文件夹说明：

Scod：提供sasl机制

Jabberd2.0s10

控制台测试

输入 telnet localhost 5222 连接到服务器，可以看到下面的信息：

Trying 127.0.0.1...

Connected to localhost.localdomain (127.0.0.1).

Escape character is '^]'.

现在打开XML流，粘贴下面的XML代码到telnet窗口：

<stream:stream to='localhost' xmlns='jabber:client' xmlns:stream='http://etherx.jabber.org/streams'>;

会立即收到如下信息

<?xml version='1.0'?><stream:stream xmlns:stream='http://etherx.jabber.org/streams' xmlns='jabber:client' from='localhost' id='u7nib194naruz8mkps0ymwd3nvobaz656toiuymi'>

发送下面的XML代码来看看需要为注册用户提供什么信息：

<iq id='reg1' type='get'>;<query xmlns='jabber:iq:register'/>;</iq>;

应该收到下面的回复，告诉你要注册一个用户需要提供名字，email地址，帐号，和密码。

<iq xmlns='jabber:client' id='reg1' type='result'>

<query xmlns='jabber:iq:register'>

<username/>

<password/>

<instructions>Enter a username and password to register with this server.</instructions>

</query>

</iq>

发送注册信息到服务器

<iq id='reg2' type='set'>;<query xmlns='jabber:iq:register'>; <username>;jabberd2</username>;<password>;secret</password>; </query>; </iq>;

结果用户已经存在，收到如下错误信息：

<iq xmlns='jabber:client' type='error' id='reg2'>;

<error type='cancel' code='409'>

<conflict xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>

</error>

<query xmlns='jabber:iq:register'>;

<username>;jabberd2</username>;

<password>;secret</password>;

</query>;

</iq>

重新输入一个不存在的用户

<iq id='reg2' type='set'>;<query xmlns='jabber:iq:register'>; <username>;jabberd2zw</username>;<password>;secret</password>; </query>; </iq>;

收到注册成功信息：

<iq xmlns='jabber:client' id='reg2' type='result'/>

现在已经注册了一个账号，可以登录了。首先，问一下服务器登录需要什么信息：

<iq id='auth1' type='get'>;<query xmlns='jabber:iq:auth'>;<username>;jabberd2zw</username>;</query>;</iq>;

服务器会回答需要哪些认证信息：

<iq xmlns='jabber:client' id='auth1' type='result'>

<query xmlns='jabber:iq:auth'>

<username>;jabberd2zw</username>

<resource/>

<password/>

<digest/>

<sequence>499</sequence>

<token>4D412274</token>

</query></iq>

<iq id='auth1' type='get'>;

<query xmlns='jabber:iq:auth'>;

<username>;jabberd2zw</username>;

</query>;

</iq>;

可以选择"password" (明文密码)或"digest" (加密密码)；为了方便，我们选择明文密码。还必须为这个连接指定"resource"（资 源）。发送下面的XML代码：

<iq id='auth2' type='set'>;<query xmlns='jabber:iq:auth'>;<username>;jabberd2zw</username>;<password>;secret</password>; <resource>;home</resource>;</query>;</iq>;

服务器会发送信息，确认认证成功：

<iq xmlns='jabber:client' id='auth2' type='result'/>

最后，发送'presence'信息，让服务器知道你已经在线并准备接收消息：

<presence/>;

下面可以发送信息给另一个用户

<message to=';jabberd2@localhost'>;<body >;hi!</body>;</message>;

若用户名打成了'jabberd2@localhost'，或者用户不存在可以收到如下错误信息：

<message xmlns='jabber:client' type='error' from='jabberd2@localhost' to=';jabberd2zw@localhost/;home'>;

<error type='cancel' code='404'>

<item-not-found xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>

</error>

<body>;hi!</body>;

