УДК 33 **DOI: 10.34670/AR.2022.71.72.039**

Механизмы обеспечения безопасности банковских АС при использовании Apache Ignite

Сенигова Александра Дмитриевна

Бакалавр
Финансовый университет при Правительстве Российской Федерации,
125993, Российская Федерация, Москва, Ленинградский пр., 49;
e-mail: asenigova@gmail.com

Ларионова Светлана Львовна

Кандидат технических наук,
доцент департамента «Информационная безопасность»,
Финансовый университет при Правительстве Российской Федерации,
125993, Российская Федерация, Москва, Ленинградский пр., 49;
e-mail: sllarionova@fa.ru

Аннотация

В этой статье подробно рассматриваются требования к использованию Apache Ignite в банковских автоматизированных системах и определяются необходимые механизмы кибербезопасности и параметры их конфигурирования при использовании Apache Ignite в банковских АС. Apache Ignite – это распределенная база данных с открытым исходным кодом, платформа для кэширования и обработки, предназначенная для хранения и вычисления больших объемов данных. Во многих крупных банках программный продукт стал «основой ИТ-ландшафта» для организации хранилищ данных и обработки больших объемов данных онлайн в параллельных процессах. При всех преимуществах, главным недостатком данного продукта является отсутствие изначально сконфигурированных параметров обеспечения информационной безопасности. Это приводит к невозможности эффективного мониторинга систем, построенных на базе продукта, и обеспечения безопасности. В рамках работы был проведен анализ продукта на соответствие требованиям законодательства РФ в сфере информационной безопасности банковских автоматизированных систем и определен перечень уязвимостей, возникающих при использовании продукта в банковских АС. В статье представлен перечень необходимых параметров конфигурирования кластера Apache Ignite и плагина безопасности, и список требований к конфигурированию безопасности на стороне инфраструктуры с учетом требований законодательства.

Для цитирования в научных исследованиях

Сенигова А.Д., Ларионова С.Л. Механизмы обеспечения безопасности банковских АС при использовании Apache Ignite // Экономика: вчера, сегодня, завтра. 2022. Том 12. № 5А. С. 432-442. DOI: 10.34670/AR.2022.71.72.039

Ключевые слова:
In-Memory Data Grid, Apache Ignite, кибербезопасность, банковские автоматизированные системы, уязвимости.

Введение

Для обеспечения информационной безопасности организаций кредитно-финансовой сферы необходимо обеспечить реализацию информационной безопасности в части:

– безопасности инфраструктуры;
– безопасности прикладного программного обеспечения;
– безопасности технологий обработки данных;
– мониторинга и протоколирования действий и операций;
– обеспечения информационной безопасности на всех этапах жизненного цикла автоматизированной системы.

Реализация данных принципов при применении программных продуктов при разработки и эксплуатации автоматизированных банковских систем строится посредством следующих подходов:

– соблюдение требований зафиксированных в нормативных документах и требованиях регуляторов;
– устранение уязвимостей в программном обеспечении;
– обеспечение целостности и подлинности данных при информационном обмене;
– протоколирование действий и операций для осуществления надзорной деятельности.

Программный продукт Apache Ignite в своей версии «из коробки» не соответствует требованиям информационной безопасности, предъявляемым к банковской автоматизированной системе (ГОСТ Р 57580.1-2017, Положения Банка России 719-П и 1119-П, приказ ФСТЭК №21).

Использование программного продукта Apache Ignite в банковских автоматизированных системах приводит к возникновению ряда уязвимостей, а именно:

– отсутствие системы аутентификации и хранения пользовательских сессий;
– отсутствие настроенных механизмов контроля доступа (реализованной авторизации) и ролевой модели;
– отсутствие управления пользовательскими сессиями и хранения идентификаторов сессий;
– отсутствие журналирования и мониторинга пользовательских и системных действий (аудит событий безопасности);
– отсутствие настроенных механизмов шифрования;
– отсутствие входного фильтра запросов (механизм противодействия инъекционным атакам, в частности SQL-инъекциям).

