iRule

Технологии анализа информации

и визуализации знаний

РУКОВОДСТВО ПО УСТАНОВКЕ И НАСТРОЙКЕ

СОДЕРЖАНИЕ

1. Перечень сокращений 4

2. Перечень терминов и определений 6

3. Введение 9

3.1. О продукте 9

3.2. Ключевые преимущества 10

3.3. Техническая поддержка 11

4. Описание архитектуры 12

4.1. Компоненты платформы 12

4.2. Характеристики технических средств 13

5. Структура и состав дистрибутива 15

6. Порядок установки и настройки 17

7. Сервис проверки лицензий 18

8. Сервер приложения 22

8.1. Установка 22

8.2. Запуск 41

8.3. Перезагрузка 42

8.4. Остановка 42

8.5. Обновление 42

8.6. Удаление 44

9. Сервис мониторинга 45

9.1. Установка 45

9.2. Обновление 49

9.3. Удаление 50

10. Настройка обращения к внешним информационным ресурсам 51

11. Клиент 53

11.1. Установка 53

11.2. Обновление 62

11.3. Удаление 63

Руководство пользователя

ООО «Институт проблем безопасности и анализа информации»

1. ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

Сокращение	Расшифровка
БД	База данных
ГОСТ	Государственный стандарт
ОС	Операционная система
ПО	Программное обеспечение
СУБД	Система управления базами данных
AD	Active Directory
CA	Certificate Authority
IP	Internet Protocol
HTTP, HTTPS	HyperText Transfer Protocol — протокол передачи гипертекста, HyperText Transfer Protocol Secure — расширение протокола HTTP для поддержки шифрования в целях повышения безопасности
JDBC	Java DataBase Connectivity — соединение с базами данных на Java
JMS	Java Messaging Service - коммуникационная служба, используемая для связи между различными приложениями в указанной сети
KDC	Key Distribution Center - сетевая служба, предоставляющая билеты сеансов и временные ключи сеансов, используемые в протоколе проверки подлинности Kerberos V5
LDAP	Lightweight Directory Access Protocol — открытый и кроссплатформенный протокол, используемый для аутентификации служб каталогов
MS	Microsoft
SAN	Storage Area Network

SPN	Service Principal Name
TCP	Transmission Control Protocol
TGT	Ticket-Granting Ticket
UDP	User Datagram Protocol — протокол пользовательских датаграмм
X.509	Стандарт ITU–T для инфраструктуры открытого ключа

2. ПЕРЕЧЕНЬ ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

Термин	Определение	Источник
Архитектура	Описание (модель) основного устройства (структуры) и связей частей системы (физического или концептуального объекта или сущности). Существует только два типа архитектур, имеющих отношение к интеграции предприятия, а именно: a) системные архитектуры (называемые иногда архитектурами «типа 1»), действие которых распространяется на проектирование системы, например, на компьютеризированную, являющуюся частью системы интеграции предприятия; b) стандартные проекты предприятия (называемые иногда архитектурами «типа 2»), действие которых распространяется на организацию разработки и выполнения проекта, например, интеграцию предприятия или другую программу развития предприятия	ГОСТ Р ИСО 15704–2008
Доступ к информации	Возможность получения информации и ее использования	Федеральный закон №149–ФЗ от 27.07.2006
Комплекс	Программа, состоящая из двух или более компонентов и (или) комплексов, выполняющих взаимосвязанные функции, и применяемая самостоятельно или в составе другого комплекса	ГОСТ 19.102–77
Компонент	Программа, рассматриваемая как единое целое, выполняющая законченную функцию и применяемая самостоятельно или в составе комплекса	ГОСТ 19.102–77
Лицензия	(от лат. Licentia – разрешение) – разрешение на право, либо право на	Электронный словарь

	выполнение некоторых действий, которое может удостоверяться (подтверждаться) одноименным документом. На практике лицензиями также сокращённо именуются лицензионные договоры (соглашения), предусматривающие выдачу частноправовых лицензий	финансовых терминов https://normative_reference_dictionary.academic.ru/32526/Лицензия
Метаданные	Структурированные данные, представляющие собой характеристики описываемых сущностей для целей их идентификации, поиска, оценки, управления ими. Набор допустимых структурированных описаний, которые доступны в явном виде и предназначение которых может помочь найти объект (в контексте поиска объектов, сущностей, ресурсов)	Информационный портал «Российские электронные библиотеки» www.elbib.ru
Метаобласть	Схема в базе данных, предназначенная для хранения метаданных	
Модель	Абстрактное представление реальности в любой форме (включая математическую, физическую, символическую, графическую или описательную) для представления определённого аспекта этой реальности для ответа на рассматриваемые вопросы	ГОСТ Р ИСО 15704–2008
Мониторинг	Постоянное наблюдение за конфигурационной единицей, ИТ–услугой или процессом с целью обнаружения событий и обеспечения информированности о текущем состоянии	Словарь терминов ITIL®, от 29 июля 2011
Организация	Государственное, коммерческое или некоммерческое учреждение имеющее цели, задачи и функции, а также распределение обязанностей и полномочий в рамках целей, задач и функций	ГОСТ Р ИСО 15704–2008
Показатель	Параметр, характеризующий объект учета, государственную услугу, специфические и типовые полномочия государственного	Приказ Минкомсвязи России от

	органа, деятельность государственного органа, относящуюся к установленной сфере ведения государственного органа	01.04.2013 №71
Пользователь (системы)	Лицо или группа лиц, извлекающих пользу в процессе применения системы	ГОСТ Р ИСО/МЭК 15288–2005
Система	Комбинация взаимодействующих элементов, организованных для достижения одной или нескольких поставленных целей	ГОСТ Р ИСО/МЭК 15288–2005
Техническое обеспечение автоматизированной системы	Совокупность всех технических средств, используемых при функционировании автоматизированной системы	ГОСТ 34.003–90
Характеристика	Совокупность количественных и качественных отличительных свойств программного и технического обеспечения, предусмотренных для реализации мероприятия по информатизации	Приказ Минкомсвязи России от 01.04.2013 №71
Элемент системы	Представитель совокупности элементов, образующих систему. Элемент системы является отдельной частью системы, которая может быть создана для выполнения заданных требований	ГОСТ Р ИСО/МЭК 15288–2005

3. ВВЕДЕНИЕ

Настоящий документ является наиболее полным описанием установки программного продукта **iRule®** и его настройки, доступным на момент создания документа.

Разработчиком и правообладателем **iRule®** является **ООО «Институт проблем безопасности и анализа информации»** (**Security Problems Institute Ltd**).

iRule® зарегистрирован Федеральной службой по интеллектуальной собственности в Реестре программ для ЭВМ 09 января 2013 года (Свидетельство о государственной регистрации программы для ЭВМ № 2013610874).

iRule® зарегистрирован Министерством связи и массовых коммуникаций РФ в Едином реестре российских программ для электронных вычислительных машин и баз данных 29 марта 2017 года (Рег. номер ПО: 3242).

iRule® BigData зарегистрирован Министерством связи и массовых коммуникаций РФ в Едином реестре российских программ для электронных вычислительных машин и баз данных 03 декабря 2018 года (Рег. номер ПО: 5033).

Доступные функциональные возможности определяются в соответствующем лицензионном договоре.

Снимки интерфейса, используемые для демонстрации возможностей **iRule®**, могут отличаться от экранных форм. Эти отличия определяются версией **iRule®** и настройкой операционной системы, и не являются существенными при описании функциональных возможностей данной версии.

3.1. О ПРОДУКТЕ

iRule® – это функционально полная технологическая платформа для построения информационно-аналитических систем и ситуационных центров на базе средств визуального анализа информации.

Созданные на базе **iRule** информационно-аналитические системы результативно применяются для обеспечения различных видов деятельности (проведение проверок и расследований, выявление и предотвращение мошенничества, противодействие отмыванию денежных средств и финансированию терроризма, оптимизация бизнес-процессов, управление рисками и не только).

iRule предоставляет мощные инструменты анализа и представления информации, позволяющие проводить:

- *анализ транзакционных связей*, например, анализ трафика – телефонного, почтового, сетевого, финансового, транспортного
- *анализ ролевых связей*, например, анализ сетевых типологий, структур организаций, деловых и личных связей между людьми

- *анализ многомерных данных* (в том числе с использованием OLAP-технологий), например, анализ различных статистических, экономических показателей
- *анализ текстовой информации* – контекстный поиск в файлах, формирование схематического представления объектов и связей, содержащихся в текстах

3.2. КЛЮЧЕВЫЕ ПРЕИМУЩЕСТВА

Простые в использовании инструменты и методы **iRule** позволяют не только сократить время на решение стандартных задач, связанных со сбором информации из различных источников и её предварительной оценкой, но и эффективно решать задачи детального анализа и построения точных обоснованных выводов.

Ключевые преимущества решений на базе **iRule**:

- эффективная поддержка всех основных стадий аналитического процесса: от сбора информации до представления аналитических выводов для принятия решений;
- встроенные мощные инструменты анализа и представления информации (анализ связей, анализ потоков, временной анализ событий, анализ версий (гипотез), табличный и кросс-табличный анализ, картографический анализ и др.);
- прозрачный для пользователя поиск по всем доступным внутренним и внешним информационным ресурсам;
- создание выразительных аналитических материалов, интуитивно понятных как коллегам, так и руководителям, формирование отчётов в электронном и бумажном виде;
- открытость решения для интеграции с другими системами;
- лёгкость в использовании и быстрое освоение;
- соответствие требованиям и рекомендациям **Международной ассоциации аналитиков правоприменительных органов** (International Association of Law Enforcement Intelligence Analysts, IALEIA);
- кратчайшие сроки разработки и внедрения.

Преимуществом **iRule** является возможность его использования не только индивидуально как настольное приложение, но и в качестве единого корпоративного решения.

При разработке **iRule** был учтён опыт работы аналитических подразделений правоприменительных органов России.

iRule не просто программное обеспечение для визуализации информации, это комплексное интеллектуальное решение для поддержки аналитической деятельности на любом уровне и в различных сферах.

3.3. ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

Техническую поддержку **iRule** осуществляет **ООО «Институт проблем безопасности и анализа информации»**.

Порядок осуществления технической поддержки и условия её предоставления, включая доступ в систему управления требованиями, устанавливаются в договоре.

Свои вопросы, замечания или предложения можно направить по электронной почте или телефону. Контакты приведены на сайтах www.spi2.ru и www.irule.ru.

4. ОПИСАНИЕ АРХИТЕКТУРЫ

4.1. КОМПОНЕНТЫ ПЛАТФОРМЫ

iRule® представляет собой совокупность функциональных элементов, установленных на технических средствах из состава комплекса технических средств и обеспечивающих автоматизацию деятельности пользователей.

Программный продукт **iRule®** построен по трёхзвенной архитектуре: клиент – сервер приложения – сервер БД, функционирует на базе серверов и персональных компьютеров и содержит следующие компоненты:

- **Рабочая станция пользователя** предназначена для решения задач анализа и формирования схем, выявления и мониторинга типологий. Для извлечения объектов и связей запрос в понятиях предметной области направляется на сервер приложения. От сервера приложения ответ приходит в виде объектов и связей. Взаимодействие с сервером приложения осуществляется по протоколам **HTTP/HTTPS** и **JMS**
- **Рабочая станция администратора** предназначена для изменения параметров сервера приложения, управления пользователями и их правами, подключения источников данных. Взаимодействие с сервером приложения осуществляется по протоколам **HTTP/HTTPS** и **JMS**
- **Сервер приложения** предназначен для выполнения прикладных процессов. Сервер приложения, с одной стороны, взаимодействует с клиентами, получая

задания, с другой стороны, взаимодействует с источниками данных (реляционными или многомерными БД, веб-сервисами), извлекая данные, необходимые для обработки. Взаимодействие с сервером БД осуществляется с помощью **JDBC**-драйвера соответствующей СУБД. Взаимодействие с внешними и внутренними информационными ресурсами осуществляется по поддерживаемым ими протоколам

- **Сервер метаобласти**. Данный элемент входит в структуру функционально, на нём развёрнута метаобласть сервера приложения. На сервере должна быть установлена любая свободная или проприетарная реляционная СУБД, предоставляющая **JDBC**-драйвер
- **Внутренние и внешние информационные ресурсы**. Данные элементы входят в структуру функционально, выступают в качестве источников данных для проведения анализа. Данные могут быть размещены под управлением свободной или проприетарной СУБД (должна предоставить **JDBC**-драйвер, например, **Arenadata DB, Arenadata QuickMarts, ClickHouse, Greenplum, Postgres Pro, Oracle Database, MS SQL Server, PostgreSQL, MySQL, Sybase, DB2, SAP HANA, MariaDB, TeraData** и т.п.), выставлены в виде веб-сервисов (**SOAP** или **REST**) или доступны по специализированному API

Необходимость выделения отдельных функциональных элементов является следствием модульности структуры программного обеспечения **iRule®** и вызвана требованиями обеспечения независимости функций накопления и анализа данных, а также необходимостью обработки и продолжительного хранения больших объёмов данных.

Допускается размещение нескольких компонент совместно на одном сервере (например, для формирования тестовой среды). Возможно обеспечить терминальный доступ пользователей и администраторов.

Данная архитектура позволяет минимизировать администрирование: не требуется установка и обновление ПО пользователя и администратора на каждой рабочей станции, этот процесс происходит автоматически с сервера приложения.

4.2. ХАРАКТЕРИСТИКИ ТЕХНИЧЕСКИХ СРЕДСТВ

Требуемые характеристики технических средств зависят от количества пользователей, вычислительной сложности выполняемых задач и объёма обрабатываемой информации.

Рекомендуемые минимальные характеристики технических средств:

Сервер приложения	**Рабочая станция пользователя и администратора**
• 2 процессора Intel Xeon с тактовой частотой каждого ядра 3,0 ГГц	• процессор с тактовой частотой 3,0 ГГц

• оперативная память - 16 Гбайт • дисковая подсистема HDD - 100 Гбайт • сетевой адаптер - 1 Гбит/с Ethernet • операционная система - Windows, Linux, Solaris	• оперативная память - 8 Гбайт • дисковая подсистема HDD - 80 Гбайт • монитор - 1920*1080 • сетевой адаптер - 1 Гбит/с Ethernet • операционная система - Windows, Linux

В качестве операционной системы поддерживаются и отечественные решения.