</message>

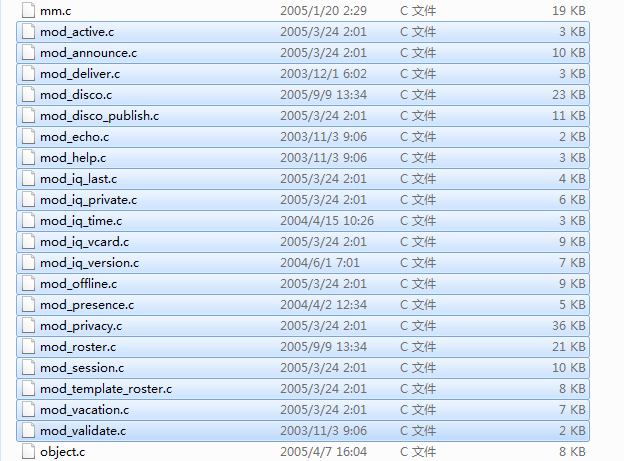
正常则不收到回复。输入下面字符串可以关闭xml流：

</stream:stream>;

代码修改方案

邮件网关及短信网关添加方案：

Sm模块中包含很多数据包处理模块，所在文件名以mod\_开头，包含如下：



可在此添加一个mod\_email模块。

在sm模块，sm\_sx\_callback（sm.h）中当包含<presence>或者<msg to=’’ from=’’>的数据包到来时，经过event\_PACKET的处理，最重要的是如下一段：

pkt = pkt\_new(sm, (nad\_t) data);

if(pkt == NULL) {

log\_debug(ZONE, "invalid packet, dropping");

nad\_free(nad);

return 0;

}

/\* go \*/

dispatch(sm, pkt);

return 0;

在pkt\_new函数中，message的处理由以下一句说明：

/\* messages are simple \*/

if(NAD\_ENAME\_L(pkt->nad, 1) == 7 && strncmp("message", NAD\_ENAME(pkt->nad, 1), 7) == 0) {

pkt->type = pkt\_MESSAGE;

if(attr >= 0 && NAD\_AVAL\_L(pkt->nad, attr) == 5 && strncmp("error", NAD\_AVAL(pkt->nad, attr), 5) == 0)

pkt->type |= pkt\_ERROR;

return pkt;

}

直接返回，交由dispatch（dispatch.c）

/\* non route packets are special-purpose things from the router \*/

if(!(pkt->rtype & route\_UNICAST)) {

ret = mm\_pkt\_router(pkt->sm->mm, pkt);

switch(ret) {

case mod\_HANDLED:

break;

case mod\_PASS:

default:

/\* don't ever bounce these \*/

pkt\_free(pkt);

break;

}

return;

}

深入mm\_pkt\_router，发现数据包交由pkt\_router处理，sm.xml中

<chain id="**pkt-router**">

<module>**session**</module>

- <!--

take sessions offline their c2s disappears

-->

<module>**disco**</module>

- <!--

query new components for service information

-->

</chain>

先交由mod\_session的pkt\_router函数处理



整个就是一个这样的过程，我们添加邮件处理的位置，不妨选在：

[**-**](file:///C:\Users\ZW\Desktop\jabberd2.0项目文件\xml\sm.xml) <chain id="**in-sess**">

<module>**validate**</module>

- <!--

validate packet type

-->

<module>**privacy**</module>

- <!--

manage privacy lists

-->

<module>**roster**</module>

- <!--

handle roster get/sets and s10ns

-->

<module>**vacation**</module>

- <!--

manage vacation settings

-->

<module>**iq-vcard**</module>

- <!--

store and retrieve the user's vcard

-->

<module>**iq-private**</module>

- <!--

manage the user's private data store

-->

<module>**disco**</module>

- <!--

respond to agents requests from sessions

-->

<module>**offline**</module>

- <!--

if we're coming online for the first time, deliver queued messages

-->

<module>**announce**</module>

- <!--

deliver motd

-->

<module>**presence**</module>

- <!--

process and distribute presence updates

-->

<module>**deliver**</module>

- <!--

deliver packets with full jids directly

-->

</chain>

在in-session链上最上面添加一个<module>**email**</module>，而在email模块中，处理形如<message to=’\_email\_@domain/resource’ from=’jid’>或者形如<message type=’’>等形式的数据包即可。