RunaWFE. Клиент-оповещатель о поступивших задачах. Руководство администратора.

Версия 3.0

Настройка rtn на RunaWFE сервер

Клиент оповещатель по умолчанию настроен на работу с RunaWFE сервером, расположенном на компьютере с именем wfe_server. Настроить клиент на другой сервер можно следующими способами:

1. Необходимо сопоставить на клиентских машинах (с клиентом-оповещателем) ip адрес RunaWFE сервера имени wfe_server. В Windows системах это возможно сделать, прописав в C:\WINDOWS\system32\drivers\etc\hosts строку «ip_addr wfe_server». В linux системах аналогичную строку необходимо прописать в файле /etc/hosts.
2. Перенастроить клиент-оповещатель на RunaWFE сервер, заменив в файлах af_delegate.properties и application.properties wfe_server на ip адрес или имя сервера. В случае, если RunaWFE сервер использует порты, отличные от портов по умолчанию (1099 и 8080), то необходимо заменить порты на актуальные.

Аутентификация

Выбор типа аутентификации

Аутентификация в клиенте-оповещателе состоит из двух частей:

- Аутентификация, проводимая по RMI. Используется клиентом для получения информации о заданиях пользователя и отображении на основе этой информации значка в трее и всплывающих сообщений при приходе новых задач
- Аутентификация, проводимая во встроенном web браузере. Необходима для корреткной работы web интерфейса системы RunaWFE.

Для аутентификации по RMI доступно 2 механизма аутентификации: с использованием логина и пароля пользователя в системе RunaWFE и аутентификация по протоколу kerberos. Тип используемой аутентификации определяется параметром authentication.type в файле application.properties. Для использования аутентификации с использованием логина и пароля необходимо установить параметр в «userinput», а для использования аутентификации по протоколу kerberos установить «kerberos». При аутентификации по протоколу kerberos клиент-оповещатель аутентифицируется без дополнительной информации со стороны пользователя системы. При аутентификации с использованием логина и пароля пользователю будет предложено ввести имя и пароль.

Тип аутентификации во встроенном web браузере задается параметром login.relative.url в файле application.properties в виде url, относительно адреса web интерфейса RunaWFE. Для аутентификации во встроенном web браузере доступно 3 url:

- /login.do — аутентификация с использованием логина и пароля пользователя. Для корреткной работы требует, что-бы RMI аутентификация так-же использовала аутентификацию по имени и паролю.
- /ntlmlogin.do — аутентификация с использованием протокола ntlm.

- /krblogin.do — аутентификация по протоколу kerberos.

Настройка аутентификации с использованием имени и пароля пользователя

При использовании аутентификации по имени и паролю клиент-оповещатель при старте попросит пользователя ввести имя и пароль. Параметры userinput.default.login и userinput.default.password в файле application.properties определяют имя и пароль, отображаемые по умолчанию в диалоге пользователя.

Настройка Kerberos аутентификации

Замечание. В данном разделе во всех именах и принципалах пользователей и серверов нужно учитывать, что заглавные и прописные буквы разнятся.

Настройка клиентской части

Последовательность действий:

1. Внести в следующий ключ реестра параметр

- Для Windows Server 2003 и Windows 2000 SP4

ключ: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters

параметр: allowtgtsessionkey=dword:0x01

- Для Windows XP SP2

ключ: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos

параметр: allowtgtsessionkey=dword:0x01

Замечание. После внесения параметра необходима перезагрузка.

Описание проблемы, которая решается при помощи данного действия: http://java.sun.com/j2se/1.5.0/docs/ guide/ security/ jgss/ tutorials/ Troubleshooting. html глава "javax.security.auth.login.LoginException: KrbException: KDC has no support for encryption type (14) - KDC has no support for encryption type".

2. Создать/отредактировать файл конфигурации Kerberos krb5.ini

Файл должен находиться в %SystemRoot% и иметь имя krb5.ini.

Обязательно следует указать в качестве алгоритмов шифрования следующие:

[libdefaults]

default_tkt_enctypes = des-cbc-md5 des-cbc-crc des3-cbc-sha1

default_tgs_enctypes = des-cbc-md5 des-cbc-crc des3-cbc-sha1

permitted_enctypes = des-cbc-md5 des-cbc-crc des3-cbc-sha1

Детальное описание файла конфигурации Kerberos http://web.mit.edu/kerberos/www/krb5-1.4/krb5-1.4.3/ doc/krb5-admin/krb5.conf.html.(Пример конфигурационного файла krb5.ini прилагается.)

