoting	+	+	+

Таблица 1. Результат анализа инструментария для дистанционного мониторинга.

Выбор авторского клиент-серверного приложения SQLNetRemoting [11] обуславливается повышенной безопасностью и информативностью, так как клиентская часть может выступать в качестве административной панели для выполнения задачи дистанционного мониторинга посредством языка SQL.

Системные базы данных содержат в себе всю необходимую информацию о состоянии SQL Server, поэтому обращение к ним средствами языка запросов SQL

может обеспечить все необходимые знания о состоянии SQL Server [12]. Для разработки инструкций, которые будут внедрены в SQLNetRemoting необходимо проанализировать существующие объекты SQL Server, нацеленные на оценку состояния сервера и безопасность информационных процессов. Результаты анализа представлены в Таблице 2.

Тип объекта	Полнота	Сложность конструирования запросов	Возможность расширения функциональности
Системные динамические представления (view)	+	-	-
Системные функции	+	-	-
Activity Monitor	Отсутствует возможность удаленного применения		
Хранимые процедуры [13]	+	+	+

Таблица 2. Результат анализа объектов SQL Server.

Хранимые процедуры предоставляют сведения в виде таблиц и охватывают наибольшее количество информации о состоянии сервера, но помимо системных хранимых процедур существуют недокументированные хранимые процедуры, обладающие возможностью удобной выборки столбцов и предоставлением сведений,

обеспечивающих полноту мониторинга состояния сервера. В таблице 3 представлено обоснование выбора состава хранимых процедур, решающих задачи мониторинга.

Хранимая процедура	Интегральная информативность
Sp_who	-
Sp_spaceused	-
Sp_Monitor	-
Sp_who2	+
Sp_WhoIsActive	+

Таблица 3. Выбор хранимых процедур.

Выбранные хранимые процедуры возвращают больше сведений, чем документированные, но их необходимо фильтровать для повышения информативности о значимых контролируемых ресурсах, чтобы повысить оперативность предотвращения перегрузки сервера.

Для дальнейших исследований наиболее перспективной является авторская хранимая процедура Sp_WhoIsActive (http://whoisactive.com), которая поддерживается автором и динамически развивается, но она требует дополнительной интеграции в MS SQL Server, в отличие от предустановленной sp_who2.

Определение состава контролируемых ресурсов сервера SQL

Важным этапом разработки является определение значимых ресурсов, которые нужно контролировать для корректной фильтрации выбранных хранимых процедур.

Первым ресурсом, который стоит контролировать, являются блокирующие процессы на сервере. Опасность блокирующих процессов определяется тем, что слишком длительная блокировка может вызвать чрезмерную неплановую нагрузку на

сервер, к тому же блокировка какого-то процесса может привести к потере целостности данных [14, 15, 16].

Второй контролируемый ресурс – это все пользовательские процессы. Системными процессами управляет сам SQL Server, и они не могут вызвать неплановой перегрузки, а действия пользователей могут, поэтому стоит контролировать инструкции SQL, в особенности незавершенные транзакции [17, 18].

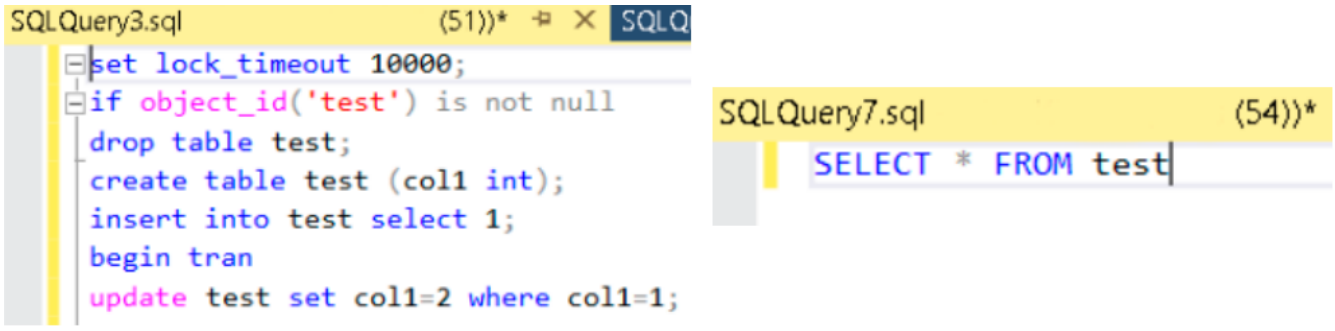
Следующими контролируемыми ресурсами являются ресурсы SQL Server. Перегрузка ресурсов SQL Server может вызвать прекращение журналирования транзакций, и, если понадобится восстановление данных из-за аварии на сервере, то это будет невозможно, так как они будут потеряны [19].