Коммерческая реализация плагина безопасности GridGain не позволяет закрыть все уязвимости программного продукта Apache Ignite, так как не обладает большим набором возможностей обеспечения безопасности и в части реализованных изменений сам по себе приводит к возникновению новых уязвимостей, а именно:

– незащищенное хранение паролей к учетным записям кластера и хранилищам сертификатов;
– отсутствие средства проверки согласованности, целостности и непротиворечивости

конфигураций разных узлов, средства контроля целостности и подлинности конфигурационных файлов;

– невозможно применение настроек без прерывания работоспособности кластера.

Для грамотного применения программного продукта необходимо реализовать дополнительный функционал, который будет обеспечивать безопасность автоматизированной системы.

Требования к плагину безопасности

Опираясь на результаты анализа соответствия программного продукта требованиям законодательства к обеспечению безопасности банковских автоматизированных систем и анализ уязвимостей, возникающих при использовании данного продукта в банковских автоматизированных системах, необходимый функционал, которым необходимо дополнить на стороне плагина безопасности. Полный перечень требований, который необходимо реализовать на стороне плагина безопасности Apache Ignite представлен в таблице 1.

Таблица 1 – Требования к реализации плагина безопасности Apache Ignite

Группа требований	Требования
Аутентификация и авторизация	Должна быть реализована двухфакторная аутентификация, включающая, например: проверка логина и пароля пользователя; проверка (авторизация) сертификата.
	Должна осуществляться однозначная идентификация каждого компонента при любом взаимодействии с кластером Apache Ignite.
Межсервисная аутентификация и авторизация	Аутентификация при межсервисных взаимодействиях должна проводиться с использованием протокола TLS версии не ниже 1.2.
Пароли	Используемые пароли должны соответствовать парольной политике организации.
	Должно обеспечиваться защищенное хранения паролей.
Аудит	Должен осуществляться аудит всех попыток аутентификации и авторизации (как успешных, так и не успешных) с обязательной фиксацией атрибутов учетной записи и CN (общее имя) сертификата.
	Должен осуществляться аудит всех попыток операций с объектами и субъектами кластера Apache Ignite, осуществляемых администраторами кластера (для любого протокола взаимодействия, посредством которого осуществляются любые действия администраторов кластера).
	Должна присутствовать возможность доступа к событиям из средств внешнего мониторинга.
	Должна отсутствовать техническая возможность отключения фиксации событий аудита.
	События должны храниться централизовано вне аудируемой системы.
	Все действия администраторов должны быть запротоколированы.
Журналирование	Необходимо реализовать журналирование и мониторинг пользовательских и системных действий.

Группа требований	Требования
Управление сессиями	Должен быть реализован функционал управления пользовательскими сессиями.
	Должен быть реализован функционал, обеспечивающий завершение активной сессии пользователя после смены пароля.
	Должно контролироваться наличие не более N одновременных сессий одного пользователя, где N – настраиваемый параметр. Для персонализированной учетной записи N=1, для коллективной – в зависимости от потребностей клиента, критичности данных, ресурсов и конфигурации сервера.
Клиентские подключения	При проведении критичной операции внутри АС/компонентов АС в ручном режиме должна быть исключена возможность единоличного проведения данной операции (принцип «two persons»)
	Необходима повторная аутентификация пользователей с ролью «Администратор АС» или подтверждение действия при выполнении критичных операций или изменении критичных параметров АС/компонентов АС
Разграничение доступа (ролевая модель)	Должна быть разработана и применена ролевая модель со следующими атрибутами, а так же матрица несовместимости ролей/полномочий.
	Должны быть предусмотрена реализация следующие обязательные роли: администратор АС/компонента АС; администратор ИБ АС/компонента АС; клиентская роль.
	Роли не должны быть связаны (должны быть исключены операции наследования и агрегирования прав).
Фильтрация входящих запросов	Должна обеспечиваться фильтрация входящих запросов на предмет инъекционных атак.

Реализация данного набора функциональных возможностей позволит закрыть большую часть существующих уязвимостей, которые привносит в автоматизированную систему программный продукт Apache Ignite. Помимо реализации данного набора требований, необходимо так же правильно сконфигурировать механизмы безопасности на стороне информационной инфраструктуры и остальных компонентов банковской автоматизированной системы.

Описание плагина безопасности

Рассмотрим пример реализации плагина безопасности Apache Ignite, который будет соответствовать требованиям информационной безопасности.