5. СТРУКТУРА И СОСТАВ ДИСТРИБУТИВА

Стандартно дистрибутив с программным продуктом **iRule** имеет следующую структуру:

- **/doc**
- **/irule-app**
- **/irule-gate**
- **/irule-metadata**
- **/irule-server-8080**
- **/irule-watchdog**
- **/utils**
- **readme.txt**

В папке **doc** находятся документы:

- **Руководство администратора** (необязательный документ, в котором приводится описание работы с **iRule Administrator**)
- **Руководство по установке и настройке** (настоящий документ)
- **Регламент работы с системой управления требованиями** (необязательный документ, в котором приводится описание работы с системой управления требованиями; возможность работы с системой управления требованиями (и, соответственно, наличие данного документа) определяется в лицензионном договоре)
- **Примечания к выпуску** (необязательный документ, в котором приводятся изменения между представленной на дистрибутиве и предыдущими версиями ПП)

Документы представлены в формате .pdf. Для чтения документов в данном формате можно воспользоваться находящемся в папке **utils** ПП Adobe Reader.

В папке **irule-app** находится дистрибутив (упакованный в виде war-файла) актуальной версии ПП **iRule**.

Папка **irule-gate** может отсутствовать. В папке **irule-gate** находится дистрибутив сервера взаимодействия. Возможность работы через сервер взаимодействия (и, соответственно, наличие папки) определяется в лицензионном договоре.

В папке **irule-metadata** находятся скрипты для развёртывания метаобласти сервера приложения в поддерживаемых СУБД.

В папке **irule-server-8080** находится дистрибутив контейнера сервлетов с открытым исходным кодом **Apache Tomcat**.

В папке **irule-watchdog** находится дистрибутив сервиса мониторинга функционирования платформы **iRule Watchdog**.

В папке **utils** находятся дистрибутивы сервиса проверки лицензий и ряда вспомогательных программ.

В документе **readme.txt** приведена информация о версии ПП **iRule**, составе и структуре дистрибутива.

6. ПОРЯДОК УСТАНОВКИ И НАСТРОЙКИ

Установка и настройка **iRule** выполняется в следующем порядке:

1. Установка сервиса проверки лицензий.
2. Установка сервера приложения.
3. Установка сервиса мониторинга (необязательный шаг).
4. Установка сервера взаимодействия (необязательный шаг).
5. Установка клиента.

7. СЕРВИС ПРОВЕРКИ ЛИЦЕНЗИЙ

В рамках лицензионного договора лицензиату предоставляется право на запуск определённого количества серверов приложений (**iRule Server**) и клиентских приложений (**iRule Client** и **iRule Administrator**). Данные сведения записываются на аппаратное средство (HASP-ключ). Это устройство подключается через USB-порт. Взаимодействие с HASP-ключом осуществляется через сервис проверки лицензий.

При запуске сервер приложения и клиентское приложение резервируют соответствующую свободную лицензия на HASP-ключе на всё время работы. Если отсутствует свободная лицензия, приложение не запустится. Сервер приложения всегда сам взаимодействует с сервисом проверки лицензий, а клиент это взаимодействие может делегировать серверу.

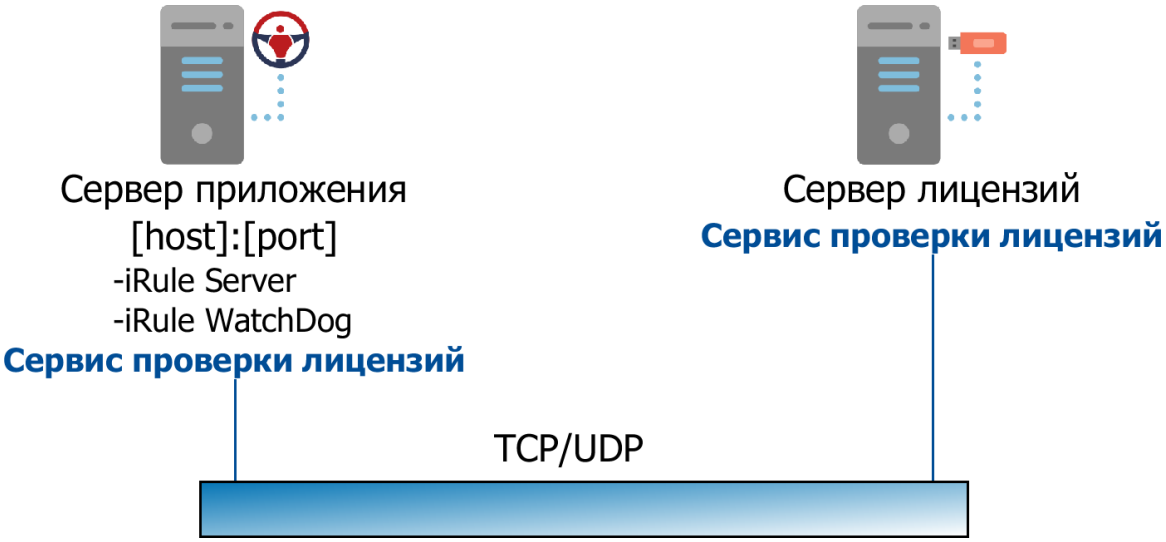
Проверка лицензии сервером приложения

Возможны следующие варианты:

1 вариант (основной). Как правило подключают HASP-ключ и устанавливают сервис проверки лицензий на том же компьютере, на котором разворачивают и сервер приложения (**iRule Server**).

Сервер приложения
[host]:[port]
-iRule Server
-iRule WatchDog
Сервис проверки лицензий

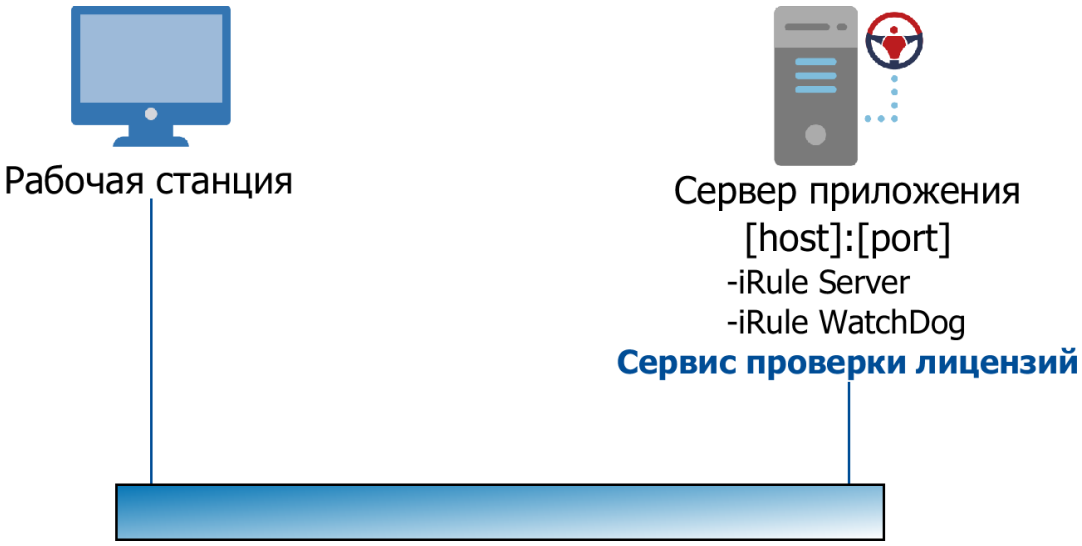
2 вариант. Если на сервере приложения нет возможности подключить HASP-ключ (например, на компьютере может отсутствовать свободный USB-порт или это виртуальная машина, на которую нет возможности его пробросить), то подключить HASP-ключ и установить сервис проверки лицензий можно на любом компьютере сети, доступном с сервера приложения по протоколам **TCP** и **UDP**. И в этом варианте на сервере приложения также необходимо установить сервис проверки лицензий и настроить доступ к серверу лицензий.



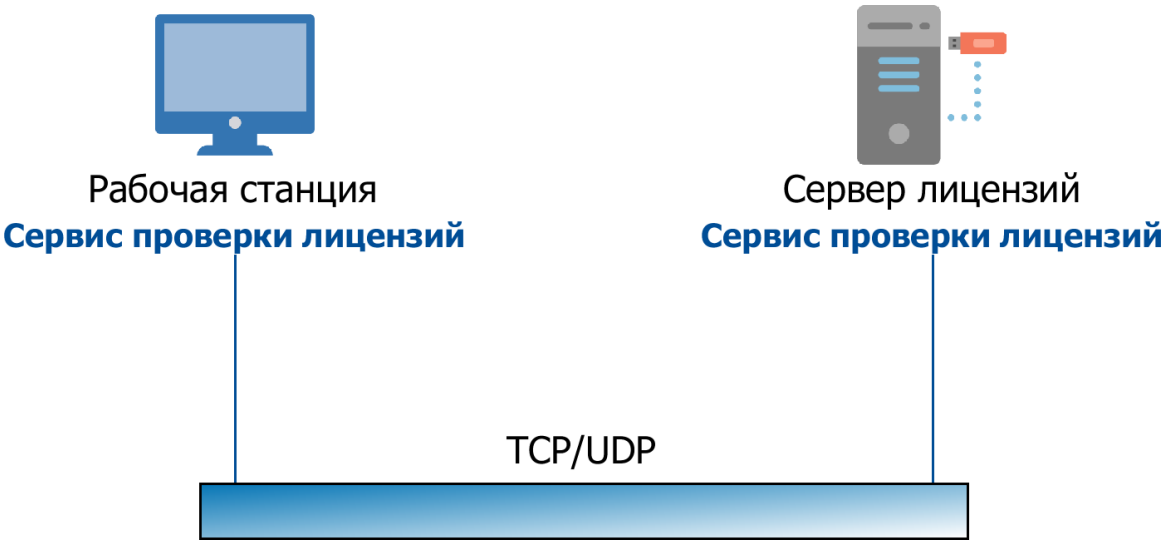
Проверка лицензии клиентским приложением

Возможны следующие варианты:

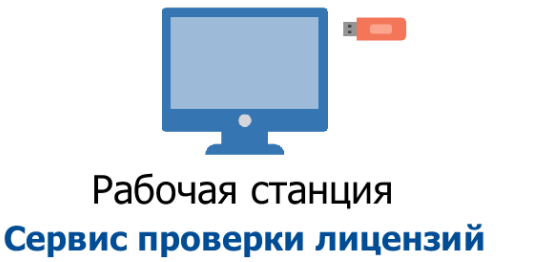
1 вариант (основной). Клиентское приложение делегирует работу с лицензией серверу приложения. Установки сервера лицензий на рабочей станции клиента не требуется. Информация о данном варианте работы записывается на HASP-ключ, при этом работа с клиентом без подключения к серверу невозможна.



2 вариант. Клиентское приложение запрашивает свободную лицензию с сервера лицензий. В этом случае на рабочей станции также необходимо установить сервис проверки лицензий и настроить доступ по протоколам **TCP** и **UDP** к серверу лицензий.



3 вариант. Подключить HASP-ключ и установить сервис проверки лицензий непосредственно на рабочей станции.



Установка сервиса проверки лицензий

Для установки сервиса проверки лицензий выполните скрипт с правами Администратора:

\irule-server\utils\hasp-driver\install-hasp-driver.bat.

После установки подключите HASP-ключ к свободному USB-порту. Если HASP-ключ оборудован светодиодом, он должен замигать.

Проверка доступности и состояния HASP-ключа

Для проверки доступности и состояния HASP-ключа выполните следующие действия:

1. В браузере откройте страницу http://localhost:1947.
2. В колонке **Options** выберите пункт **Sentinel Keys**.
3. В строке необходимого ключа в колонке **Actions** нажмите **Blink on**.
4. На ключе загорится светодиод.

Настройка сервиса проверки лицензий

Если сервис проверки лицензий с HASP-ключом развернут на отдельном хосте (см. 2 вариант) в другой подсети, то на сервере приложения и клиентском приложении

необходимо выполнить дополнительную настройку развёрнутого сервиса проверки лицензий.

Выполните следующие действия:

1. В браузере откройте страницу http://localhost:1947.
2. В меню **Options** выберите пункт **Configuration**.
3. Выберите вкладку **Access to Remote License Managers**.
4. Снимите флажок **Broadcast Search for Remote Licenses**.
5. В поле **Specify Search Parameters** явным образом укажите IP-адрес или имя сервера приложения, где установлен ключ HASP-ключ.
6. Нажмите **Submit**, закройте браузер и перезагрузите ОС или перезапустите службу **Sentinel HASP LicenseManager**.

Удаление сервиса проверки лицензий

Для удаления драйвера выполните следующие действия:

1. Войдите в систему как администратор.
2. Если возможно, временно отключите любое защитное ПО (антивирус, брандмауэр).
3. Если на сервис проверки лицензий установлен на хосте, то извлеките HASP-ключ из USB-порта.
4. Запустите **\irule-server\utils\hasp-driver\haspdinst.exe** с опциями `-r -kp`.

8. СЕРВЕР ПРИЛОЖЕНИЯ

Установка, запуск, обновление и удаление сервера приложения **iRule**.

8.1. УСТАНОВКА

Последовательность установки

Для установки и запуска сервера выполните следующие действия:

1. Скопируйте дистрибутивы.
2. Создайте метаобласть.
3. Настройте подключение к метаобласти.
4. Настройте используемые порты для **HTTP**, **HTTPS**, **JMS** и **Tomcat**.
5. Настройте размер выделяемой оперативной памяти.
6. Настройте уровень журналирования.
7. Настройте авторизацию через **Контроллер домена**, **Active Directory (AD)**, **Kerberos**.
8. Запустите сервер.

Примечание.
Запущенный антивирус Касперского может приводить к значительному замедлению запуска и работы сервера приложения. Если это для Вас критично, то рекомендуется отключить его или настроить исключения.

Копирование дистрибутивов

Создайте папку **irule-server** (например, **C:\irule-server**) и скопируйте в неё папки **irule-app** и **irule-server-8080** (с дистрибутивами актуальной версии ПП **iRule** и контейнера сервлетов).

Создание метаобласти

Для создания метаобласти выполните следующие действия:

1. Запустите приложение, использующееся для работы с СУБД, развёрнутой на сервере БД.
2. Установите соединение с БД под пользователем, обладающим правами создания пользователей.
3. Выполните скрипт создания пользователя-владельца метаобласти: **create-user.sql** (расположен в папке **irule-metadata\[database]**)
4. Установите соединение с БД под пользователем-владельцем метаобласти.