Помимо этого, нужно отслеживать информацию о подключении: логины пользователей; имя рабочей станции; имя клиентского приложения. Получив эту информацию, можно выявить процессы агентов SQL Server или пользователей, если подключение происходит в локальной сети. Если подключение происходит извне, то по трем указанным параметрам можно выявить, что к серверу был подключен нежелательный пользователь [20].

Структура клиентской и серверной части информационной системы, используемой для тестирования средств дистанционного мониторинга

Для того, чтобы убедиться в достоверности возвращаемых данных в созданных инструкциях, необходимо создать тестовый инцидент, который имитирует работу пользователей на сервере и создает блокирующий процесс. Таким инцидентом может

выступать одновременное чтение таблицы с процессом, в котором происходят изменения в этой же таблице. (Рисунок 1)



а) *б)*

Рис. 1. а) Создание транзакции изменения данных, б) создание блокировки.

Как следует из обоснования выбора хранимых процедур, необходимо фильтровать получаемые данные. Для этого требуется создание временных таблиц, в которые помещаются необходимые столбцы в соответствии с выбранными контролируемыми ресурсами. Все созданные инструкции должны быть расположены, согласно со структурой клиент-серверного приложения, в файле высокоуровневого протокола SQLXMLMaps серверной части приложения, который обеспечивает кодирование информации [11].

Первая разработанная SQL-инструкция основана на хранимой процедуре sp_who2, фильтрация выполнена по принципу существующих блокирующих процессов и отсортирована по ресурсоемкости. Отметим, что в отличие от оригинальных запросов используются смысловые названия на русском языке, что обеспечивает наибольшее понимания для русскоязычных пользователей.

Тестирование запроса клиента № 1: «Получить сведения о блокирующих процессах на сервере» представлено на Рисунке 2.

Датаивремя	IDсеанса	Состояниепроцесса	Логин	Имяхоста	IDблокпроцесса	ИмяБД	ИнструкцияTSQL	Процессорноевремя
06/03 12:52:48	54	В ожидании	sa	INUCI5-1	51	testBase	SELECT	0

Времячтенияилизаписидиска	Названиепрограммы	Числоявныхтранзакций
0	Среда Microsoft SQL Server Management Studio - запрос	

Рис. 2. Результат тестирования запроса клиента № 1.

Вторая SQL-инструкция отфильтрована по принципу отображения только пользовательских процессов на сервере из хранимой процедуры sp_who2, для этого задан индикатор для нахождения идентификатора сеанса больше 50-го, так как до этого значения выполняются системные процессы SQL Server, которые не нагружают сервер. Также выполнена сортировка по ресурсоемкости.

Тестирование запроса клиента № 2: «Получить сведения о процессах, выполняемых пользователями на сервере» представлено на Рисунке 3.