Узел Ignite выполняет следующие проверки безопасности:

- аутентификация подключающегося субъекта безопасности.
- авторизация разрешения (поддерживаемые разрешения безопасности системе, указываются на уровне кэша, задачи или службы) перед выполнением какой-либо операции (например, добавление данных в кэш).

Вместо реальных проверок Apache Ignite содержит только реализации-заглушки, но позволяет сконфигурировать процессор, реализованный в стороннем плагине, которому

передаются все операции аутентификации и авторизации.

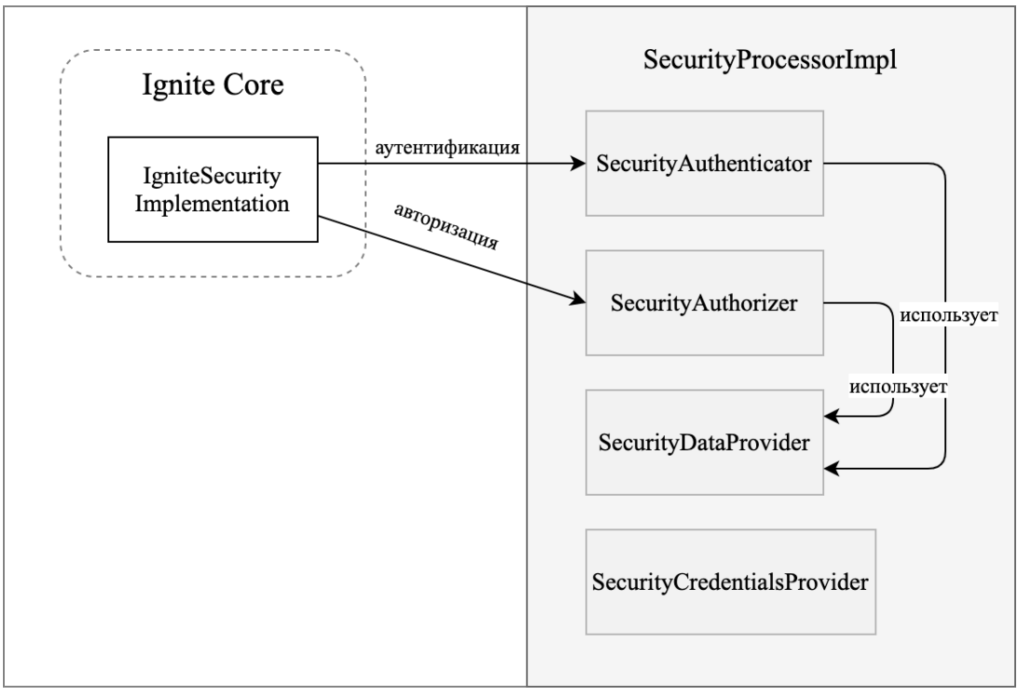

Рисунок 1 – Структура процессора в плагине безопасности Apache Ignite

Реализация процессора в плагине (SecurityProcessorImpl) агрегирует следующие сущности:

1) SecurityAuthenticator – аутентифицирует пользователя в широком смысле. В качестве пользователя может выступать другой узел Ignite либо пользователь, подключившийся через клиентское соединение;
2) SecurityAuthorizer – авторизует действия пользователя;
3) SecurityDataProvider – хранит и позволяет считывать данные о пользователях:

– логин и пароль – для аутентификации;

– разрешения – для авторизации операций.

4) SecurityCredentialsProvider – предоставляет логин и пароль пользователя, от имени которого запущен текущий узел Ignite, чтобы другие узлы кластера могли аутентифицировать и авторизовать попытку входа этого узла в кластер.

На практике удобно управлять правами пользователей не на уровне отдельных разрешений, а на уровне ролей, представляющих собой набор разрешений.

Для хранения данных о пользователях и ролях реализация использует внутреннее распределенное хранилище (distributed metastorage), данные в котором синхронизированы между узлами в кластере и недоступны внешним пользователям через общедоступный API. Записи плагина безопасности в distributed metastorage бывают двух видов:

– данные пользователя: логин, хэш пароля, список имен ролей;

– данные роли: имя роли, набор разрешений в системе.

Для аутентификации используются только записи первого вида, для авторизации – обоих видов (чтобы получить полный набор разрешений пользователя). Помимо обычной проверки логина и пароля пользователя выполняется проверку TLS-сертификата, предоставленного пользователем при соединении с кластером.