5. Выполните скрипт создания и наполнения структур метаобласти: **create-metadata.sql** (расположен в папке **irule-metadata\[database]**)

Настройка подключения к метаобласти

Настройка подключения сервера к метаобласти выполняется в файле **irule-server-8080\conf\irule\irule-server.properties**. Данный файл формируется в кодировке **UTF-8**.

Примечание. Во избежание проблем с кодировкой для редактирования данного файла рекомендуется использовать текстовые редакторы, корректно определяющие кодировку **UTF-8**. Например, к ним относятся инструменты **Notepad++** и **FAR Manager**, которые входят в перечень поставляемых утилит.

В файле приведены шаблоны настроек для следующих СУБД: **Oracle**, **Postgres**, **MS Sql Server**, **SAP Hana Server**. По умолчанию выбрана СУБД **Oracle**. Остальные закомментированы (в начале строки поставлен знак #).

Задайте строку подключения к базе данных с метаобластью, логин и пароль пользователя-владельца метаобласти:

```
persistence.security.metadata.url=jdbc:oracle:thin:@[host]:[port]:[sid]
persistence.security.metadata.username=[username]
persistence.security.metadata.password=[password]
```

Пароль может быть внесён как в открытом, так и в зашифрованном виде.

Чтобы внести в свойство зашифрованный пароль:

1. Задайте свойство `persistence.security.metadata.password.encrypted=true`.
2. Зашифруйте пароль в соответствии с инструкцией, приведённой в подразделе Шифрование паролей.
3. Внесите зашифрованный пароль в свойство `persistence.security.metadata.password`.

Настройка портов для протоколов HTTP, HTTPS, JMS и служебного порта

Подключение клиентов к серверу приложения происходит по протоколам **HTTP** и **JMS**. Может быть настроена работа по протоколу **HTTPS**. Также используется служебный порт.

По умолчанию используются следующие порты:

8080 – для **HTTP**

8443 – для **HTTPS** (не обязателен, не настроен)

7676 – для **JMS**

8005 – служебный

Для того чтобы увидеть список занятых и свободных портов, выполните команду:

```
netstat -a
```

Смена порта для протокола HTTP

По умолчанию сервер приложений настроен на **HTTP** соединение по порту **8080**.

Для смены порта HTTP-доступа выполните следующие действия:

1. Откройте на редактирование файл **irule-server-8080\conf\server.xml**.
2. Найдите настройки элемента **Connector** с протоколом **HTTP/1.1**, например:

```
<Connector executor="tomcatThreadPool" port="8080"
protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443"
maxParameterCount="1000" />
```

3. Замените значение в поле **port**.
4. Рекомендуется переименовать папку сервера приложения с **irule-server-8080** на **irule-server-[port]**.

Примечание. В случае настроенного **iRule Watchdog**, в скриптах запуска/остановки сервера приложения необходимо выполнить правки в соответствии с новым именем каталога.

Для проверки использования порта выполните в командной строке:

```
> netstat -anop tcp | find "[port]"
```

Если указанный порт используется приложением, то при выполнении приведенной команды для данного порта в статусе будет указано состояние **LISTENING**, а также приведен **PID** процесса, который его использует.

Примечание. Если вы изменили порт, то в дальнейшем вместо папки **irule-server-8080** необходимо использовать папку **irule-server-[port]**.

Смена порта для протокола JMS

Для смены порта выполните следующие действия:

1. Откройте на редактирование файл **irule-server-8080\conf\irule\irule-server.properties**.
2. Найдите свойство `jms.server.port=7676`.

Примечание. Если сервер был запущен, то для того чтобы изменения вступили в силу, необходимо перезапустить сервер приложения. Для этого выполните действия, описанные в подразделе Перезагрузка сервера приложения.

Для проверки того, не используется ли уже необходимый серверу порт, можно выполнить в командной строке команду

```
>netstat -anop tcp | find "[port]"
```

Предварительные условия

Рекомендуется выполнять настройку **HTTPS** на сервере **iRule** только убедившись, что сервер приложения корректно работает в стандартном режиме - по протоколу **HTTP**.

Для проверки выполните следующие действия:

1. Запустите сервер приложения.
2. Запустите клиент.
3. Подключитесь к серверу приложения. Если подключение прошло успешно, перейдите к следующему пункту.
4. Запросите данные из клиента. Если данные получены, то сервер работает корректно.
5. Остановите сервер приложения.

Краткое описание подключения по HTTPS

HTTPS-соединение позволяет шифровать трафик между клиентом и сервером, а также используется для аутентификации клиента и сервера. В нашем случае **HTTPS** используется односторонняя аутентификация - клиент удостоверяется, что попал точно на указанный в настройках сервер и передаёт ему зашифрованные данные.

Для настройки **HTTPS-соединения** необходимо создать **SSL-сертификат** и установить его на сервер приложения. Клиентское приложение при подключении будет получать сертификат и, используя встроенные механизмы работы по **HTTPS**, проводить аутентификацию, шифровать данные и отправлять их на сервер.

Создание SSL-сертификата

Сертификат создается для хоста, на котором будет работать сервер приложения. Вы должны знать доменное имя хоста. Если вы создадите сертификат для одной машины, а сервер приложения с этим сертификатом запустите на другой машине, механизмы проверки **HTTPS-подключения** не дадут возможности подключиться клиенту к серверу, так как это случай, когда сервер пытается выдать себя за другую машину.

Для того чтобы узнать доменное имя машины по её **ip-адресу**, выполните следующую команду в командной строке:

```
nslookup [ip_adr]
```

Например, если `[ip_adr]` равен 10.0.0.135:

```
nslookup 10.0.0.135
name:     spidc2.spi.new
Address:  10.0.0.100
name:     pc31.spi.new
```

```
Address:  10.0.0.135
```

Здесь машина с адресом **10.0.0.135** имеет доменное имя **pc31** в домене **SPI.NEW**.

Сертификат создаётся с помощью утилиты **keytool**, которая входит в **jdk**. Утилита создаёт хранилище сертификатов, в котором хранится сертификат, его приватная и публичная части. Публичная часть будет доступна пользователям для скачивания при подключении к серверу приложения.

При создании сертификата с помощью **keytool** потребуется ввести пароль к хранилищу сертификатов (`Enter keystore password`) и к приватной части сертификата (`Enter key password for <irule-tomcat>`). Рекомендуется задавать одинаковый пароль, он потребуется в шаге **Изменение конфигурации HTTPS-подключения к серверу**.

Пароль может быть введён как в открытом, так и в зашифрованном виде.

Чтобы используемый пароль был зашифрован:

1. Откройте на редактирование файл **irule-server-8080\conf\irule\irule-server.properties** и задайте свойство **https.security.keystore.password.encrypted=true**.
2. Зашифруйте пароль в соответствии с инструкцией, приведённой в подразделе Шифрование паролей.
3. Используйте зашифрованный пароль в шаге **Изменение конфигурации HTTPS-подключения к серверу**.

Перейдите в директорию **irule-server-8080\java\jdk-21.0.4-full\bin**, откройте командную строку в этой директории.

Выполните команду:

```
keytool -genkey -alias [alias_name]> -keyalg RSA -keystore
[keystore_name] -ext "SAN=dns:[pc_name],ip:[ip_adr]" -validity
1825
```

- `genkey` - требуется создать сертификат
- `alias` - в хранилище будет помещён сертификат с названием `[alias_name]`
- `keyalg` - выбор алгоритма
- `keystore` - в `[keystone_name]` указывается путь к файлу, где будет создано хранилище сертификатов
- `ext SAN` - указание свойства `SAN`, с информацией о доменном имени `[pc_name]` и адресе `[ip_adr]` машины, на которой развернут сервер, которая используется для валидации сертификата

Пример выполнения команды, где `[alias_name]` – irule-server, `[keystore_name]` – C:\spi\https\irule-server.keystore, `[pc_name]` – pc31, `[ip_adr]` – 10.0.0.135:

```
keytool -genkey -alias irule-server -keyalg RSA -keystore
C:\spi\https\irule-server.keystore -ext
"SAN=dns:pc31,ip:10.0.0.135" -validity 1825

Enter keystore password:

Re-enter new password:

What is your first and last name?

  [Unknown]:  pc31 <-- здесь вводите [pc_name]

What is the name of your organizational unit?

  [Unknown]:  IT

What is the name of your organization?

  [Unknown]:  SPI

What is the name of your City or Locality?

  [Unknown]:  Moscow

What is the name of your State or Province?

  [Unknown]:  Russia

What is the two-letter country code for this unit?

  [Unknown]:  RU

Is CN=pc31, OU=IT, O=SPI, L=Moscow, ST=Russia, C=RU correct?

  [no]: yes


Enter key password for <irule-server>

  (RETURN if same as keystore password):

Re-enter new password:
```

Создание запроса на подпись **SSL** сертификата для сервера **iRule** удостоверяющим центром **CA**:

```
keytool -keystore [keystore_name] -certreq -alias [alias_name] -
keyalg rsa -file [query_file]
```

В качестве `[query_file]` указывается путь к файлу, содержащему запрос в удостоверяющий центр, например, C:\spi\https\irule-server.csr.

Импорт публичного сертификата **CA** в хранилище **SSL** сертификатов сервера **iRule**:

```
keytool -import -keystore [keystore_name] -file [ca_crt] -alias
[ca-cert]
```

В качестве `[ca_crt]` указывается путь до публичного сертификата, например, C:\spi\https\ca.crt, а в качестве `[ca-cert]` - присваиваемое имя сертификата.

Импорт подписанного сертификата для сервера **iRule** в хранилище сертификатов **iRule**:

```
keytool -import -keystore [keystore_name] -file [crt_name] -alias
[alias_name]
```

В качестве `[crt_name]` указывается путь к подписанному сертификату, например, C:\spi\https\irule-server.crt, а в качестве `[alias_name]` - присваиваемое имя сертификата.

Подпись сертификата удостоверяющим центром

Для создания запроса на подпись **SSL** сертификата для сервера **iRule** удостоверяющим центром (**CA**) выполните следующую команду:

```
keytool -keystore [keystore_name] -certreq -alias [alias_name] -
keyalg rsa -file [query_file]
```

В качестве `[query_file]` указывается путь к файлу, содержащему запрос в удостоверяющий центр, например, C:\spi\https\irule-server.csr.

После создания запроса он передаётся в удостоверяющий центр. На основе данного запроса центр создаёт необходимый сертификат с подписью и отправляет его в ответ.

В случае, если необходимо самостоятельно подписывать сертификаты (самостоятельно выполнять функции **CA**), то необходимо создать приватный ключ и сам корневой сертификат. Для этого используется утилита **OpenSSL**.

Выполните команду для генерации приватного ключа:

```
openssl genrsa -out ca.key 4096
```

Выполните команду для генерации корневого сертификата:

```
openssl req -new -x509 -days 1825 -key ca.key -out [ca_crt]
```

Далее, имея запрос в формате **csr**, можно перейти к созданию сертификата, подписанного корневым сертификатом **CA**.

Создайте конфигурационный файл **openssl-csr.cnf**. Откройте его с помощью текстового редактора и сохраните в нем следующее содержимое, начиная с первой строки:

```
[ req_ext ]

basicConstraints = CA:FALSE

keyUsage = nonRepudiation, digitalSignature, keyEncipherment


############################################

# Edit this line to set subjectAltName contents

############################################


subjectAltName = DNS:[pc_name], IP:[ip_adr]
```

В строке **subjectAltName** указываются доменные имена `[pc_name]` и ip-адрес машины `[ip_adr]`, которые будут внесены в сертификат при его создании.

Выполните команду для генерации сертификата:

```
openssl x509 -req -in [query_file] -CA [ca_crt] -CAkey ca.key -
CAcreateserial -out [crt_name] -days 1825 -extfile openssl-csr.cnf
-extensions req_ext
```

Результатом выполнения команды является подписанный сертификат `[crt_name]`.

Независимо от способа получения подписанного сертификата, необходимо его, а также корневой сертификат, импортировать в хранилище `[keystore_name]`.

Для импорта корневого сертификата выполните команду:

```
keytool -import -alias [ca-cert] -keystore [keystore_name] -
trustcacerts -file [ca_crt]
```

Введите пароль от **keystore** и дайте согласие, введя "`y`", на вопрос о доверии сертификату.

Для импорта сгенерированного для сервера сертификата выполните команду:

```
keytool -import -alias [alias_name] -keystore [keystore_name] -
trustcacerts -file [crt_name]
```

Настройка сервера приложения на работу по протоколу HTTPS

Изменение конфигурации HTTPS-подключения к серверу

Откройте на редактирование файл **irule-server-8080\conf\server.xml**.

Найдите следующий коннектор, который используется для **HTTP**-подключений.

```
<Connector port="8080"
executor="tomcatThreadPool"
protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
```

В случае, если необходима поддержка работы и по **HTTP**, и по **HTTPS**, необходимо удалить свойство `redirectPort="8443"`.

В случае, если поддержка **HTTP** вообще не нужна, необходимо удалить данный коннектор.

Затем добавьте следующий коннектор для включения **HTTPS**:

```
<Connector
   protocol="ru.spi2.security.tomcat.Http11Nio2Protocol"
   port="8443"
   maxThreads="150"
   SSLEnabled="true">
```

```
 <SSLHostConfig>
   <Certificate
     certificateKeystoreFile="[keystore_name]"
     certificateKeystorePassword="[password]"
     type="RSA"
     />
 </SSLHostConfig>
</Connector>
```

Поле `certificateKeystorePassword` указывается в соответствии с введенным паролем хранилища ключей и самого ключа. Он указывался в пункте **Создание SSL-сертификата**.

Проверка установки HTTPS на сервере приложения

Перезапустите сервер приложения. Если приложение будет успешно запущено на сервере, сервер настроен.

Трассировка подключения к серверу по HTTPS

Для просмотра информации в логе сервера, добавьте в конфигурацию журналирования строку:

```
<Logger name="HTTPS-LOGGER" level="TRACE" />
```

Возможность выбора подключения к серверу между HTTP и HTTPS

Сервер приложения может одновременно поддерживать **HTTP** и **HTTPS** соединения.

В этом случае в настройках из пункта **Изменение конфигурации HTTPS-подключения к серверу** следует оставить **connector** со свойством `protocol="HTTP/1.1"`.

В этом случае, если сервер подключается по строке **http://...**, все его подключения будут происходить по протоколу **HTTP**. Если по строке **https://...** - по протоколу **HTTPS** соответственно.