IDсеанса	Состояниепроцесса	Логин	Имяхоста	IDблокпроцесса	ИмяБД	ИнструкцияTSQL
52	Спящий режим	sa	INUCI5-1	.	master	AWAITING COMMAND
51	Спящий режим	sa	INUCI5-1	.	master	AWAITING COMMAND
54	Спящий режим	sa	INUCI5-1	.	testBase	AWAITING COMMAND
53	Готов к запуску	sa	INUCI5-1	.	NewServerTest	SELECT INTO
55	Спящий режим	sa	INUCI5-1	.	testBase	AWAITING COMMAND
56	Спящий режим	sa	INUCI5-1	.	testBase	AWAITING COMMAND

Процессорноевремя	Времячтенияилизаписидиска	Датаивремя	Названиепрограммы
31	164	06/03 12:10:51	Microsoft SQL Server Management Studio
1141	19	06/03 12:36:11	Среда Microsoft SQL Server Management Studio - запрос
0	2	06/03 12:11:51	Среда Microsoft SQL Server Management Studio - запрос
16	0	06/03 12:41:49	.Net SqlClient Data Provider
0	0	06/03 12:12:54	Среда Microsoft SQL Server Management Studio - запрос
0	0	06/03 12:12:43	Среда Microsoft SQL Server Management Studio - запрос

Рис. 3. Результат тестирования запроса клиента № 2.

В основе третьей SQL-инструкции используется хранимая процедура sp_WhoIsActive, фильтрация настроена на вывод пользовательских инструкций,

переданных SQL-серверу, а именно — незавершенные транзакции, как было описано в определении состава контролируемых ресурсов, такие транзакции могут быть потеряны при аварии на сервере.

Тестирование запроса клиента № 3 «Просмотр выполняемых SQL инструкций на сервере» представлено на Рисунке 4.

Датаивремя	IDceанса	TSQLинструкция	Логин	Имяхоста
00 00:31:19.003	56	<?query -- BEGIN TRAN SELECT * FROM STUDENTS --?>	sa	INUCI5-1

Рис. 4. Результат тестирования запроса клиента № 3.

Четвертая инструкция отображает переданные инструкции в заблокированных процессах. Для этого была произведена фильтрация в столбце блокирующего процесса «IS NOT NULL», что означает: вывод информации возможен только в случае, если блокирующий процесс существует.

Практическая ценность данной инструкции заключена в том, что, если сеанс заблокирован, то скорее всего, он обращен к таблице, в которой производятся какие-либо изменения.

Тестирование запроса клиента № 4 «Просмотр выполняемых SQL-инструкций в заблокированных сеансах» представлено на Рисунке 5.

Датаивремя	IDceанса	TSQLинструкция	Логин	Имяхоста	IDблокпроцесса	Процессорноевремя
00 00:03:15.896	54	<?query -- select * from test; --?>	sa	INUCI5-1	51	0

Рис. 5. Результат тестирования запроса клиента № 4.

Пятая SQL-инструкция, основанная на хранимой процедуре sp_WhoIsActive, отфильтрована для отображения количества заблокированных процессов в текущем сеансе, для этого был добавлен параметр @find_block_leaders и установлен индикатор заблокированных процессов «IS NOT NULL». Большое количество блокировок может привести к перегрузке сервера.

Тестирование запроса клиента № 5 «Получить сведения о количестве заблокированных процессов» представлено на Рисунке 6.

Датаивремя	IDсеанса	TSQLинструкция	Логин	Имяхоста	IDблокпроцесса	Процессорноевремя	Количествозаблокированныхпроцессов
00 00:25:26.143	54	<?query -- select * from test; --?>	sa	INUCI5-1	51	0	1

Рис. 6. Результат тестирования запроса клиента № 5.

Шестая инструкция SQL возвращает сведения о расходе ресурсов в реальном режиме транзакцией. И добавлен параметр @delta_interval, равный 10 секундам. Это значит, что будет подсчитано, сколько ресурсов сервер затратил за 10 секунд времени.

Практическая ценность данной инструкции заключается в оценке затрат SQL Server на выполнение операции в определенном сеансе на данный момент, а не за всё время. Проблема состоит в том, что бывают операции, которые активны и продолжают потреблять ресурсы сервера, а во время выполнения хранимой процедуры отобразились сведения только за продолжительность всего сеанса.

Тестирование запроса клиента № 6 «Получить сведения о расходе ресурсов в реальном времени» представлено на Рисунке 7.