В качестве фреймворка для журналирования используются библиотеки JUL (java.util.logging), Log4j или Log4j2, JCL, SLF4J.

Перечень событий для журналирования:

– запрос на активацию кластера;

– транзакция отменена по истечении времени;

– транзакция заблокирована;

– кластер успешно активирован;

– ошибка активации кластера;

– узел становится координирующим узлом;

– начало ребалансировки (перераспределение данных/объектов между узлами кластера);

– завершение ребалансировки;

– ошибка ребалансировки.

Плагин отслеживает события, относящиеся к информационной безопасности, и отправляет их в централизованный сервис хранения событий аудита.

Полный перечень событий, передаваемых в централизованную систему аудита, включая события информационной безопасности, представлен в таблице 2.

Таблица 2 – Полный перечень событий, передаваемых в централизованную систему аудита

Субъект/объект	Операция
Аутентификация	Аутентификация выполнена успешно
	Аутентификация неудачна
Узел	Подключение узла в топологию
	Отключение узла от топологии штатными средствами
	Ошибка подключения узла
	Событие серверного узла, когда он «понимает», что находится вне топологии
	Клиентский узел отключился от серверного узла
	Клиентский узел переподключился к серверному узлу
Создание кэша	Операция создания кэша
	Операция удаления кэша
Управляющие операции с кэшем	Все операции, проводимые по протоколу Binary Rest (с помощью утилит управления Visor, Shell-утилиты)
Авторизация	Успешная
	Неудачная
Запрос (SQL)	Исполнение
Сертификат (клиентский/серверный)	Истек срок действия
Резервное копирование	Создание резервной копии начато
	Создание резервной копии окончено
	Создание резервной копии не удалось

Необходимые параметры конфигурирования кластера Apache Ignite и дополнительные требования:

1) Для исключения возможности несанкционированного доступа к кластеру Apache Ignite необходимо заблокировать, удалить все ненужные или неиспользуемые учетные записи в кластере и ОС на серверах с установленным кластером Apache Ignite.
2) Требуется сменить пароли УЗ, установленные по умолчанию на постоянные, сложность которых соответствует требованиям (для исключения возможности несанкционированного доступа к кластеру Apache Ignite).
3) Создать дополнительные персонализированные УЗ администраторов эксплуатации (для контроля над учетными записями лиц, ответственных за сопровождение кластера, инфраструктуры кластера и механизмов безопасности кластера для исключения ситуаций злоупотребления правами доступа и реализации угроз безопасности информации).
4) Передача информации из кластера Apache Ignite по открытым общедоступным сетям не допускается (для исключения возможности несанкционированного доступа к информации).
5) Для исключения возможности несанкционированного доступа к кластеру Apache Ignite требуется установить и настроить двухфакторную аутентификацию с использованием встроенной системы безопасности плагина безопасности кластера (логин/пароль + сертификат, данная реализация является классической и поддерживается большинством автоматизированных банковских систем и систем потребителей).
6) Сертификаты, используемые для подключения, должны быть защищены паролем (client key store password, trust store password; для исключения возможности компрометации аутентификационных данных и несанкционированного доступа к кластеру).
7) При конфигурировании второго фактора аутентификации должны использоваться официальные сертификаты, подписанные центрами сертификации организации, использование самоподписанных сертификатов не допускается (для исключения возможности несанкционированного доступа к кластеру Apache Ignite).
8) Параметр authenticationEnable (включение аутентификации) должен быть установлен в значение «true». (Данный параметр необходим для дальнейшей настройки конфигурации кластера. Первый запускаемый узел, должен иметь включенную аутентификацию. При запуске создается учетная запись пользователя с именем «ignite» и паролем «ignite». Эта учетная запись предназначена для создания других учетных записей пользователей в соответствии с конкретными потребностями потребителя. В последствии данная учетная запись подлежит удалению).
9) Для обеспечения защищенного удаленного административного доступа необходимо использовать шифрование соединений с использованием криптографии (для исключения возможности несанкционированного доступа к кластеру Apache Ignite).
10) Должен использоваться протокол шифрования TLS версии не ниже 1.2 (для исключения возможности перехвата, модификации данных при передачи их по внутренним и внешним по отношению к кластеру информационным потокам).
11) Должна осуществляться однозначная идентификация каждого компонента при любом взаимодействии с кластером Apache Ignite (серверные/клиентские узлы кластера, тонкие/толстые клиенты, утилиты управления кластером, внешние базы данных).
12) Необходимо настроить механизмы аутентификации для подключения узлов кластера и для подключения тонких клиентов для исключения возможности несанкционированного доступа к кластеру Apache Ignite (TCP Discovery SPI, TCP Communication SPI, Thin client,

