

ToughRadius 认证计费系统安装配置手册

jamiesun.net@gmail.com

1 系统介绍

ToughRadius 是基于开源 TinyRadius 开发的一个支持高并发的 Radius 服务器。

TinyRadius 提供了 Radius 协议实现基础,ToughRadius 增加了多线程的支持,可以支持高并发的认证计费请求。

ToughRadius 使用 java 开发,支持跨平台的部署。

ToughRadius 增加了数据库存储支持,通过 jdbc 驱动支持多种数据库,比如 Mysql,Hsqldb,Oracle 等,当前 ToughRadius 内置 Hsqldb 的支持,Hsqldb 是一个高性能的 java 嵌入式数据库,强烈建议你使用它来存储您的用户数据。

ToughRadius 实现了一个 Web 管理界面,提供用户管理,用户组管理,客户端管理等功能,支持灵活的属性扩展。

ToughRadius 的目标是实现一个更高效易用扩展灵活的认证计费系统,用一个精简的认证计费核心提供基础服务,通过扩展接口模块来提供更丰富的业务功能。

在使用过程中遇到问题,可以通过发邮件到jamiesun.net@gmail.com 咨询。也可以加入 QQ 群 (247860313) 参与讨论。

2 最简单的部署方式

如果您不想去做系统环境配置,版本构建这些太过专业的事情,您可以下载已经包含了 jdk 的发布版本,解压直接运行启动脚本即可

或者您系统中已经安装了 jdk1.6 或更高版本的 jdk,您可以通过修改 sh 或 bat 脚本中的 jdk 路径即可,然后直接运行启动脚本

3 系统环境配置

ToughRadius 是一个跨平台的认证计费系统，支持多种操作系统，这里我们主要介绍 windows 和 linux 下的安装配置为主。

1. java 环境设置

我们建议以 jdk1.6 来作为 ToughRadius 提供 java 运行环境，您可以从 java 官方网站上下载 jdk 的二进制安装包。根据您的目标平台选择 windows 版本还是 linux 版本，是 32 位版本还是 64 位版本。注意在设置完环境变量后要验证是否生效，输入 `java -version` 查看 java 版本是否正确，输入 `ant` 查看结果 `ant` 命令是否生效。在 linux 下设置完成后需要注销重登陆一次系统。

Jdk1.6 下载：<http://pan.baidu.com/share/link?shareid=2774752657&uk=604278103>

- windows 下的环境变量设置

右键点击我的电脑—属性—高级—环境变量，新建系统变量

`JAVA_HOME=C:\java\jdk1.6.0_37` //具体路径以实际为准

在 path 变量后加入 `;%JAVA_HOME%\bin`

- linux 下的环境变量设置

修改/etc/profile 文件，加入以下内容

`export JAVA_HOME=/opt/jdk1.6.0_37`

`export PATH=$JAVA_HOME/bin:$PATH`

2. ant 环境设置

如果你需要在目标平台上用源码构建版本，按以下步骤进行：

- 下载安装配置 ant

从 apache 网站 <http://ant.apache.org/bindownload.cgi> 下载 ant 安装包，解压到指定目录

和 java 环境变量配置一样

在 windows 下加入

`ANT_HOME=C:\ant1.9.0` //具体路径以实际为准

在 path 变量后加入 `;%ANT_HOME%\bin`

ToughRadius 认证计费系统安装配置手册

在 linux 下修改/etc/profile 文件，加入以下内容

```
export ANT_HOME=/opt/ant1.9.0
```

```
export PATH=$ANT_HOME/bin:$PATH
```

4 版本构建

请确认您已经设置好 ant 环境

从 ToughRadius 源码托管网站下载代码压缩包，解压到指定，以 linux 为例

<https://github.com/jamiesun/ToughRadius>

<http://git.oschina.net/jamiesun/toughradius>

```
cd /usr/local/src/
```

```
unzip ToughRadiusV1.0.zip
```

```
cd ToughRadiusV1.0
```

```
ant -f build.xml
```