Времязапускарежима	TSQLинструкция	IDсеанса	Логин	времяCPUза10сек	времяCPUзавесьсеанс	Потребляемаяпамятьза10сек	Потребляемаяпамятьзавесьсеанс
2021-06-03T12:52:46.643+03:00	<?query -- set lock_timeout 10000; if object_id ('test') is not null drop table test; create table test (col1 int); insert into test select 1; begin tran update test set col1=2 where col1=1; --?>	51	sa	0	0	0	3
2021-06-03T12:53:05.52+03:00	<?query -- select * from test; --?>	54	sa	0	0	0	4

Рис. 7. Результат тестирования запроса клиента № 6.

Заключение

Анализ существующих методов дистанционного мониторинга показал, что для того, чтобы обеспечить полноту дистанционного мониторинга, необходимо использовать средства языка запросов SQL.

Предложенный метод дистанционного мониторинга решает проблему безопасности передачи информации в открытой среде путём использования инструментария, обоснованного в анализе средств и методов, а проблему полноты дистанционного мониторинга - путём использования системных хранимых процедур sp_who2 и sp_whoisactive.

Показано, что наибольшими возможностями обладает sp_whoisactive, которая динамически развивается с целью повышения информативности задач мониторинга состояния SQL Server.

Однако, анализ показал, что выбранные хранимые процедуры содержат в себе много сведений. Поэтому для решения поставленной задачи эффективного оценивания состояния ИУС была выполнена модификация путём фильтрации данных выбранных хранимых процедур с целью устранения их избыточности.

Кроме того, оптимизированные административные запросы сокращают объем сведений о состоянии ИУС, обеспечивая безопасность в открытой среде передачи и повышая оперативность выполнения запроса, при этом достигается полнота выбранных контролируемых ресурсов.

Список источников

1. Груммет В.А., Лисовин О.А., Тюнин Е.Б. Способы защиты информационных систем // III всероссийская научно-практическая конференция «Цифровизация экономики: направления, методы, инструменты» (Краснодар, 18–23 января 2021): сборник материалов. – Краснодар: Изд-во Кубанский государственный аграрный университет имени И.Т. Трубилина, 2021. С. 58-61.

2. Титов А.Г., Неретин Е.С., Дудкин С.О., Брусникин П.М. Разработка архитектуры бортового сервера данных для применения в составе комплекса радиоэлектронного оборудования с применением концепции интегрированной модульной авионики // Труды МАИ. 2019. № 105. URL: https://trudymai.ru/published.php?ID=104257

3. Дудаков Н.С., Макаров К.В., Тимошенко А.В. Методика проектирования баз данных для автоматизированных систем управления специального назначения // Труды МАИ. 2016. № 90. URL: https://trudymai.ru/published.php?ID=74844

4. Соломатин М.С., Митрофанов Д.В. Модель интеллектуального детектора системы защиты автоматизированной системы управления // Труды МАИ. 2020. № 110. URL: https://trudymai.ru/published.php?ID=112926. DOI: 10.34759/trd-2020-110-16

5. Короткова Т.И. Многокритериальный алгоритм принятия решения в системе обеспечения информационной безопасности объектов гражданской авиации // Труды МАИ. 2015. № 84. URL: https://trudymai.ru/published.php?ID=63279

6. Медведев Ю.С. К вопросу об увеличении производительности базы данных Oracle // Международная научно-практическая конференция «Экономическое

развитие: состояние, проблемы, перспективы» (Пенза, 28-29 июня 2018): сборник статей. - Пенза, Поволжский дом знаний, 2018. С. 71-75.

7. Алимжанова Ж.М., Балтабекова А.Б., Турдалы А.Д. Защита и безопасность базы данных // Актуальные научные исследования в современном мире. 2019. № 12-4 (56). С. 61-64.