JDBC, ODBC, JMX, Rest). Протоколы тонких подключений конфигурируются в зависимости от необходимости конкретного ПО, неиспользуемые явным образом отключаются .

13) Права доступа к конфигурационным файлам кластера Apache Ignite должны соответствовать следующей таблице 3 (для исключения возможности несанкционированного доступа к конфигурационным файлам кластера Apache Ignite).

Таблица 3– Права доступа к конфигурационным файлам кластера

Учетная запись	Чтение	Запись
Административная УЗ	+	+
ТУЗ, от имени которой работают службы кластера	+	–
Пользовательские УЗ	–	–

1) Недопустимо отображения в записях журналирования конфиденциальной информации (исключение возможности несанкционированного доступа к конфиденциальной инфомрации, обрабатываемой в кластере Apache Ignite).
2) На серверных узлах кластера Apache Ignite, не должно находиться сторонних компонентов, для исключения возможности компрометации узла через интерфейсы сервисов.
3) На клиентских узлах кластера Apache Ignite должны находиться только высокодоверенные модули (технологические и прикладные сервисы). Все остальные прикладные модули и компоненты должны располагаться на узлах, на которых нет никаких компонентов Apache Ignite и которые не могут взаимодействовать с кластером при помощи любых протоколов.
4) За каждым приложением клиентского узла должен быть закреплен определенный перечень данных (кэшей), в которые он может вносить изменения. Остальные данные должны быть доступны только на чтение или недоступны совсем (для исключения возможности несанкционированного доступа к данным кластера Apache Ignite).
5) Каждое приложение в рамках кластера Apache Ignite должно иметь отдельную технологическую учетную запись с правами доступа к массивам данных. Конфигурирование кластера Apache Ignite должно реализовывать механизм «черных и белых листов» (ACL, access list) разграничения доступа потребителей к кэшам (для исключения возможности несанкционированного доступа к данным кластера Apache Ignite).
6) Разграничение доступа к массивам данных должно осуществляться по атрибутам субъекта (модуля на клиентском узле) и атрибутам объекта (массива данных), а также типу операции.
7) Режим управления по протоколу JMX должен быть запрещен в явном виде (для исключения возможности несанкционированного использования управляющего протокола для управления кластером Apache Ignite).
8) Для каждого кластера должны быть сконфигурированы параметры безопасности.
9) Настройки безопасности (реализация пунктов 1-21 данного перечня) не должны быть доступны для отключения.
10) Периметр кластера Apache Ignite должен быть защищен сетевыми средствами защиты информации, включая межсетевые экраны, средства противодействия вторжениям,

сетевые сканеры безопасности (защита на инфраструктурном уровне).

11) Должно быть обеспечено использование и регулярное обновление антивирусного программного обеспечения на ОС серверов с установленным кластером Apache Ignite (защита на инфраструктурном уровне).

При правильной реализации механизмов обеспечения информационной безопасности в плагине безопасности и на стороне инфраструктуры можно свести к минимуму вероятность возникновения уязвимостей при применении в банковских автоматизированных системах программного продукта Apache Ignite. В этом случае применение функциональных возможностей позволит упростить выполнение многих задач, связанных с обработкой огромных массивов данных в режиме реального времени, и при этом не приведет к реализации угроз информационной безопасности.

Заключение

Внедрение предложенной реализации плагина безопасности позволит снизить риски информационной безопасности – возможность того, что угроза сможет воспользоваться уязвимостью актива или группы активов и тем самым нанести ущерб организации. С экономической точки зрения мера защиты оправдана, если эффект от ее применения, выраженный через уменьшение ожидаемого экономического ущерба, превышает затраты на ее реализацию. Внедрение плагина безопасности Apache Ignite уменьшит частоту появления риска в 10 раз при неизменном размере ущерба. Эффект защиты будет составлять 90% риска, а сам риск уменьшится в 10 раз.