Известные проблемы

В **Firefox** по умолчанию не считаются безопасными сертификаты, которые являются доверенными на уровне **ОС Windows**. Для **Firefox** версии больше **49** внесите изменения в настройки браузера:

1. В строке адреса введите **about:config**.
2. Установите свойство **security.enterprise_roots.enabled** в значение **true**.

Для **Firefox** с версией меньше **49** (релиз ранее сентября 2016) решением является добавление указанного сертификата в локальное хранилище сертификатов браузера.

Смена служебного порта

Сервер приложения разворачивается под управлением контейнера сервлетов с открытым исходным кодом **Apache Tomcat**. Для функционирования ему требуется служебный порт (по умолчанию 8005). Рекомендуется с помощью **firewall** разрешить только локальный доступ по этому порту.

Для смены порта выполните следующие действия:

1. Откройте на редактирование файл **irule-server-8080\conf\server.xml**.
2. Перейдите к строке и замените значение для **port**:

```
<Server port="8005" shutdown="SHUTDOWN">
```

Примечание. Если сервер был запущен, то для того чтобы изменения вступили в силу, необходимо перезапустить сервер приложения. Для этого выполните действия, описанные в подразделе Перезагрузка сервера приложения.

Развёртывание нескольких серверов приложения на одном хосте

При необходимости одновременного запуска нескольких серверов приложения на одном хосте, необходимо каждый сервер приложения расположить в отдельной папке и назначить ему свои собственные порты доступа.

Рекомендуется назначать порты согласованно, например:

`3080` – для HTTP

`3443` – для HTTPS

`3676` – для JMS

`3005` – в качестве служебного

А папку в этом случае назвать **irule-server-3080**.

Настройка размера выделяемой оперативной памяти

Для настройки размера выделяемой оперативной памяти выполните следующие действия:

1. Откройте на редактирование файл **irule-server-8080\bin\setenv.bat**. Перейдите к строке:

```
set "CATALINA_OPTS=%CATALINA_OPTS% -Xms4096M -Xmx4096M"
```

2. Измените следующие значения (для сервера рекомендуется выставлять эти два параметра равными):

- **Xms** – размер памяти, выделяемый при старте
- **Xmx** – максимальный размер памяти, который будет выделен приложению

Примечание. Если сервер был запущен, то для того чтобы изменения вступили в силу, необходимо перезапустить сервер приложения. Для этого выполните действия, описанные в подразделе Перезагрузка сервера приложения.

Настройка уровня журналирования

Для настройки уровня журналирования выполните следующие действия:

1. Откройте на редактирование файл **irule-server-8080\conf\irule\log4j2.xml**
2. Найдите строки:

```
<!-- iRule specified logging -->
<Logger name="ru.spi2" level="WARN" additivity="false">
<AppenderRef ref="server.file.log.appender"/>
</Logger>
```

3. Измените значение для **level**:

- **ALL** – вся информация о ходе выполнения
- **TRACE** – наиболее подробная информация о ходе выполнения
- **DEBUG** – подробная информация о ходе выполнения
- **INFO** – важные действия. Это ожидаемые действия системы
- **WARN** – предостережения. Записываются неожиданные действия, которые не приводят к остановке работы системы, но могут быть нежелательны
- **ERROR** – ошибки при конкретном действии. Записываются ошибки, которые не приводят к остановке работы сервера приложения, но не позволяют выполнить определённое действие
- **FATAL** – критические ошибки, приводящие к остановке работы системы

4. Изменение конфигурации журналирования применяется сразу, без перезагрузки сервера.
5. В случае, когда нужно настроить подробное журналирование для конкретных классов, для них может быть добавлена дополнительная конфигурация, например:

```
<Logger name="ru.spi2.javaee.custom.spark.SparkService2"
level="TRACE" additivity="false">
<AppenderRef ref="server.file.log.appender"/>
</Logger>
```

6. Здесь **name** содержит полное имя класса (пакет + имя класса).

Журнал функционирования контейнера сервлетов находится в **/logs/catalina.log** и **logs/manager.log**

Настройка авторизации

Настройка аутентификации через KERBEROS

Краткое описание процесса аутентификации через KERBEROS

Процесс аутентификации клиента на сервере происходит следующим образом:

1. Клиент в приложении выбирает действие **Подключение к серверу**.
2. Клиентское приложение запрашивает **TGT** у **KDC**.
3. Клиентское приложение отправляет **TGT** серверу.
4. Сервер передает полученный **TGT** контроллеру домена **KDC** и, используя ответ **KDC** и локальное хранилище ключей, аутентифицирует пользователя.
5. Сервер переходит к шагу авторизации пользователя.

Для работы описанного процесса необходимо произвести следующие настройки:

1. Для сервера приложения должен быть сформирован **SPN** и **keytab**-файл.
2. Сервер приложения должен быть настроен на аутентификацию через **Kerberos**.
3. Клиентская **ОС** должна поддерживать хранение и передачу кэшированных билетов **krbtgtr**.

Создание SPN для задачи аутентификации на сервере

SPN должен быть настроен для учётной записи домена, для этого рекомендуется использовать отдельного технологического доменного пользователя.

Создание технологического доменного пользователя

Выполните следующие действия:

1. Войдите на контроллер домена.
2. Выберите пункт **Пуск > AD - пользователи и компьютеры**.
3. Выберите пункт **Создать > Пользователи**.
4. В поле **Имя входа пользователя** введите:

```
spn_user_irule_1
```

5. На форме выбора пароля установите флажок **Срок действия не ограничен**.

Примечание. В случае необходимости работы нескольких серверов приложений у заказчика, для каждого сервера должен быть свой доменной пользователь с именем **spn_user_irule_x**. Для средств тестирования/разработки, для всех серверов приложений достаточно одного доменного пользователя, которому будет настроен **SPN**.

Создание SPN

Выполните следующие действия:

1. Проверьте, что для пользователя отсутствует **SPN**, введя команду:

```
> setspn -L spn_user_irule_1
```

Вариант ответа (только 1 строка):

```
Зарегистрирован ServicePrincipalNames для
CN=spn_user_irule_1,CN=Users,DC=SPI,DC=NEW:
```

2. Создайте **Kerberos Service Principal Name** для пользователя домена, созданного в предыдущем шаге:

```
> setspn -A HTTP/spn_irule_1 spn_user_irule_1
```

Здесь:

- `HTTP/spn_irule_1` - имя создаваемого **SPN**
- `spn_user_irule_1` - доменный пользователь, для которого будет создан **SPN**

Ожидаемый ответ:

```
Регистрация ServicePrincipalNames для
CN=spn_user_irule_1,CN=Users,DC=SPI,DC=NEW

 HTTP/spn_irule_1
```

Примечание. В случае необходимости работы нескольких серверов приложений у заказчика, для каждого сервера должен быть свой **SPN** с именем **spn_irule_x**. Для средств тестирования/разработки, для всех серверов приложений достаточно одного **SPN**, который будет указан в **irule-server.properties** этих серверов.

3. Проверить, что для пользователя создан **SPN**:

```
setspn -L spn_user_irule_1
```

Ожидаемый ответ (выводится имя **SPN**, добавленного на предыдущем шаге):

```
Зарегистрирован ServicePrincipalNames для
CN=spn_user_irule_1,CN=Users,DC=SPI,DC=NEW:

 HTTP/spn_irule_1
```

Формирование хранилища ключей (Keytab) для SPN

Хранилище ключей в дальнейшем будет храниться на машине сервера **iRule**, и использоваться им в процессе аутентификации.

```
ktpass -out [keytab_name] -mapuser spn_user_irule_1@SPI.NEW -princ
HTTP/spn_irule_1@SPI.NEW -pass [password] -ptype KRB5_NT_PRINCIPAL
-kvno 18 -crypto All
```

Здесь:

- `[keytab_name]` - каталог, в котором будет создан **keytab**-файл, например, `C:\spn_irule_1.keytab`
- `-mapuser spn_user_irule_1@SPI.NEW` - имя доменного пользователя, для которого осуществляется маппинг **SPN**, **SPI.NEW** - это **REALM**, его значение указано в файле **krb5.conf**
- `-princ HTTP/spn_irule_1@SPI.NEW` - имя **SPN**

- `-pass` - пароль доменного пользователя

Ожидаемый результат:

```
Targeting domain controller: SPIDC2.SPI.NEW

Using legacy password setting method

Successfully mapped HTTP/spn_irule_1 to spn_user_irule_1.

Key created.

Key created.

Key created.

Key created.

Key created.

Output keytab to c:\spn_irule_1.keytab:

Keytab version: 0x502

keysize 51 HTTP/spn_irule_1@SPI.NEW ptype 1 (KRB5_NT_PRINCIPAL)
vno 18 etype 0x1 (DES-CBC-CRC) keylength 8 (0x8394adba2c435b80)

keysize 51 HTTP/spn_irule_1@SPI.NEW ptype 1 (KRB5_NT_PRINCIPAL)
vno 18 etype 0x3 (DES-CBC-MD5) keylength 8 (0x8394adba2c435b80)

keysize 59 HTTP/spn_irule_1@SPI.NEW ptype 1 (KRB5_NT_PRINCIPAL)
vno 18 etype 0x17 (RC4-HMAC) keylength 16
(0x77c58bf53216e15ffa1d9e84b7bf3023)

keysize 75 HTTP/spn_irule_1@SPI.NEW ptype 1 (KRB5_NT_PRINCIPAL)
vno 18 etype 0x12 (AES256-SHA1) keylength 32
(0xc15601f10234e4a0f72b226ac17d8d4a8ab78485a41893770982c1f7ee21a17
f)

keysize 59 HTTP/spn_irule_1@SPI.NEW ptype 1 (KRB5_NT_PRINCIPAL)
vno 18 etype 0x11 (AES128-SHA1) keylength 16
(0xff620d5d658a39556a5d1e6250a0dfe4)
```

Примечание. Keytab-файл представляет собой хранилище паролей пользователей домена. На продуктивной среде **keytab**-файл должен находиться на машине, где работает сервер приложений. Данный файл не должен быть доступен извне. Для средств тестирования/разработки, для всех серверов приложений можно пользоваться одним и тем же **keytab**-файлом.

Настройка сервера приложения для аутентификации по Kerberos

Следующие изменения следует производить на машине, где будет работать сервер приложения.

Добавление файлов на машину сервера

Переместите сформированный файл **keytab** в папку сервера, для которого он предназначен, например, в папку **C:\irule-server\irule-server-8080\conf\irule\kerberos**. В дальнейшем, в настройках **irule-server.properties** будет указан путь к данному файлу.

Установка серверных свойств для работы Kerberos

Впишите в файл настроек сервера **irule-server-8080\conf\irule\irule-server.properties** настройки **SPN** и **keytab**.

```
security.authentication.mode=kerberos

security.authentication.sso.kerberos.domains=DOMAIN.ONE,DOMAIN.TWO

security.authentication.sso.kerberos.service.principal=HTTP/spn_irule_1@DOMAIN.ONE,HTTP/spn_irule_2@DOMAIN.TWO

security.authentication.sso.kerberos.keytab.location=C:\\irule-server\\irule-server-8080\\conf\\irule\kerberos\\spn_irule_1.keytab, C:\\irule-server\\irule-server-8080\\conf\\irule\kerberos\\spn_irule_2.keytab

security.authentication.sso.kerberos.debug=true
```

Здесь:

- `DOMAIN.ONE, DOMAIN.TWO` - имя доменов
- `HTTP/spn_irule_1@DOMAIN.ONE` - созданный **SPN**
- `C:\\irule-server\\irule-server-8080\\conf\\irule\kerberos\\spn_irule_1.keytab` - путь до **keytab**-файла

Аутентификация по **kerberos** может быть настроена для нескольких доменов. Для этого в свойствах **domains**, **service.principal** и **keytab.location** укажите соответствующие значения через запятую. Для каждого из доменов должен быть указан **SPN** и **keytab**.

Свойство **security.authentication.mode** может содержать несколько типов аутентификации.

Например:

```
security.authentication.mode=kerberos,jdbc
```

В данном случае клиенту будет доступен выбор между аутентификацией по **kerberos** и базовой.

Настройка подключения к KDC

Для работы **Kerberos** необходимо сконфигурировать подключение к **KDC** для **Java**, на которой работает сервер приложения.

Для этого откройте на редактирование файл **irule-server-8080\bin\setenv.bat** и раскомментируйте строку (уберите символ комментария `REM`)

```
REM -Djava.security.krb5.conf="%CATALINAHOME_JAVASEP%\\conf\\irule\\krb5.conf" ^
```

Откройте на редактирование файл **irule-server-8080\conf\irule\krb5.conf** с содержимым:

```
[libdefaults]
 allow_weak_crypto=true
 default_realm = DOMAIN.ONE
 forwardable = true
 clockskew = 3000
 default_tkt_enctypes = aes256-cts aes128-cts rc4-hmac
 default_tgs_enctypes = aes256-cts aes128-cts rc4-hmac
 permitted_enctypes  = aes256-cts aes128-cts rc4-hmac
 default_keytab_name=[keytab_name]
[realms]
 DOMAIN.ONE = {
  kdc = DOMAIN.ONE
  master_kdc = DOMAIN.ONE
  default_domain = DOMAIN.ONE
  }
 DOMAIN.TWO = {
  kdc = DOMAIN.TWO
  master_kdc = DOMAIN.TWO
  default_domain = DOMAIN.TWO
  }
[domain_realm]
 .DOMAIN.ONE = DOMAIN.ONE
 DOMAIN.ONE = DOMAIN.ONE
 .DOMAIN.TWO = DOMAIN.TWO
 DOMAIN.TWO = DOMAIN.TWO
```

Текущий **krb5.conf** сформирован для доменов **DOMAIN.ONE** и **DOMAIN.TWO**. Важно учитывать, что имплементация подключения по **Kerberos** в **Java API** требует использования в имени домена в верхнем регистре.

В качестве `[keytab_name]` указывается путь до **keytab**-файла.

Свойство **allow_weak_crypto** должно быть установлено в значение **true** в случае, когда на контроллере домена используются устаревшие алгоритмы шифрования.

Обратите внимание, в качестве **KDC** указано имя домена. Для доменов **Windows** функцию **KDC** выполняет контроллер домена. Один домен может содержать несколько контроллеров домена. В корректно настроенном домене по его имени будет возвращаться **ip** любого из доступных контроллеров домена. Для получения

всех возможных **ip** по доменному имени выполните в консоли **$ nslookup -all DOMAIN.ONE**.