8. Клепцов М.Я., Любимова Л.В., Миронов М.М. Анализ угроз и уязвимостей СУБД Oracle // Вопросы кибербезопасности. 2018. № 2 (26). С. 16-23. DOI: 10.21681/2311-3456-2018-2-16-23

9. Казарян К.К., Белан В.В. Подделка запросов на стороне сервера // StudNet. 2022. Т. 5. №. 1. URL: https://stud.net.ru/poddelka-zaprosov-na-storone-servera/

10. Yunus M.A. et al. Review of SQL injection: Problems and prevention // International Journal on Informatics Visualization, 2018, vol. 2, no. 3-2, pp. 215-219. DOI:10.30630/joiv.2.3-2.144

11. Михайлов В.Ю., Мазепа Р.Б. Практикум последовательности дисциплин в форме проектирования системы защищенного информационного взаимодействия в открытых сетях // Информационное противодействие угрозам терроризма. 2015. Т. 2. № 25 (25). С. 161-169.

12. Виноградский В.Г., Егоров В.Г., Егорова О.М. Основной инструментарий MS SQL management Studio для работы с данными из внешних источников // Заметки ученого. 2021. № 13. С. 36-42.

13. Рыжикова Е.Г. Использование хранимых процедур в комплексах проблемно-ориентированных программ // V Международная научно-практическая конференция

«Актуальные научные исследования» (Пенза, 25 апреля 2022): сборник статей. - Пенза, Изд-во Наука и Просвещение, 2022. С. 39-41.

14. Okardi B., Asagba O. Overview of distributed database system // International Journal of Computer Techniques, 2021, vol. 8, issue 1, pp. 83-100.

15. Al-Hussaini K., Al-Amdi N.., Abdulrazzak F. A New Multi-resource Deadlock Detection Algorithm Using Directed Graph Requests in Distributed Database Systems // International Conference of Reliable Information and Communication Technology, Springer, Cham, 2020, pp. 462-474.

16. Ленкин А.В. Управление ресурсами и блокированием в централизованных базах данных // Постулат. 2020. № 6 (56). URL: http://e-postulat.ru/index.php/Postulat/article/view/3240/3287

17. Танаев И.В, Швейкин В.В., Завгородний С.Д., Дмитриев Е.А. Обеспечение согласованности данных в СУБД при помощи транзакций // IV Международная студенческая научно-практическая конференция «Научные исследования и разработки студентов» (Чебоксары, 29 июня 2017): сборник материалов. – Чебоксары: Изд-во Интерактив Плюс, 2017. С. 186-189.

18. Третьяков И.А., Кожекина Е.Н., Журавлев И.В. Оптимизация SQL-запросов // Вестник Донецкого национального университета. Серия Г: Технические науки. 2021. № 2. С. 39-49.

19. Тетенькин А.Ю. Проблемы параллельного выполнения транзакций в базах данных // Материалы XVII Международной научно-практической конференции «Татищевские чтения: актуальные проблемы науки и практики» (Тольятти, 24–25

апреля 2020). – Тольятти: Изд-во Волжский университет имени В.Н. Татищева, 2020. С. 16-19.

20. Амосова А.Ф. Функции безопасности системы управления базами данных Microsoft SQL server 2017 // XLVII научная и учебно-методическая конференция Университета ИТМО «Альманах научных работ молодых ученых Университета ИТМО» (Санкт-Петербург, 31 января – 03 февраля 2018). – СПб: Изд-во ИТМО, 2018. С. 37-40.

References

1. Grummet V.A., Lisovin O.A., Tyunin E.B. *III vserossiiskaya nauchno-prakticheskaya konferentsiya «Tsifrovizatsiya ekonomiki: napravleniya, metody, instrumenty»*: sbornik materialov. Krasnodar, Izd-vo Kubanskii gosudarstvennyi agrarnyi universitet imeni I.T. Trubilina, 2021, pp. 58-61.

2. Titov A.G., Neretin E.S., Dudkin S.O., Brusnikin P.M. *Trudy MAI,* 2019, no. 105. URL: https://trudymai.ru/eng/published.php?ID=104257

3. Dudakov N.S., Makarov K.V., Timoshenko A.V. *Trudy MAI,* 2016, no. 90. URL: https://trudymai.ru/eng/published.php?ID=74844

4. Solomatin M.S., Mitrofanov D.V. *Trudy MAI,* 2020, no. 110. URL: https://trudymai.ru/eng/published.php?ID=112926. DOI: 10.34759/trd-2020-110-16

5. Korotkova T.I. *Trudy MAI,* 2015, no. 84. URL: https://trudymai.ru/eng/published.php?ID=63279

6. Medvedev Yu.S. *Mezhdunarodnaya nauchno-prakticheskaya konferentsiya «Ekonomicheskoe razvitie: sostoyanie, problemy, perspektivy»*: sbornik statei. Penza, Povolzhskii dom znanii, 2018, pp. 71-75.