Практическая значимость исследования заключается в возможности использования разработанных и обоснованных предложений в практической деятельности кредитных организаций при использовании систем класса IMDG в автоматизированных системах.

Коммерческая реализация плагина безопасности GridGain не позволяет закрыть все уязвимости программного продукта Apache Ignite, так как не обладает большим набором возможностей обеспечения безопасности и в части реализованных изменений сам по себе приводит в возникновению новых уязвимостей. Именно поэтому существует два сценария «безопасного» использования Apache Ignite: первый - на базе коммерческой реализации плагина GridGain и создания дополнительного плагина безопасности, который будет устанавливаться рядом с GridGain, поднимать уровень информационный безопасности и предоставлять инструменты работы с основой – Apache Ignite; второй - использование Apache Ignite с самостоятельно реализованным плагином безопасности (если не требуется использование функций коммерческого решения GridGain, не связанных с безопасностью).

Библиография

1. Шамим Бхуян. Комплексная обработка событий (CEP) с помощью Apache Storm и Apache Ignite // CoderLessons – 2018. – 30 декабря – URL: https://coderlessons.com/articles/java/kompleksnaia-obrabotka-sobytii-cep-s-pomoshchiu-apache-storm-i-apache-ignite (дата обращения: 21.01.2022) – Текст: электронный.
2. Грофф, Джеймс Р., Вайнберг, Пол Н., Оппелъ, Эндрю Дж. SQL: полное руководство, 3-е изд. : Пер. с англ. – М.: ООО "И.Д. Вильяме", 2015. – 960 с. : ил. – Парал. тит. англ. – ISBN 978-5-8459-1654-9 (рус.).
3. Дейт К. Введение в системы баз данных. Introduction to Database Systems. – 8-е изд. – М.: Вильямс, 2006. – С. 1328. – ISBN 5-8459-0788-8.
4. Диго С.М. базы данных. Проектирование и создание: Учебно-методический комплекс. – М.: Изд. центр ЕАОИ. 2008. – 171 с. – ISBN 978-5-374-00055-9.
5. Л.И. Ефремова, Ю.В. Еремкина Информационная безопасность банковской деятельности на примере ПАО

«Сбербанк России» // Контентус – 2016. – №1 (42). – С.20-27.
6. Молинаро Э. SQL. Сборник рецептов. – Пер. с англ. – СПб: Символ-Плюс, 2009. – 672 с., ил. – ISBN-13: 978-5-93286-125-7.
7. Фролов Д.Б. Обеспечение информационной безопасности современных информационных технологий с использованием комплекса документов Банка России в области стандартизации «Обеспечение информационной безопасности организаций банковской системы Российской Федерации» //Деньги и кредит – 2014. – №12 – С.63-66.
8. Ilya Grigorik. High-Performance Browser Networking. September 2013. O'Reilly Media, Inc. ISBN: 9781449344764.
9. Miriam Ramos-Barberán, Miriam Vanessa Hinojosa-Ramos, José Ascencio-Moreno, Francisco Vera, Omar Ruiz-Barzola & María Purificación Galindo-Villardón (2018) Batch process control and monitoring: a Dual STATIS and Parallel Coordinates (DS-PC) approach, Production & Manufacturing Research, 6:1, 470-493, DOI: 10.1080/21693277.2018.1547228.
10. Priyanshi Sharma. Top 10 Software Vulnerabilities And How to Mitigate Them. DEV Community – A constructive and inclusive social network for software developers // DEV Community – 2021. – URL: https://dev.to/priyanshi_sharma/top-10-software-vulnerabilities-and-how-to-mitigate-them-do0 (дата обращения: 27.11.2021) – Текст: электронный.