Проверьте, что каждый из указанных **ip**-адресов доступен с сервера с помощью команды **ping**. Если в списке есть адреса, недоступные с сервера, возможна ситуация, когда во время аутентификации клиента будет выбран этот недоступный адрес **KDC**. О данной проблеме нужно сообщить администратору. В обычном случае подобная проблема быстро решается (например, остановили машину, **DNS** в кешах хранил ее **ip**). Но встречались ситуации, когда администратор не мог/не хотел исправлять проблему. В таком случае, в качестве **KDC** должен быть указан конкретный **ip** контроллера домена. И администратор должен понимать, что выключив эту машину, аутентификация с сервером **iRule** перестанет работать. Как промежуточное решение - администратору необходимо завести **DNS-запись**, которая будет содержать список доступных **ip** контроллеров домена для аутентификации по **kerberos** через **iRule**.

Тестирование аутентификации по KERBEROS

Удостовериться, что на сервере настроена связка **аутентификация по Kerberos, авторизация по метаобласти**.

В **irule-server-8080\conf\irule\irule-server.properties**:

```
security.authentication.mode=kerberos
security.authorization.mode=jdbc
```

Возможно использование несколько видов аутентификации, например базовая и **kerberos**:

```
security.authentication.mode=kerberos,jdbc
```

Трассировка процесса аутентификации

Выполните следующие действия:

1. Запустите клиент **iRule** с логами **System.out** с подробными шагами генерации ключа клиентским приложением.
2. Откройте директорию, в которую установлен клиент, например, **user\SPI\iRule\bin**.
3. Откройте на редактирование файл **irule-client-startup.bat**.

В конце файла строку

```
ru.spi2.MainLauncher %*
```

отредактируйте следующим образом:

```
ru.spi2.MainLauncher -lirule-client-system-out-err.log %*
```

4. Запустите отредактированный **irule-client-startup.bat**.
5. В директории с установленным клиентом, например, **user\SPI\iRule\bin\logs\irule-client-system-out-err.log** появится лог **System.out** и **System.err**.

Настройка авторизации через AD на сервере

Предварительные требования

Перед настройкой авторизации по **AD** должна работать связка **Аутентификация по Kerberos - Авторизация через метаобласть**. Сервер проверяет, что к нему подключается именно указанный доменный пользователь через **Kerberos**, а затем находит соответствующего пользователя в метаобласти. Такой порядок настройки необходим, чтобы исключить проблемы в конфигурации **Kerberos**, так как всегда перед авторизацией через **AD** будет шаг аутентификации через **Kerberos**.

Типы авторизации на сервере

Возможны следующие типы авторизации на сервере:

- базовая (через метаобласть)
- AD

Тип авторизации настраивается в **irule-server.properties**.

Сервер поддерживает одновременную работу нескольких видов авторизации:

- Настройки для авторизации только по **AD**:

```
security.authentication.mode=kerberos
security.authorization.mode=ad
```

- Настройки для авторизации только по **JDBC**:

```
security.authentication.mode=jdbc
security.authorization.mode=jdbc
```

- Настройки для авторизации и по **JDBC**, и по **AD**:

```
security.authentication.mode=kerberos,jdbc
security.authorization.mode=jdbc
security.authorization.mode.kerberos=ad
```

В этом случае при базовой авторизации, аутентификация и авторизация происходят в метаобласти. При авторизации через **AD** - аутентификация через **Kerberos**, авторизация через **AD**.

Порядок логина пользователя в систему при авторизации по AD

1. Выберите на клиенте действие **Подключиться к серверу**.
2. Происходит шаг аутентификации клиента с помощью **Kerberos**
3. Происходит шаг авторизации клиента с помощью **AD**. На этом шаге сервер подключается к **AD** и узнает, принадлежит ли пользователь одной из авторизованных групп.
4. Если доменный пользователь не существует в метаобласти, система создает пользователя в метаобласти.

Определяется серверным свойством, подробнее в пункте Автосоздание пользователя в метаобласти.

Настройки подключения сервера к AD

Настройки в **irule-server.properties**:

```
security.authorization.ad.url=ldap://[url]

security.authorization.ad.username=spn_user_irule_1@SPI.NEW

security.authorization.ad.password=[password]

security.authorization.ad.groups.connect=[group_name1],[group_name2]

security.authorization.ad.searchcontext=DC=SPI,DC=NEW
```

где:

- `security.authorization.ad.url` - **url** подключения к **AD** по **LDAP**
- `security.authorization.ad.username` - имя доменного пользователя, по которому сервер будет подключаться к **AD** для получения информации. Доменный пользователь для сервера создается на этапе конфигурации **Kerberos**
- `security.authorization.ad.password` - зашифрованный пароль доменного пользователя
- `security.authorization.ad.groups.connect` - имена доменных групп, которые авторизованны для входа в **iRule**
- `security.authorization.ad.searchcontext` - корневой контекст поиска в **AD**

Автосоздание пользователя в метаобласти

Для работы доменного пользователя в **iRule**, должен существовать соответствующий пользователь в метаобласти. Доступна функциональность автосоздания пользователя при авторизации через **AD**. Таким образом, администратор создает пользователя **AD** и добавляет его в авторизованную группу. Пользователь заходит в клиентское приложение **iRule**, подключается к серверу. Сервер авторизует его по информации из **AD**. Если такого пользователя не существует, то он будет создан.

Работать с системой под данным пользователем возможно только с помощью авторизации по **AD**.

Для активации автосоздания требуется проверить значение соответствующей серверной настройки в конфигурационном файле **irule-server.properties**:

```
security.authorization.ad.user.auto.create.enabled=true
```

Синхронизация прав доступа к моделям в соответствии с группами AD

Авторизация доступа к моделям разграничена на уровне групп **AD**. Синхронизация доступа к моделям происходит на этапе логина пользователя. Установка доступа выполняется для всех моделей метаобласти.

Для активации синхронизации доступа к моделям требуется проверить значение соответствующей серверной настройки в конфигурационном файле **irule-server.properties**:

```
security.authorization.ad.sync.group.permissions.for.user=true
```

Различаются виды доступа пользователя к моделям:

- Доступ отсутствует
- Доступ на получение данных (**execute**)
- Доступ на обновление модели (**create**, **update**, **delete**)

Пользователи групп, которым доступно получение данных из моделей, определяются серверным свойством:

```
security.authorization.ad.groups.read=[group_name1],[group_name2]
```

Группы, пользователям которых доступно изменение модели, определяются серверным свойством:

```
security.authorization.ad.groups.save=[group_name1]
```

Группа пользователей является объектом **AD**. Значение его свойства `cn` - имя группы, которое требуется указать в серверной настройке.

Формирование установщика клиента

Установка клиента возможна с помощью **exe-файла**. В конфигурационном файле **irule-server.properties** доступны следующие настройки:

- `system.deploy.generate.client.installer.distr` – если необходимо формирование **exe-файла** установщика клиента, установите это свойство в значение **true**
- `server.address.http.scheme` – установите это свойство в **http** или **https** в зависимости от используемой схемы подключения
- `server.address.http.host` – укажите IP-адрес или доменное имя компьютера, на котором уставлен сервер. Может быть вычислено автоматически, если в системе одно сетевое подключение
- `server.address.http.port` – укажите **http** порт

8.2. ЗАПУСК

После окончания настройки выполните следующие действия для запуска сервера:

1. Перейдите в папку **irule-server-8080\bin**.
2. Откройте на редактирование файл **deploy-war.bat**
3. Найдите строку:

```
REM Определить абсолютный путь до war-файла здесь
```

```
SET WAR_PATH="C:\spi\irule-RELEASE-2022.1.war"
```

4. Пропишите абсолютный путь до **war**-файла.
5. Сохраните файл **deploy-war.bat**.
6. Запустите файл **deploy-war.bat**
7. Проверьте, что сервер с новой версией серверной части **iRule** успешно запущен, перейдя по ссылке http://[host]:8080.

В случае возникновения проблем при установке клиента для нахождения причин их появления необходимо обратиться к журнальным файлам:

irule-server-8080/logs/catalina.log

irule-server-8080/logs/irule/irule-server.log

8.3. ПЕРЕЗАГРУЗКА

Для перезагрузки сервера выполните следующие действия:

1. Перейдите в папку **irule-server-8080**.
2. Выполните скрипт **irule-server-stop.bat**.
3. Сервер будет остановлен.

   Примечание. Иногда при остановке сервера приложения лицензия освобождается с задержкой и при попытке последующего запуска сервера приложения могут возникать ошибки. Проверить статус ключа можно на странице http://localhost:1947.

4. Выполните скрипт **irule-server-start.bat**.
5. Сервер будет запущен.

8.4. ОСТАНОВКА

Для остановки сервера выполните следующие действия:

1. Перейдите в папку **irule-server-8080**.
2. Выполните скрипт **irule-server-stop.bat**.
3. Сервер будет остановлен.

   Примечание. Иногда при остановке сервера приложения лицензия освобождается с задержкой и при попытке последующего запуска сервера приложения могут возникать ошибки. Проверить статус ключа можно на странице http://localhost:1947.

8.5. ОБНОВЛЕНИЕ

Для обновления версии сервера выполните следующие действия:

1. Поместите новую версию war-файла в папку **irule-server\irule-app**.
2. Перейдите в папку **irule-server-8080\bin**.
3. Откройте на редактирование файл **deploy-war.bat**

Примечание. Текущее состояние сервера не имеет значения, при необходимости скрипт автоматически определит необходимость и выполнит остановку сервера.

4. Найдите строку:

```
REM Определить абсолютный путь до war-файла здесь
 SET WAR_PATH="C:\spi\irule-RELEASE-2022.1.war"
```

5. Измените версию используемого **war**-файла клиента.
6. Сохраните файл **deploy-war.bat**.
7. Запустите файл **deploy-war.bat**
8. Проверьте, что сервер с новой версией серверной части **iRule** успешно запущен, перейдя по ссылке http://[host]:8080.

Обновление версий используемых приложений

Для просмотра текущих версий используемых приложений откройте файл **readme.txt**. Во время работы может потребоваться обновление версий используемых приложений до актуальных:

- Java - JDC сервера приложения. Посмотреть используемую версию можно в **irule-server-8080\bin\setenv.bat** в следующей строке:

```
SET JAVA_HOME=%CATALINA_HOME%\java\jdk-21.0.4-full
```

- Java - JRE клиентских приложений. Посмотреть используемую версию можно в **irule-server-8080\java\client-jre**.
- JDBC-драйвера. Посмотреть используемую версию можно в **irule-server-8080\lib\jdbc**.
- Tomcat. Если требуется обновление контейнера сервлетов, то рекомендуется остановить сервер, переименовать папку **irule-server-8080** в **irule-server-8080-old**, скопировать из нового дистрибутива папку **irule-server-8080**, в соответствии с инструкцией по установке внести новые параметры и провести полноценный процесс установки

Обновление версии Java (JDK и JRE)

Для смены **JDK** сервера приложения выполните следующие действия:

1. Переместите **JDK-дистрибутив** в каталог **JDK** аналогично ранее используемому дистрибутиву **Java**.
2. Измените файл **irule-server-8080\bin\setenv.bat**

`set JAVA_HOME=%CATALINA_HOME%\java\jdk-21.0.4-full` - укажите новую папку с **JDK**

Для обновления **JRE** клиентских приложений требуется поместить **JRE-дистрибутивы** в каталог **irule-server-8080\java\jre-client**.

Примечание. Если сервер был запущен, то для того чтобы изменения вступили в силу, необходимо перезапустить сервер приложения. Для этого выполните действия, описанные в подразделе Перезагрузка сервера приложения.

Следует обратить особое внимание, что сервер приложения и клиент **iRule** должны использовать **Java** одной версии. На уровне развертывания серверного приложения происходит проверка наличия дистрибутивов **JRE** для клиентского приложения **iRule**. В случае исключительной ситуации возможно отключение этой проверки с помощью следующей настройки в файле **irule-server.properties**, путём изменения значения с `true` на `false`:

```
system.deploy.check.client.jre.storage=false
```

Обновление JDBC-драйверов

JDBC-драйвера расположены в каталоге **irule-server-8080\lib\jdbc**.

При добавлении новой версии **JDBC-драйвера** важно удалить его предыдущую версию.

Примечание. Если сервер был запущен, то для того чтобы изменения вступили в силу, необходимо перезапустить сервер приложения. Для этого выполните действия, описанные в подразделе Перезагрузка сервера приложения.

8.6. УДАЛЕНИЕ

Для удаления сервера выполните следующие действия:

1. Перейдите в папку **irule-server-8080**.
2. Выполните скрипт **irule-server-stop.bat** для остановки сервера.
3. Выделите все используемые для работы сервера папки и удалите их.

9. СЕРВИС МОНИТОРИНГА

Сервис мониторинга выполняет функции обновления и позволяет следить за работоспособностью системы.

Для настройки автоматического старта сервера **iRule** при старте операционной системы используется приложение **iRule Watchdog**. Оно прописывается в качестве автоматически запускаемого сервиса операционной системы. После старта приложение **iRule Watchdog** запускает серверы приложений **iRule**, на которые оно настроено.

В операционной системе должно быть запущено только одно приложение **iRule Watchdog**, которое может быть настроено на работу с несколькими серверами приложений **iRule**.

9.1. УСТАНОВКА

Последовательность установки

Для установки и запуска приложения **iRule Watchdog** выполните следующие действия:

1. Настройте порт для запуска.
2. Добавьте сервер приложения в качестве проверяемого объекта.
3. Добавьте базу данных в качестве проверяемого объекта.
4. Настройте работу **Watchdog** как сервис операционной системы.
5. Запустите **Watchdog**.
6. Зашифруйте пароль.
7. Настройте доступ администратора.
8. Просмотрите логи.

Настройка порта для запуска

По умолчанию **iRule Watchdog** запускается по порту **9999**.

Для настройки порта, на котором работает **Watchdog**, выполните следующие действия:

1. Откройте на редактирование файл **irule-watchdog\conf\irule-watchdog\application.yaml**.
2. В открывшемся окне найдите следующие строки.

```
server:
 port: 9999
```

3. Измените значение в поле **port** и сохраните файл.

Добавление сервера приложения в качестве проверяемого объекта

Список проверки находится в файле **irule-watchdog\conf\irule-watchdog\configuration.yaml**.