注意构建过程的日志输出，确认构建成功

构建完成后，会生成 release 目录，release 目录下有构建完成的版本，将其拷贝到运行目录

```
cp /usr/local/src/ToughRadiusV1.0/release/ToughRadiusV1.0 /opt/ToughRadiusV1.0
```

数据库安装配置

进入 ToughRadius 运行目录，执行以下命令

```
cd /opt/ToughRadiusV1.0
```

```
chmod +x setupdb.sh
```

```
./setupdb.sh
```

按提示完成数据库的初始化

即将初始化数据库,请输入数据库名

```
>> radius
```

数据库已经存在，是否删除： y/n

```
>> y
```

```
...
```

ToughRadius 认证计费系统安装配置手册

是否导入数据脚本(docs/data/*.sql) : y/n

>> y

...

导入的数据 sql 脚本放在 docs/data/目录下，参考 import.sql

完成以上操作，就完成了数据库初始化了

应用系统配置

ToughRadius 的配置包括来系统配置文件和日至配置文件

conf/system.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<config>
```

```
  <radius>
```

```
    <version>ToughRadius V1.0</version> <!-- 版本 -->
```

```
    <authPort>1812</authPort> <!-- 认证端口 -->
```

```
    <acctPort>1813</acctPort> <!-- 计费端口 -->
```

```
    <adminPort>1815</adminPort> <!-- 版本 -->
```

```
    <maxSessionTimeout>86400</maxSessionTimeout> <!-- 默认最大会话时  
长 -->
```

```
  </radius>
```

```
  <webapp>
```

```
    <port>8080</port> <!-- web 管理端口 -->
```

```
    <maxConn>1024</maxConn> <!-- web 管理最大连接数 -->
```

```
  </webapp>
```

```
  <database>
```

```
    <driver>org.hsqldb.jdbc.JDBCdriver</driver> <!-- 数据库 jdbc 驱动
```

```
-->
```

```
    <url>jdbc:hsqldb:./data/radius</url> <!-- 数据库 url -->
```

```
    <username>sa</username> <!-- 数据库用户名 -->
```

```
    <password></password> <!-- 数据库密码 -->
```

```
  </database>
```

```
</config>
```

conf/log4j.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration PUBLIC "-//ZLAB //DTD Log4j Configuration
1.0.1//EN" "http://www.ly-bns.net/dtd/log4j.dtd">
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

  <appender name="console.log" class="org.apache.log4j.ConsoleAppender">
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d [%t] %p - %m%n" />
    </layout>
  </appender>

  <appender name="error.log"
class="org.apache.log4j.DailyRollingFileAppender">
    <param name="File" value="logs/error.log" />
    <param name="Append" value="true" />
    <param name="DatePattern" value="'.'yyyy-MM-dd"/>
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d [%t] %p - %m%n" />
    </layout>
    <filter class="org.apache.log4j.varia.LevelRangeFilter">
      <param name="LevelMin" value="WARN" />
      <param name="LevelMax" value="FATAL" />
    </filter>
  </appender>

  <appender name="info.log"
class="org.apache.log4j.DailyRollingFileAppender">
    <param name="File" value="logs/info.log" />
    <param name="Append" value="true" />
    <param name="DatePattern" value="'.'yyyy-MM-dd"/>
    <layout class="org.apache.log4j.PatternLayout">
```

```
<param name="ConversionPattern" value="%d [%t] %p - %m%n" />
</layout>
<filter class="org.apache.log4j.varia.LevelRangeFilter">
  <param name="LevelMin" value="DEBUG" />
  <param name="LevelMax" value="INFO" />
</filter>
</appender>

<appender name="acct.log"
class="org.apache.log4j.DailyRollingFileAppender">
  <param name="File" value="logs/acct/acct.rec" />
  <param name="Append" value="true" />
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%m%n" />
  </layout>
</appender>

<logger name="acct" additivity="false">
  <level value="info" />
  <appender-ref ref="acct.log" />
</logger>

<logger name="com.ibatis" additivity="true">
  <level value="DEBUG" />
</logger>
<logger name="java.sql.Connection" additivity="true">
  <level value="DEBUG" />
</logger>
<logger name="java.sql.Statement" additivity="true">
  <level value="DEBUG" />
</logger>
<logger name="java.sql.PreparedStatement" additivity="true">
  <level value="DEBUG" />
</logger>
<logger name="java.sql.ResultSet" additivity="true">
```

```
        <level value="DEBUG" />
    </logger>

    <root>
        <level value="info" />  <!--生产环境中，将日志级别改为error -->
        <appender-ref ref="console.log" />
        <appender-ref ref="error.log" />
        <appender-ref ref="info.log" />
    </root>

</log4j:configuration>
```