Mechanisms for ensuring the security of banking systems when using Apache Ignite

Aleksandra D. Senigova

Bachelor
Financial University under the Government of the Russian Federation,
125993, 49, Leningradskii ave., Moscow, Russian Federation;
e-mail: asenigova@gmail.com

Svetlana L. Larionova

Candidate of Technical Sciences,
Associate Professor of the Department "Information Security",
Financial University under the Government of the Russian Federation,
125993, 49, Leningradskii ave., Moscow, Russian Federation;
e-mail: sllarionova@fa.ru

Abstract

This article discusses in detail the requirements for using Apache Ignite in automated banking systems and defines the necessary cybersecurity mechanisms and their configuration parameters when using Apache Ignite in banking systems. Apache Ignite is an open source distributed database, caching and processing platform designed to store and compute large amounts of data. In many large banks, the software product has become the "basis of the IT landscape" for organizing data warehouses and processing large amounts of data online in parallel processes. With all the advantages, the main disadvantage of this product is the lack of initially configured information security parameters. This leads to the impossibility of effective monitoring of systems built on the basis of the product and ensuring security. As part of the work, the product was analyzed for compliance with the requirements of the legislation of the Russian Federation in the field of information security of banking automated systems and a list of vulnerabilities that arise when using the product in banking systems was determined. The article presents a list of necessary parameters

for configuring the Apache Ignite cluster and the security plugin, and a list of requirements for configuring security on the infrastructure side, taking into account legal requirements.

For citation

Senigova A.D., Larionova S.L. (2022) Mekhanizmy obespecheniya bezopasnosti bankovskikh AS pri ispol'zovanii Apache Ignite [Mechanisms for ensuring the security of banking systems when using Apache Ignite]. *Ekonomika: vchera, segodnya, zavtra* [Economics: Yesterday, Today and Tomorrow], 12 (5A), pp. 432-442. DOI: 10.34670/AR.2022.71.72.039

Keywords

In-Memory Data Grid, Apache Ignite, cybersecurity, bank automated system, vulnerabilities.

References

1. Bhuyan, S. (2018). Complex Event Processing (CEP) using Apache Storm and Apache Ignite [CoderLessons] URL: https://coderlessons.com/articles/java/kompleksnaia-obrabotka-sobytii-cep-s-pomoshchiu-apache-storm-i-apache-ignite (accessed: January, 21, 2022). (In Russ.)
2. Groff, James R., Weinberg, Paul N., Oppel, Andrew J. SQL: a complete guide, 3rd ed.: Translated from English – M.: I.D. Williams LLC, 2015. – 960 p. : ill. – Par. tit. English – ISBN 978-5-8459-1654-9. (In Russ.)
3. ate K. Introduction to database systems. Introduction to Database Systems. – 8th ed. – Moscow: Williams, 2006. – p. 1328. – ISBN 5-8459-0788-8. (In Russ.)
4. Digo S.M. databases. Design and creation: Educational and methodical complex. – M.: Publishing house of the center of the EAOI. 2008. – 171 p. – ISBN 978-5-374-00055-9. (In Russ.)
5. Efremova, L.I., Eremkina, Yu.V. (2016) Information security of banking activity on the example of PJSC Sberbank of Russia: Contentus, №1 (42), Pp.20-27. (In Russ.)
6. Molinaro E. SQL. (2009) Collection of recipes. [Translated from English] St. Petersburg: Symbol-Plus – 672 p., ill. – ISBN-13: 978-5-93286-125-7. (In Russ.)
7. Frolov D.B. (2014) Ensuring information security of modern information technologies using a set of documents of the Bank of Russia in the field of standardization "Ensuring information security of organizations of the banking system of the Russian Federation": Money and credit, №12Б pp.63-66. (In Russ.)
8. Ilya Grigorik. (2013) High-Performance Browser Networking [O'Reilly Media, Inc.]. ISBN: 9781449344764.
9. Miriam Ramos-Barberán, Miriam Vanessa Hinojosa-Ramos, José Ascencio-Moreno, Francisco Vera, Omar Ruiz-Barzola & María Purificación Galindo-Villardón (2018) Batch process control and monitoring: a Dual STATIS and Parallel Coordinates (DS-PC) approach, Production & Manufacturing Research, 6:1, 470-493, DOI: 10.1080/21693277.2018.1547228.
10. Priyanshi Sharma. (2021) Top 10 Software Vulnerabilities And How to Mitigate Them. DEV Community – A constructive and inclusive social network for software developers [DEV Community] URL: https://dev.to/priyanshi_sharma/top-10-software-vulnerabilities-and-how-to-mitigate-them-do0 (accessed: November, 27, 2021)