В комментариях в этом файле описано, какую информацию необходимо добавить для отслеживания нового сервера приложения.

Примечание. По умолчанию список проверки **configuration.yaml** преднастроен на сервер приложения, запускаемый на порте **8080**. В этом случае достаточно запустить **Watchdog** и проверить его работоспособность.

```
hosts:
 - host: [host]
   agent: {type: local}
 checks:
 - #check
   alias: iRule
   type: http
   port: 8080
   start: ['..\\irule-server-8080\\bin\\startup-from-
watchdog.bat']
   stop: ['..\\irule-server-8080\\bin\\shutdown-from-
watchdog.bat']
```

При ручных правках файла **configuration.yaml** возможно появление ошибок в логах при старте приложения.

```
Caused by: org.yaml.snakeyaml.scanner.ScannerException: while
scanning for the next token

 found character '\t(TAB)' that cannot start any token. (Do not
use \t(TAB) for indentation)
```

Это означает, что файл, указанный в логе, содержит некорректный символ табуляции. Для устранения этой ошибки необходимо открыть файл в **Notepad++** и заменить все знаки табуляции на пробелы.

Добавление базы данных в качестве проверяемого объекта

Список проверки находится в файле **irule-watchdog\conf\irule-watchdog\configuration.yaml**.

В комментариях в этом файле описано, какую информацию необходимо добавить для проверки работоспособности базы данных.

```
#check
alias: [alias_name]
type: jdbc
url:  jdbc:oracle:thin:@localhost:[port]:dw
username: [user_name]
password: [password]
query: select * from dual
```

```
driver-class-name: oracle.jdbc.OracleDriver
delay: 45
enable: false
icon: [icon_name]
```

При ручных правках файла **configuration.yaml** возможно появление ошибок в логах при старте приложения.

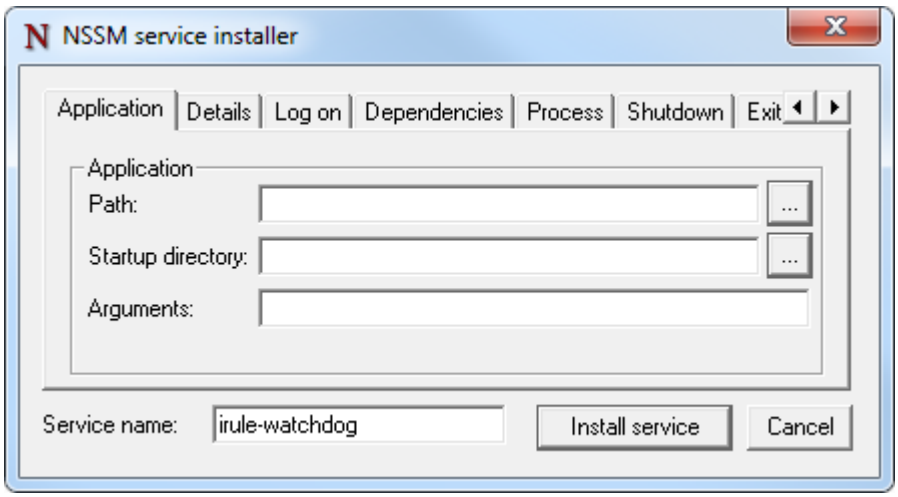
```
Caused by: org.yaml.snakeyaml.scanner.ScannerException: while
scanning for the next token

 found character '\t(TAB)' that cannot start any token. (Do not
use \t(TAB) for indentation)
```

Это означает, что файл, указанный в логе, содержит некорректный символ табуляции. Для устранения этой ошибки необходимо открыть файл в **Notepad++** и заменить все знаки табуляции на пробелы.

Настройка работы WatchDog как сервиса операционной системы

Для **Windows**:

1. **Watchdog** настраивается как сервис **Windows** с помощью приложения **NSSM**.
2. Запустите скрипт **irule-watchdog\install-as-service\setup-watchdog-service.bat**. Появится окно **NSSM service installer**.



3. В поле **Path** выберите путь до **irule-watchdog\start.bat**.
4. В поле **Service name** задается имя сервиса, предустановленное значение **irule-watchdog**.
5. Нажмите на кнопку **Install service**.
6. Зайдите в **Сервисы** (**Пуск > Выполнить > services.msc**).
7. Для добавленного сервиса **Watchdog** в свойствах **Вход в систему** проставьте учетную запись, из-под которой должен стартовать сервер приложения.

Если требуется запустить сервис, выполните следующие действия:

1. Зайдите в **Сервисы** (**Пуск > Выполнить > services.msc**).
2. Выберите службу **irule-watchdog**, в контекстном меню выберите действие **Запустить**.

Если требуется остановить сервис, выполните следующие действия:

1. Зайдите в **Сервисы (Пуск > Выполнить > services.msc)**.
2. Выберите службу **irule-watchdog**, в контекстном меню выберите действие **Остановить**.

Запуск и остановка

Для запуска **Watchdog** необходимо запустить скрипт **irule-watchdog\start.bat**

Для остановки **Watchdog** требуется закрыть окно приложения.

Если на данной машине **Watchdog** уже настроен как сервис операционной системы, запускать и останавливать его нужно как все другие сервисы. В **Windows** нужно зайти в панель **Службы** (**Пуск > Выполнить > services.msc**) и выполнить запуск/остановку сервиса.

После запуска для проверки работоспособности **Watchdog** выполните следующие действия:

1. Зайдите на страницу **Watchdog** http://[host]:9999/index.html.

Примечание. В случае изменения порта в конфигурации адрес будет отличаться.

2. В списке проверок будет текущий сервер **iRule**. При щелчке по названию осуществляется переход на страницу скачивания клиента **iRule**.
3. Выключите сервер приложения. Для этого выполните действия, описанные в подразделе Остановка сервера.

После выключения через некоторое время (в настройках по умолчанию до **45 с**) будет отображено, что сервер стал недоступен. Требуется проверить, что **Watchdog** запустит его самостоятельно.

Настройка доступа администратора

Для изменения состояния проверяемых объектов требуется быть авторизованным в приложении **Watchdog** как администратор. Для этого нужно зайти на страницу администратора: http://[host]:9999/admin.html.

Примечание. В случае изменения порта в конфигурации адрес будет отличаться.

Напротив объекта проверки будет доступен знак шестеренки со списком действий над объектом. Список действий зависит от статуса объекта. Например, Вы не можете запустить сервер, если он уже запущен.

При выполнении действия будут запрошены логин и пароль администратора.

Значения по умолчанию:

```
логин: admin
```

```
пароль: admin
```

Для изменения данных для авторизации требуется внести изменения в файл **irule-watchdog\conf\irule-watchdog\application.yaml**

```
management:
 context-path: /manage
 username: [user_name]
 password: [password]
```

В поле **password** записывается пароль, зашифрованный в соответствии с инструкцией, приведённой в разделе Шифрование паролей.

Просмотр журнала функционирования

Конфигурация журналирования находится в файле **irule-watchdog\conf\irule-watchdog\log4j2.xml**

Журнал функционирования находится в каталоге **irule-watchdog\logs\irule-watchdog**

9.2. ОБНОВЛЕНИЕ

После того, как произведена настройка **Watchdog**, любые изменения состояния сервера приложения (запуск, остановка, обновление приложения) должны производиться через веб-интерфейс **Watchdog**.

Страница **Watchdog** расположена по адресу: http://[host]:9999.

Примечание. В случае изменения порта в конфигурации адрес будет отличаться.

Действия на странице доступны авторизованным в приложении пользователям.

Значение по умолчанию:

```
логин: admin
```

```
пароль: admin
```

Установка приложения с помощью WatchDog

Выполните следующие действия:

1. Войдите на страницу **Watchdog**.
2. Для сервера **iRule** выберите действие **Установка новой версии приложения**.
3. На открывшейся странице **Установка новой версии приложения** нажмите на кнопку **Выбрать файл**.

4. Выберите **war-файл** приложения для установки, нажмите кнопку **ОК**.

После выполнения действий начнется загрузка файла на сервер и дальнейшая установка его на сервере приложения. Длительность загрузки файла зависит от скорости соединения между вашим компьютером и сервером. Следующий за этим процесс установки приложения (остановка сервера, очистка ресурсов, запуск сервера с нового приложения) занимает порядка **1-2 минут**. По истечению процесса установки приложения на сервер, вы будете переведены на страницу **Watchdog**.

В случае успешной установки для сервера **iRule** будет выставлен статус **Доступен**.

Перезагрузка сервера приложения с помощью WatchDog

Выполните следующие действия:

1. Войдите на страницу **Watchdog**.
2. Выберите действие **Рестарт сервера**.

Запуск сервера приложения с помощью WatchDog

По умолчанию настроено автовосстановление сервера приложения. В этом случае **Watchdog** должен самостоятельно запустить сервер приложения.

Если автовосстановление выключено (на странице **Watchdog** напротив **iRule Сервер** нет значка **auto**), выполните следующие действия:

1. Войдите на страницу **Watchdog**.
2. Для сервера **iRule** выберите действие **Запустить сервер**.

Остановка сервера приложения с помощью WatchDog

Выполните следующие действия:

1. Войдите на страницу **Watchdog**.
2. Для сервера **iRule** выберите действие **Выключить автовосстановление**.
3. Выберите действие **Остановить сервер**.

Примечание. С помощью **Watchdog** может быть остановлен любой сервер, в том числе запущенный вручную.

9.3. УДАЛЕНИЕ

Для удаления **iRule Watchdog** выполните следующие действия:

1. Остановите сервер.
2. Выделите все используемые для работы сервера **iRule Watchdog** папки и удалите их.

10. НАСТРОЙКА ОБРАЩЕНИЯ К ВНЕШНИМ ИНФОРМАЦИОННЫМ РЕСУРСАМ

Доступ к внешним информационным ресурсам может быть осуществлён через **Сервер взаимодействия** или через прокси-сервер.

Настройка доступа ко внешним сервисам через прокси-сервер

Для настройки доступа к внешним сервисам через прокси-сервер выполните следующие действия:

1. Откройте на редактирование файл **irule-server-8080\bin\setenv.bat**
2. Допишите строку `-Dhttp.proxyHost=[proxy_host] -Dhttp.proxyPort=[proxy_port]` после строки:

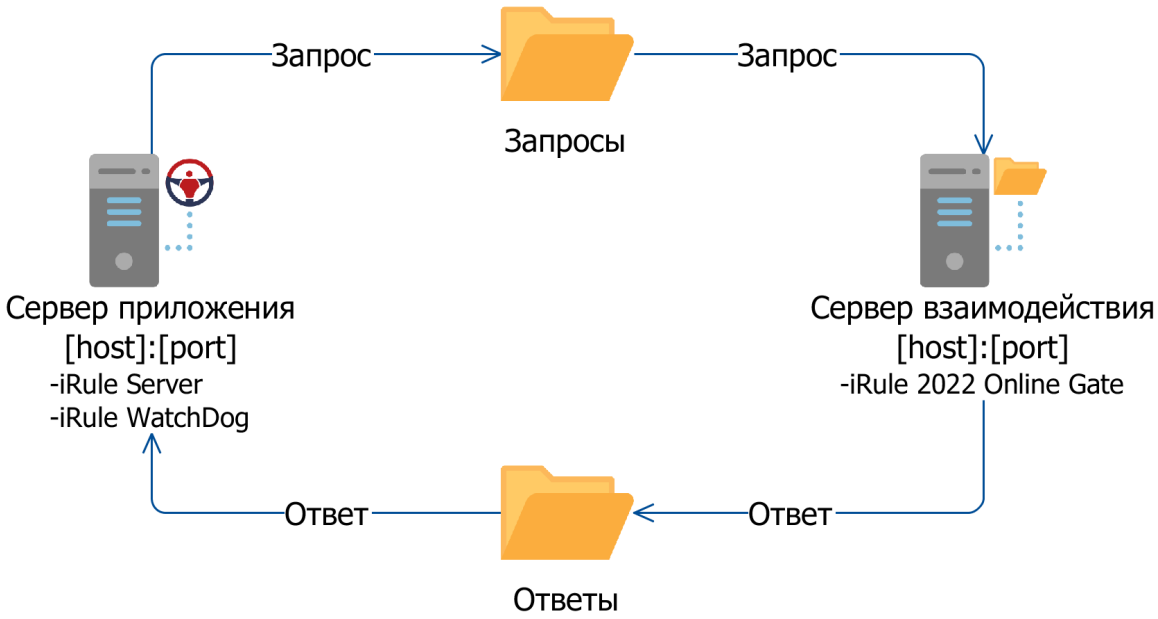
```
set "JAVA_OPTS=%JAVA_OPTS% ^ -Dirule.server.work.dir=..."
```

3. В качестве значений параметров `[proxy_host]` и `[proxy_port]` выберите параметры подключения к интернету через прокси: адрес прокси-сервера (**ip** или имя машины) и порт подключения.

Настройка сервиса взаимодействия

Сервис взаимодействия предназначен для организации вызовов внешних сервисов (например, **СПАРК**) в условиях, когда машине, на которой запущен сервер приложения **iRule**, закрыт доступ к внешней сети.

В этом режиме сервер приложения **iRule** отправляет запросы к внешним сервисам с помощью создания запросного файла в папке запросов, сервер взаимодействия вычитывает их, отправляет соответствующий запрос внешнему сервису, а его ответ в виде файла помещает в папку ответов, где он вычитывается и обрабатывается сервером приложения **iRule**.



Для настройки такого режима работы выполните следующие действия:

1. Запустите на машине, имеющей доступ к внешним сервисам, сервер взаимодействия.
2. Создайте на машине, доступной по сети и серверу приложения **iRule** и серверу взаимодействия **iRule Online Gate** две папки – папку запросов и папку ответов.
3. Предоставьте серверу приложения **iRule** право записи в папку запросов и право чтения из папки ответов.
4. Предоставьте серверу взаимодействия **iRule Online Gate** право записи в папку ответов и право чтения из папки запросов.
5. Откройте на редактирование файл **irule-server-8080\conf\irule\irule-server.properties** настроек сервера приложения **iRule** и внесите следующие изменения:

```
external.web.services.invocation.mode=gate
external.web.services.gate.executors.count=4
external.web.services.gate.execution.timeout.sec=180
external.web.services.gate.request.dir=sharedserver://spi//request
external.web.services.gate.response.dir=sharedserver://spi//respon
se
```

6. Откройте на редактирование файл **irule-server-8080\irule-gate-2080\conf\irule-gate\ irule-ws-gate.properties** настроек сервера взаимодействия **iRule Online Gate** и внесите следующие изменения:

```
external.web.services.gate.request.dir=sharedserver://spi//request
external.web.services.gate.response.dir=sharedserver://spi//respon
se
```

7. Запустите сервер взаимодействия **iRule Online Gate** и сервер приложения **iRule**.