7. Alimzhanova Zh.M., Baltabekova A.B., Turdaly A.D. *Aktual'nye nauchnye issledovaniya v sovremennom mire,* 2019, no. 12-4 (56), pp. 61-64.

8. Kleptsov M.Ya., Lyubimova L.V., Mironov M.M. *Voprosy kiberbezopasnosti,* 2018, no. 2 (26), pp. 16-23. DOI: 10.21681/2311-3456-2018-2-16-23

9. Kazaryan K.K., Belan V.V. *Student,* 2022, vol. 5, no. 1. URL: https://stud.net.ru/poddelka-zaprosov-na-storone-servera/

10. Yunus M.A. et al. Review of SQL injection: Problems and prevention // International Journal on Informatics Visualization, 2018, vol. 2, no. 3-2, pp. 215-219. DOI:10.30630/joiv.2.3-2.144

11. Mikhailov V.Yu., Mazepa R.B. *Informatsionnoe protivodeistvie ugrozam terrorizma,* 2015, vol. 2, no. 25, pp. 161-169.

12. Vinogradskii V.G., Egorov V.G., Egorova O.M. *Zametki uchenogo,* 2021, no. 13, pp. 36-42.

13. Ryzhikova E.G. *V Mezhdunarodnaya nauchno-prakticheskaya konferentsiya «Aktual'nye nauchnye issledovaniya»*: sbornik statei. Penza, Izd-vo Nauka i Prosveshchenie, 2022, pp. 39-41.

14. Okardi B., Asagba O. Overview of distributed database system, *International Journal of Computer Techniques*, 2021, vol. 8, issue 1, pp. 83-100.

15. Al-Hussaini K., Al-Amdi N.., Abdulrazzak F. A New Multi-resource Deadlock Detection Algorithm Using Directed Graph Requests in Distributed Database Systems,

International Conference of Reliable Information and Communication Technology, Springer, Cham, 2020, pp. 462-474.

16. Lenkin A.V. *Postulat,* 2020, no. 6 (56). URL: http://e-postulat.ru/index.php/Postulat/article/view/3240/3287

17. Tanaev I.V, Shveikin V.V., Zavgorodnii S.D., Dmitriev E.A. *IV Mezhdunarodnaya studencheskaya nauchno-prakticheskaya konferentsiya «Nauchnye issledovaniya i razrabotki studentov»*: sbornik materialov, Cheboksary, Izd-vo Interaktiv Plyus, 2017, pp. 186-189.

18. Tret'yakov I.A., Kozhekina E.N., Zhuravlev I.V. *Vestnik Donetskogo natsional'nogo universiteta. Seriya G: Tekhnicheskie nauki,* 2021, no. 2, pp. 39-49.

19. Teten'kin A.Yu. *Materialy XVII Mezhdunarodnoi nauchno-prakticheskoi konferentsii «Tatishchevskie chteniya: aktual'nye problemy nauki i praktiki»,* Tol'yatti, Izd-vo Volzhskii universitet imeni V.N. Tatishcheva, 2020, pp. 16-19.

20. Amosova A.F. *XLVII nauchnaya i uchebno-metodicheskaya konferentsiya Universiteta ITMO «Al'manakh nauchnykh rabot molodykh uchenykh Universiteta ITMO»*, Saint Petersburg, Izd-vo ITMO, 2018, pp. 37-40.

Статья поступила в редакцию 24.05.2022
Статья после доработки 30.05.2022
Одобрена после рецензирования 10.08.2022
Принята к публикации 26.12.2022
The article was submitted on 24.05.2022; approved after reviewing on 10.08.2022; accepted for publication on 26.12.2022