运行说明

完成所有前提工作以后，进入运行目录，启动应用

在 windows 下，直接点击 startup.bat 会打开一个控制台，并实时输出运行日志。通过编辑器可以打开查看 logs 目录下的日志文件

在 linux 下

```
cd /opt/ToughRadiusV1.0
```

```
chmod +x startup.sh
```

```
./startup.sh
```

通过过 vi 可以查看 logs 目录下的日志。

5 管理功能

打开浏览器，建议使用 ie8，chrome，火狐，猎豹等浏览器操作以便有更好的显示效果。

进入 <http://127.0.0.1:8080> 默认管理密码为 admin/admin

1. 登录界面



2. 管理首页

ToughRadius 认证计费系统安装配置手册

ToughRadius

首页修改密码在线用户退出登录

客户端管理
用户组管理
用户管理
退出系统

欢迎登陆系统【admin】 / 现在是 2013-09-07 15:23:14

系统信息

系统版本 :	ToughRadius V1.0	已运行时间 :	0天0小时7分钟	内存使用 :	已用27M / 最大910M
当前登陆IP地址 :	127.0.0.1	最后登录时间 :	2013-09-07 15:23:14	当前在线 :	0

统计信息

<ul style="list-style-type: none">认证包总数 : 0认证接受包 : 0认证拒绝包 : 0认证丢弃包 : 0	<ul style="list-style-type: none">记帐包总数 : 0记帐上线包 : 0记帐更新包 : 0记帐下线包 : 0记帐开始包 : 0记帐结束包 : 0记帐丢弃包 : 0
---	---

建议您使用IE8以上或使用其他更好的浏览器如（Chrome浏览器,火狐浏览器）

3. 客户端管理

ToughRadius

首页修改密码在线用户退出登录

客户端管理
用户组管理
用户管理
退出系统

客户端管理 / 客户端列表 / 新增客户端

客户端IP地址

192.168.1.1

客户端类型

ros

共享密钥

123456

客户端描述

ros client

保存

返回

建议您使用IE8以上或使用其他更好的浏览器如（Chrome浏览器,火狐浏览器）

4. 用户组管理

ToughRadius 认证计费系统安装配置手册

ToughRadius

首页修改密码在线用户退出登录

客户端管理
用户组管理
用户管理
退出系统

用户组管理 / 用户组查询 / 新增用户组

用户组名

test

用户组描述

test

保存

返回

用户组属性管理

资费(1/2:预付费计时/预付费包月)

FEE_POLICY

2

更新

删除

资费价格(分), 每小时/每月)

FEE_PRICE

3000

更新

删除

最大会话时长 (秒)

Session-Timeout

3600

更新

删除

状态(1/2:正常/停

▼

输入属性名

输入属性值

保存

建议您使用IE8以上或使用其他更好的浏览器如 (Chrome浏览器, 火狐浏览器)

5. 用户管理

ToughRadius

首页修改密码在线用户退出登录

客户端管理
用户组管理
用户管理
退出系统

用户管理 / 用户查询 / 新增用户

用户名

test

test ▼

密码

保存

返回

用户组属性管理

用户余额 (分)

CREDIT

0

更新

删除

RouterOS限速属性

Mikrotik-Rate-Limit

64k/64k 256k/256k 128k/128k 1k

更新

删除

状态(0/1/2:未激活/正常/停机)

STATUS

1

更新

删除

过期时间(####-##-##

▼

输入属性名

输入属性值

保存

建议您使用IE8以上或使用其他更好的浏览器如 (Chrome浏览器, 火狐浏览器)