11. КЛИЕНТ

Установка, запуск, обновление и удаление клиента **iRule**.

11.1.УСТАНОВКА

Настройка клиентских машин на работу по HTTPS

Для возможности подключения клиента к серверу по протоколу **HTTPS**, клиент должен аутентифицировать сервер. **HTTPS**-клиент загружает с сервера сертификат и анализирует его данные. Если клиент подключается к машине host1, а сертификат выписан на host2, сертификат аутентификацию не проходит. Если организация, выписавшая сертификат, отсутствует в списке доверенных у клиента, клиент анализирует, кто выдал сертификат данной организации. Эта проверка может повториться множество раз, пока один из родительских сертификатов не окажется доверенным, или пока не дойдет до корневого сертификата (**CA**). Если корневой сертификат отсутствует в списке доверенных - **HTTPS** соединение не прошло аутентификацию.

В нашем случае сертификат самоподписанный - подписавшая его организация не является **CA** (**Certificate Authority**) и нет никаких родительских сертификатов.

Единственный вариант, при котором **HTTPS-соединение** с таким сертификатом будет аутентифицировано - если сертификат будет в списке доверенных на клиенте.

Для добавления сертификата всем пользователям как доверенного используется **Active Directory**. Перед внесением соответствующих настроек в **AD** следует быть уверенным, что подключение по **HTTPS** выполнится для тестового пользователя.

Предлагается следующий порядок настройки клиента:

1. Получить публичную часть сертификата сервера в виде файла.
2. Добавить загруженную публичную часть сертификата как доверенный сертификат на уровне профиля тестовому пользователю.
3. Проверить подключение клиента по **HTTPS** под тестовым пользователем.
4. Добавить загруженную публичную часть сертификата как доверенный сертификат всем пользователям домена с помощью **Active Directory**.
5. Проверить подключение клиента по **HTTPS** под пользователем домена.

Получение публичной части сертификата сервера в виде файла

Для получения публичной части сертификата в браузере зайдите на стартовую страницу сервера https://[host]:8443.

В **Internet Explorer**:

1. В адресной строке нажмите на значок замка.

2. Во всплывающем окне **Идентификация веб-сайтов** выберите **Просмотр сертификатов**.
3. В окне **Сертификат** выберите вкладку **Состав**.
4. Нажмите на кнопку **Копировать в файл**.
5. В мастере экспорта сертификатов выберите формата файла **Файлы X.509 (.CER) в кодировке DER**.
6. Выберите путь, куда сохранить публичный сертификат.

Это не путь к файлу-хранилищу приватной и публичной части сертификата, сейчас вы сохраняете публичную часть сертификата к себе на машину в виде файла, чтобы потом добавить его в **AD** как доверенный всем пользователям.

В **Firefox**:

1. В адресной строке нажмите на значок замка.
2. Во всплывающем окне выберите **Подробнее**.
3. В окне информация о странице выберите вкладку **Защита**.
4. Нажмите на кнопку **Просмотреть сертификат**.
5. В окне **Просмотр сертификата** выберите вкладку **Подробности**.
6. Нажмите на кнопку **Экспортировать...**
7. Выберите путь, куда сохранить публичный сертификат, тип файла **Сертификат X.509 в формате PEM**.

Это не путь к файлу-хранилищу приватной и публичной части сертификата, сейчас вы сохраняете публичную часть сертификата к себе на машину в виде файла, чтобы потом добавить его в AD как доверенный всем пользователям.

Первоначальная проверка возможности подключения клиента к серверу по настроенному сертификату

1. Выберите **Пуск > Свойства браузера**.
2. Откройте вкладку **Содержание**, выберите **Сертификаты**.
3. В окне **Сертификаты** выберите вкладку **Доверенные корневые центры сертификации**.
4. Выберите **Импорт**, путь до **.pem**-сертификата, далее по стандартным настройкам.
5. Проверьте, что сертификат появился в списке на вкладке **Доверенные корневые центры сертификации**.
6. Зайдите из браузера на страницу http://[host]:8443, проверьте, считает ли браузер сертификат доверенным.

7. Запустите клиент на этой же машине и проверьте, что он подключился к серверу успешно.

Настройка подключения по HTTPS для всех пользователей домена

Добавление сертификата в AD как доверенного для всех пользователе домена

1. Передайте на машину **AD** файл с публичным сертификатом, полученным в пункте **Получение публичной части сертификата сервера в виде файла**.
2. Откройте **mmc**.
3. Выберите пункт **Файл > Добавить оснастку > Редактор управления групповыми политиками**.
4. В диалоговом окне выберите **Политика Default Domain Policy**.
5. В добавленной оснастке перейдите по дереву **Политика Default Domain Policy > Конфигурация компьютера > Политики > Конфигурация Windows > Параметры безопасности > Политики открытого ключа > Доверенные корневые центры сертификации**.
6. В окне элементов вызовите контекстное меню, выберите **Импорт...**
7. Выберите файл с публичной частью сертификата и выполните импорт.

Проверка подключения пользователя домена к серверу

Выполните следующие действия:

1. Перезагрузите машину пользователя.
2. Зайдите пользователем под своей учетной записью в операционную систему.
3. Проверьте, что в списке доверенных сертификатов хранилища **Windows** добавленный сертификат присутствует.
4. Зайдите на стартовую страницу сервера https://[host]:8443 из браузера и удостоверьтесь, что страница считается доверенной.

Для того чтобы проверить, что у пользователя домена сертификат сервера был добавлен в хранилище доверенных сертификатов **Windows**, выполните следующие действия:

1. Выберите **Пуск > Свойства браузера**.
2. В окне **Свойство:Интернет** откройте вкладку **Содержание**.
3. Нажмите на кнопку **Сертификаты**.
4. В окне **Сертификаты** перейдите на вкладку **Доверенные корневые центры сертификации**.
5. В списке должен быть сертификат сервера, в поле **Кому выдан** - доменное имя сервера.

Установка

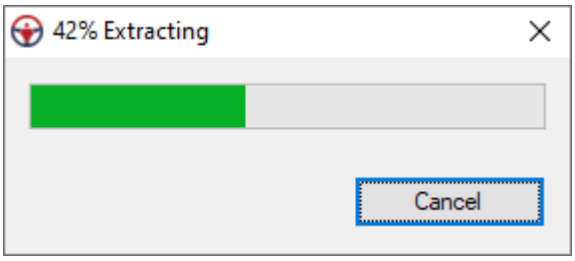
1. Запустите веб-браузер.
2. В адресную строку введите: **http://[host]:[port]**, где:

- **[host]** – IP-адрес или доменное имя компьютера, на котором установлен **iRule Server**
- **[port]** – порт, на котором запущено приложение. Если равен 80, то параметр вместе с предшествующим двоеточием может быть опущен

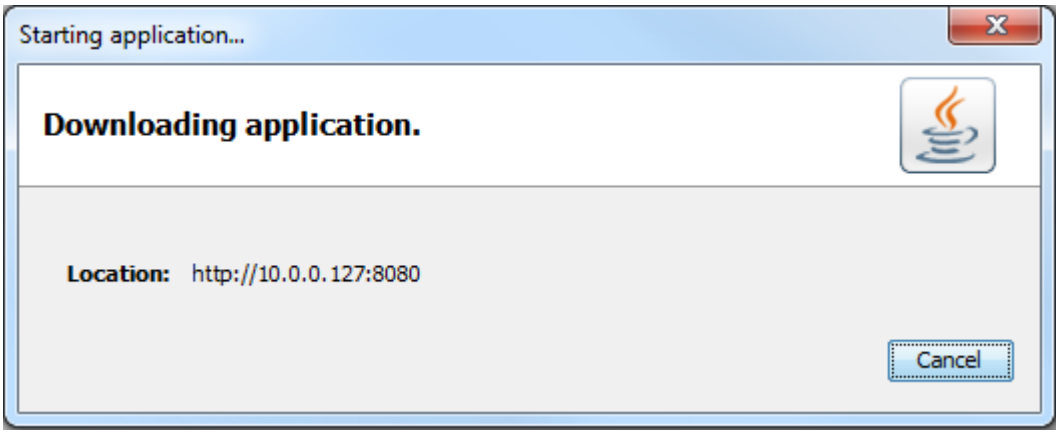
3. Возможна установка с использованием **exe-файла** и **jnlp-файла**. На открывшейся странице нажмите соответственно **Клиент (exe)** или **Клиент (jnlp)**.

   Примечание. При установке через **jnlp-файл** в адресную строку может быть введено **http://[host]:[port]/irule-client.jnlp**, в этом случае скачивание файла начнётся автоматически.

4. Для установки с помощью **exe-файла** выполните следующие действия:
   a. Запустите скачанный файл **irule_client_installer.exe** на выполнение.
   b. Дождитесь окончания процесса распаковки файлов, ход которого отображается в появившемся диалоговом окне.

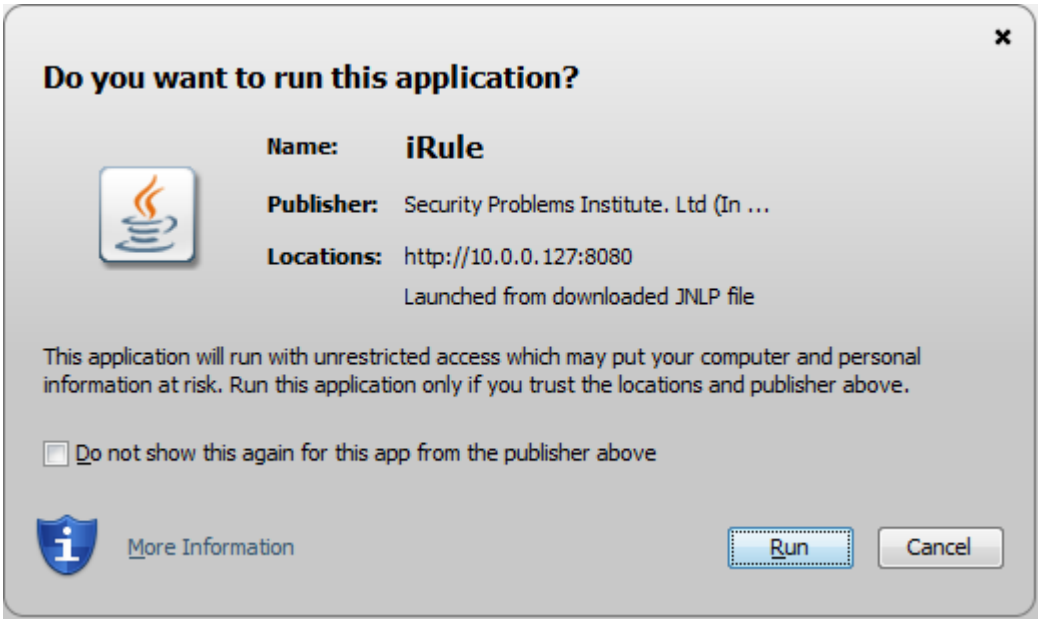


5. Для установки с помощью **jnlp-файла** выполните следующие действия:
   a. Подтвердите копирование файла **irule-client.jnlp** на компьютер и запустите на выполнение.



   b. Для работы необходима **JRE** версии **1.8 update 131** и старше, в ином случае установка будет прервана.

c. Подлинность продукта **iRule** подтверждена действительным сертификатом, выпущенным доверенным центром сертификации. В диалоговом окне указаны имя издателя, название и расположение продукта. Для отображения дополнительной информации нажмите **More Information**.



d. Нажмите кнопку **Run**.

6. Будет открыто диалоговое окно **Установка iRule**.
7. Для установки приложения необходимо задать следующие параметры:

- Выбрать **Тип установки**:
  - **Локальная** – приложение будет развёрнуто непосредственно на компьютере пользователя. При выборе опции **Для всех пользователей компьютера** запуск приложения сможет выполнить любой пользователь (не только осуществивший установку), которому будет разрешён доступ к папке приложения
  - **Сетевая** – приложение будет развёрнуто на сетевом ресурсе локальной сети. Пользователи, которым будет разрешён доступ к данному ресурсу, смогут запустить приложение, не устанавливая его на свой персональный компьютер
- **Папка для установки приложения** – задайте папку, где необходимо развернуть приложение
- **Папка для хранения пользовательских данных** – задайте папку, где необходимо хранить пользовательские данные. В случае сетевой или многопользовательской локальной установки в качестве элементов пути можно использовать следующие переменные:
  - **%USER_HOME%** – каталог пользователя
  - **%USER_DOCUMENTS%** – папка, где хранятся документы пользователя
  - **%USER_NAME%** – имя пользователя

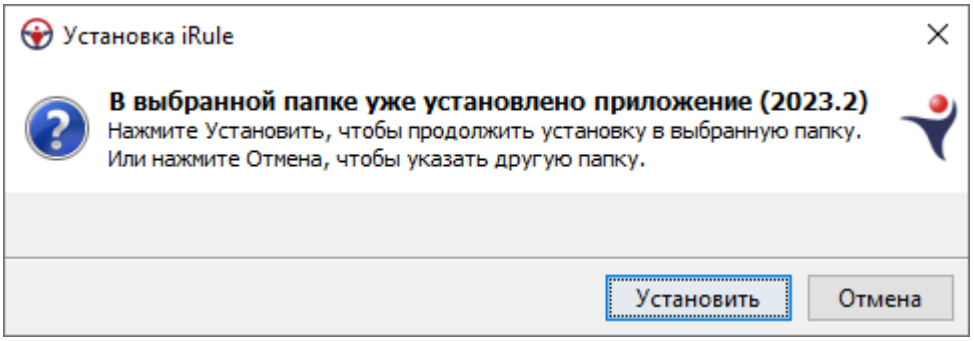
- **Папка для хранения пользовательских настроек** – задайте папку, где необходимо хранить пользовательские настройки. В случае сетевой или многопользовательской локальной установки в качестве элементов пути можно использовать следующие переменные:
  - **%USER_HOME%** – каталог пользователя
  - **%USER_DOCUMENTS%** – папка, где хранятся документы пользователя
  - **%USER_NAME%** – имя пользователя
- Указать **параметры запуска** приложения
  - **Создать ярлык на рабочем столе** – установка флажка позволит поместить ярлык на рабочий стол для быстрого доступа к приложению. Если установлен флажок **Для всех пользователей компьютера**, то ярлык будет помещён на рабочие столы всех пользователей
    - **Изменить имя ярлыка** – если необходимо, измените автоматически формируемое имя ярлыка
  - **Автоматически запускать обновление приложения при старте** – установка флажка позволит осуществлять автоматическую проверку соответствия версии приложения с версией, находящейся на сервере, и, в случае необходимости, обновление до актуальной версии
- Выбрать **разрядность версии**. При **Локальной** установке выбор будет доступен только на 64-х разрядных операционных системах. При **Сетевой** установке могут быть выбраны оба варианта, а используемая разрядность на персональном компьютере будет зависеть от технических характеристик компьютера пользователя
- Определить **Объем оперативной памяти**. Объём оперативной памяти зависит от разрядности версии приложения. По умолчанию предлагается **1024 Мб** для 32-разрядной версии и **2048 Мб** для 64-разрядной версии. Можно изменить значение, однако необходимо учесть, что при старте приложение не может проверить наличия заданного объёма памяти, и в случае, если операционная система не сможет предоставить запрошенный объём оперативной памяти, приложение не запустится