3. Проинсталлировать на клиенте JRE5.0.10+, (находится http://java.sun.com/j2se/1.5.0/download.jsp)

4. После настройки серверной части клиентское приложение можно будет активизировать, запустив на выполнение файл $(NOTIFIER_ROOT)\run.exe (run.sh).

Настройка серверной части

См. подробное описание настройки RunaWFE сервера в документе WF-system_Installation_guide_ru. Обратите внимание, что для корректной аутентификации клиента-оповещателя по RMI с использованием протокола kerberos, необходимо в файле kerberos_module.properties, находящемся в папке rtn установить тот же serverPrincipal, что и на RunaWFE сервере (Одноименное свойство в kerberos_module.properties находящемся в jboss-root/server/default/conf).

Настройка безопасности JVM

Включить security manager. Для включения security manager'a для всех локально запускаемых приложений необходимо определить переменную окружения _JAVA_OPTIONS и установить ее значение -Djava.security.manager

После этого на все локально запускаемые приложения будут накладываться ограничения security manager'a по умолчанию. Эти ограничения описываются в файле $JAVA_HOME\lib\security\java.policy.

Формат этого файла описан http://java.sun.com/j2se/1.5.0/docs/guide/security/PolicyFiles.html. Список возможных полномочий используемых security manager'ом – http://java.sun.com/j2se/1.5.0/docs/guide/security/permissions.html.

Пример политики (файла java.policy) дающий классам (в том числе и из JAR архивов) из директории D:\tmp все полномочия и не дающим никаких полномочий классам (в том числе и из JAR архивов) из любых других директорий файловой системы:

```
// Standard extensions get all permissions by default
grant codeBase "file:${{java.ext.dirs}}/*" {
permission java.security.AllPermission;
};
// default permissions granted to all domains
grant {
// Allows any thread to stop itself using the java.lang.Thread.stop()
// method that takes no argument.
// Note that this permission is granted by default only to remain
// backwards compatible.
// It is strongly recommended that you either remove this permission
// from this policy file or further restrict it to code sources
// that you specify, because Thread.stop() is potentially unsafe.
// See "http://java.sun.com/notes" for more information.
permission java.lang.RuntimePermission "stopThread";
// allows anyone to listen on un-privileged ports
permission java.net.SocketPermission "localhost:1024-", "listen";
// "standard" properies that can be read by anyone
permission java.util.PropertyPermission "java.version", "read";
permission java.util.PropertyPermission "java.vendor", "read";
permission java.util.PropertyPermission "java.vendor.url", "read";
permission java.util.PropertyPermission "java.class.version", "read";
```

permission java.util.PropertyPermission "os.name", "read";

permission java.util.PropertyPermission "os.version", "read";

permission java.util.PropertyPermission "os.arch", "read";

permission java.util.PropertyPermission "file.separator", "read";

permission java.util.PropertyPermission "path.separator", "read";

permission java.util.PropertyPermission "line.separator", "read";

permission java.util.PropertyPermission "java.specification.version", "read";

permission java.util.PropertyPermission "java.specification.vendor", "read";

permission java.util.PropertyPermission "java.specification.name", "read";

permission java.util.PropertyPermission "java.vm.specification.version", "read";

permission java.util.PropertyPermission "java.vm.specification.vendor", "read";

permission java.util.PropertyPermission "java.vm.specification.name", "read";

permission java.util.PropertyPermission "java.vm.version", "read";

permission java.util.PropertyPermission "java.vm.vendor", "read";

permission java.util.PropertyPermission "java.vm.name", "read";

};

grant codeBase "file:/D:/tmp/*" {

permission java.security.AllPermission;

};

Как запустить клиент-оповещатель

Установите в af_delegate.properties ссылку на RunaWFE сервер.

Установите swt-win32-3232.dll в директорию, ссылка на которую находится в переменной окружения Path.

Запустите javaw -cp .;rtn.jar ru.runa.notifier.PlatformLoader

javaw -cp .;rtn.jar ru.runa.notifier.PlatformLoader

Замечание. Можно не использовать переменную окружения Path. В этом случае положите swt-win32-3232.dll в ту же директорию, что и rtn.jar.

Запустите javaw -Djava.library.path=. -cp .;rtn.jar ru.runa.notifier.PlatformLoader

или runa_tasks.exe