  Примечание. Изменение объёма выделяемой оперативной памяти возможно после установки приложения. Для этого в папке **bin** каталога, в котором установлено приложение, в файле `irule-client-init-vars-ext.bat` (для ОС **Windows**) или `irule-client-init-vars-ext.sh` (для ОС **Linux**) измените значение переменной `IRULE_XMX`. Изменения вступят в силу после перезагрузки приложения. Данный файл не обновляется при обновлении приложения.
- Для установки драйвера лицензионного ключа установите флажок **Установить драйвер лицензионного ключа**. Проверка лицензии может быть осуществлена на сервере или на клиенте. Если устанавливаемая версия **iRule**

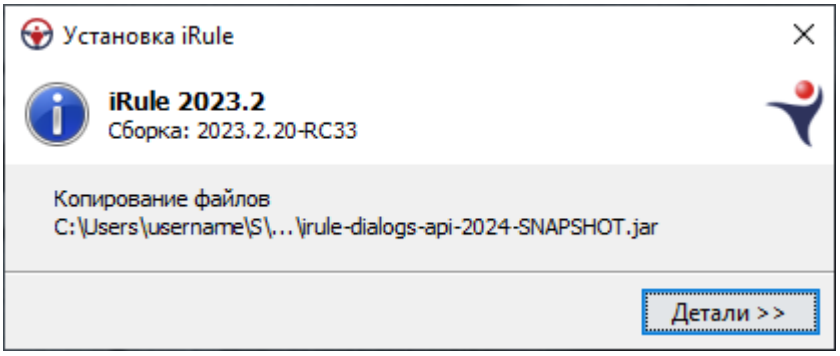
предполагает проверку лицензии на клиенте, то потребуется установка драйвера лицензионного ключа

Примечание. Для установки драйвера лицензионного ключа потребуются права администратора.

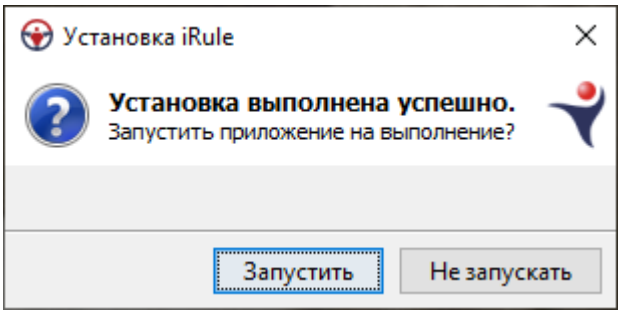
8. Нажмите кнопку **Установить**.
9. В случае если в указанной папке уже установлено приложение, появится соответствующее окно **Установка iRule**. При нажатии кнопки **Установить** приложение будет обновлено с потерей отдельных пользовательских файлов.



10. Ход установки приложения отображается в окне **Установка iRule**.



11. В случае успешной установки появится диалоговое окно **Установка iRule**. Нажмите кнопку **Запустить**, чтобы открыть приложение.



Запуск

1. Запуск **iRule** можно выполнить, используя:

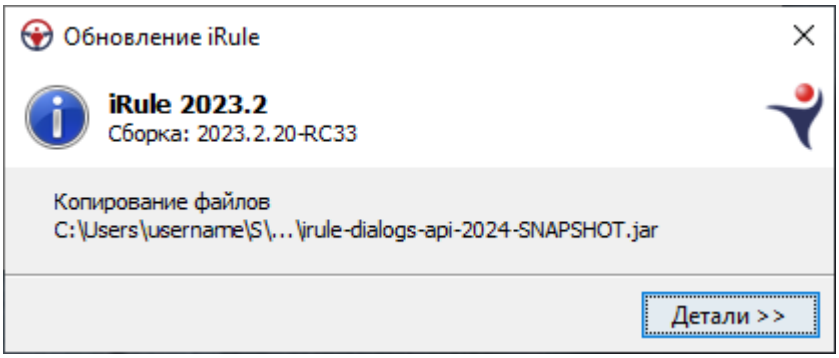
- Веб-браузер. Для этого:
    1. Запустите веб-браузер.

2. В адресную строку введите: **http://[host]:[port]/irule-client.jnlp**, где:
   - **[host]** – IP-адрес или доменное имя компьютера, на котором установлен **iRule Server**
   - **[port]** – порт, на котором запущено приложение. Если равен 80, то параметр вместе с предшествующим двоеточием может быть опущен
3. Подтвердите копирование файла **irule-client.jnlp** на компьютер и запустите на выполнение.

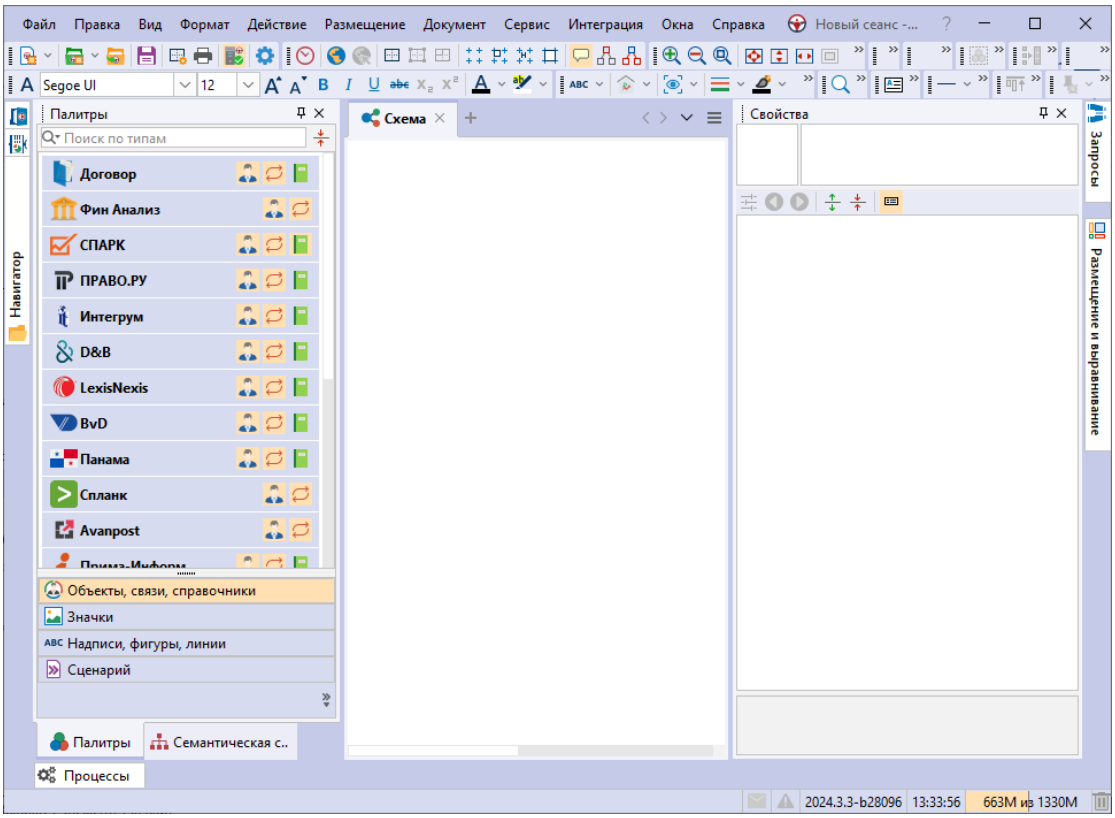
- Ярлык на рабочем столе. Дважды щелкните по ярлыку на рабочем столе:

2. Через некоторое время появится заставка:

3. Во время запуска происходит проверка и при необходимости обновление установленной версии **iRule**.



4. Будет открыто главное окно **iRule** и создан новый сеанс на основе стандартного шаблона.



Примечание.

1. Размещение окон может отличаться от приведенного.
2. Если сеанс не был создан (на вкладке **Палитры** нет ни одной палитры), необходимо выполнить настройку шаблона по умолчанию.

5. После отображения главного окна будут последовательно открыты диалоги:

- **Примечания к выпуску**
- **Совет дня**
- **Подключение к серверу**

Примечание. Все или некоторые из указанных диалогов не будут отображены, если во время предыдущего запуска был снят флажок **Показывать при запуске**.

Установка клиента без использования Java Web Start

Иногда при установке клиента с помощью ссылки в браузере (технология **Java Web Start**) возникают проблемы. В этом случае рекомендуется устанавливать клиент с помощью отдельного инсталлятора клиента.

Формирование инсталлятора клиента

Выполните следующие действия:

1. Распакуйте архиватором 7-zip war-файл приложения. Для этого в контекстном меню **war-файла** выберите **7-Zip > Распаковать в irule-2024.1** (имя папки формируется по имени war-файла). Если на машине отсутствует 7-zip, установите его из дистрибутива **\irule-server\utils\7-zip**.

2. Запустите скрипт формирования инсталлятора клиента.
3. Перейдите в созданную архиватором папку. Запустите **prepare-client-installer-no-jnlp.bat**
4. Инсталлятор будет сформирован в папке **client-installer** на том же уровне, где лежит **war-файл** приложения. Об этом будет сообщено в окне скрипта по окончанию создания инсталлятора:

```
Инсталлятор клиента iRule сформирован и находится в C:\spi\client-installer
 Для продолжения нажмите любую клавишу . . .
```

Установка клиента

Выполните следующие действия:

1. Перейдите в каталог инсталлятора клиента: **C:\spi\client-installer**.
2. Откройте на редактирование файл **start-client-no-jnlp.bat**
3. Введите адрес подключения (адрес и порт сервера):

```
REM Адрес сервера
 SET SERVER_HOST=[host]
 REM Порт сервера
 SET SERVER_PORT_HTTP=[port]
```

4. Проверьте, что сервер доступен по указанному адресу. Для этого в браузере перейдите по адресу **http://[host]:[port]**, где [**host**] - указанное значение **SERVER_HOST**, [**port**] - указанное значение **SERVER_PORT_HTTP**.
5. Запустите **start-client-no-jnlp.bat**

Изменение языка

Для изменения языка клиента **iRule** на английский выполните следующие действия:

1. Откройте каталог, в котором установлен клиент.
2. Откройте на редактирование файл **bin\irule-client-init-vars-ext.bat**.
3. Найдите следующие строки:

```
set IRULE_LANGUAGE=ru
set IRULE_REGION=RU
```

4. Измените значения указанных строк на следующие:

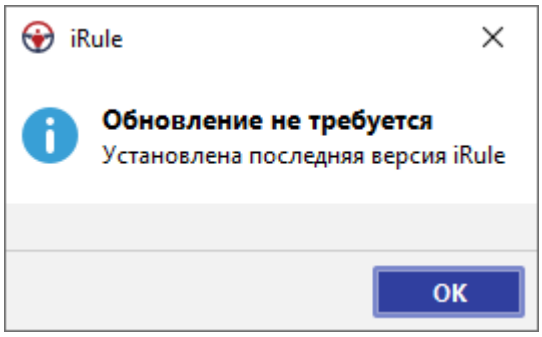
```
set IRULE_LANGUAGE=en
set IRULE_REGION=US
```

5. Сохраните файл и запустите клиент.

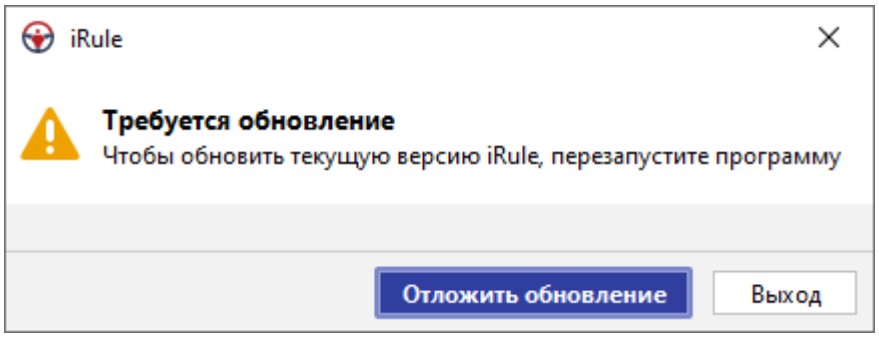
11.2.ОБНОВЛЕНИЕ

Для проверки обновлений **iRule** выберите пункт меню **Справка > Проверить обновления**.

Откроется диалоговое окно, содержащее информацию об актуальности версии продукта:



Если установлена неактуальная версия клиента, при подключении к серверу появится диалоговое окно:



Примечание. Данное диалоговое окно также появится при повторном подключении к серверу в случае разрыва соединения из-за обновления версии сервера во время работы с **iRule**.

Сохраните текущий сеанс и перезапустите программу.

11.3.УДАЛЕНИЕ

Для удаления клиента выделите и удалите следующие папки:

- Папка, в которой установлено приложение. По умолчанию она находится в **C:\Users\username\SPI\iRule**
- Папка для хранения пользовательских данных. По умолчанию она находится в **C:\Users\username\Documents\iRule**
- Папка для хранения пользовательских настроек. По умолчанию она находится в **C:\Users\username\SPI\iRule**