

## Asterisk 事件监控

asterisk 有一个很好的东东，就是 Event，会主动通知客户端服务器发生了什么……不过前提是你得先连上服务器。

好，不废话了，下面开工。

首先声明下环境：CentOS 6

一、安装并配置 asterisk

1、下载 asterisk

wget <http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-1.8.7.1.tar.gz>

2、解压并安装

tar zxvf asterisk-1.8.7.1.tar.gz

cd asterisk-1.8.7.1

./configure && make && make install && make samples

涉及的命令：

yum install libxml2-devel

yum install ncurses-devel

3、开启 AMI 管理

编辑 manager.conf 文件，将 general 中的 enabled 置为 yes

```
21 ; default is 60 seconds.
22 ;
23 [general]
24 enabled = yes
25 ;webenabled = yes
```

添加管理员：

[admin]

secret = 123456

read = system,call,log,verbose,agent,user,config,dtmf,reporting,cdr,dialplan

write = system,call,agent,user,config,command,reporting,originate

```
147 [admin]
148 secret = 123456
149 read = system,call,log,verbose,agent,user,config,dtmf,reporting,cdr,dialplan
150 write = system,call,agent,user,config,command,reporting,originate
151
```

二、编写 Asterisk 事件监控程序

原理：通过 login action 连上 Asterisk 的 5038 端口，监听此端口并把消息输出。

下面是 C++实现的代码：

```
/*
    File      : asteriskEventCat.cpp
    Author    : Mike
    E-Mail    : Mike_Zhang@live.com
*/

#include <iostream>
#include <string>
#include <fstream>
#include <boost/asio.hpp>
#define BLOCK_SIZE 10*1024

using namespace std;
using namespace boost::asio;

string strLogin(string userName, string pswwd)
{
    string msg="";
    msg = "Action: login\r\n";
```

```

    msg += "UserName: " + userName + "\r\n";
    msg += "Secret: " + pswwd + "\r\n";
    msg += "\r\n";
    return msg;
}

int main()
{
    io_service iosv;
    ip::tcp::socket s(iosv);
    string svrIp = "";
    cout<<"Input server ip : ";
    cin>>svrIp;
    ip::tcp::endpoint ep(ip::address_v4::from_string(svrIp.c_str()), 5038);

    boost::system::error_code ec;
    s.connect(ep, ec);
    if(ec)
    {
        cout << boost::system::system_error(ec).what() << endl;
        return -1;
    }
    else
    {
        cout<<"Connect success!"<<endl;
    }

    string msg="";

    string userName,password;
    cout<<"User      : ";
    cin>>userName;
    cout<<"Password : ";
    cin>>password;

    msg += strLogin(userName.c_str(), password.c_str());
    //      msg += strLogin("admin", "admin");
    size_t len = s.write_some(buffer(msg.c_str()), ec);
    if(len <= 0)
    {
        cout<<"Send message fail!"<<endl;
        return -1;
    }

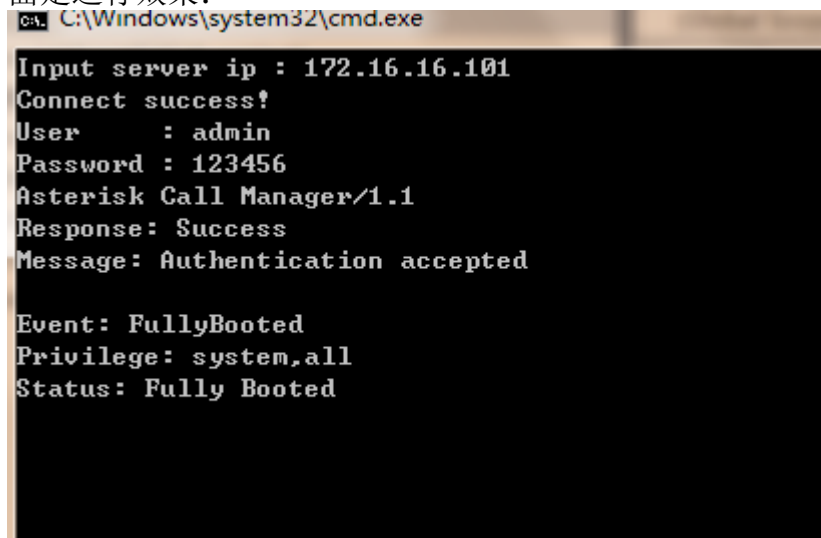
    std::ofstream fout("EventCat.txt");
    while(true)
    {
        char buf[BLOCK_SIZE] = {0};
        len=s.read_some(buffer(buf), ec);
        if(len<=0)
            break;
        cout.write(buf, len);
        fout<<buf;
        fout.flush();
    }

    return 0;
}

```

### 三、进行事件监控

首先要保证你的防火墙让你过 5038 端口，嫌配置麻烦的话把防火墙关闭就行了，下面是运行效果：



```
C:\Windows\system32\cmd.exe
Input server ip : 172.16.16.101
Connect success!
User      : admin
Password  : 123456
Asterisk Call Manager/1.1
Response: Success
Message: Authentication accepted

Event: FullyBooted
Privilege: system,all
Status: Fully Booted
```

我这人比较懒，虽是 C/C++ 程序员，可是总嫌 C++ 写的代码多，经常用 python 做模型，下面是我之前写的，这里也顺便粘出来一起总结，仅供参考，欢迎拍砖。

Python 代码：

```
'''
    File      : asteriskEventCat.py
    Author    : Mike
    E-Mail    : Mike_Zhang@live.com
'''
import socket

bufLen = 1024 * 10
def strLogin(userName,passwd):
    msg = 'Action: login\r\n'
    msg += 'UserName: '+userName+'\r\n'
    msg += 'Secret: '+passwd+'\r\n'
    msg += '\r\n'
    return msg
def main():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((raw_input("Input ip : "), 5038))

    if not s :
        print "Connect fail!"
        return
    s.send(strLogin(raw_input("Input user name :"),raw_input("Input
password :")))
    while True:
        data = s.recv(bufLen)
        if len(data) <=0 : continue
        print data
if __name__ == '__main__':
    main()
    raw_input("Press Enter to continue")
```

在学习 go 语言，捎带下，也方便我以后 Ctrl + C:

```
/*
    File      : asteriskEventCat.go
    Author    : Mike
    E-Mail    : Mike_Zhang@live.com
*/
package main

import (
    "net"
    "bufio"
    "os"
)

var bufLen = 1024 * 10
var lineLen = 2 // windows : "\r\n" 2 ; linux : "\n" 1

func main() {
    reader := bufio.NewReader(os.Stdin)
    print("Input ip : ") ; svrIp, _ := reader.ReadBytes('\n')
    print("Input userName : ") ; usrName, _ := reader.ReadBytes('\n')
    print("Input passwd : ") ; pwd, _ := reader.ReadBytes('\n')
    conn, err := net.Dial("tcp", string(svrIp[0:len(svrIp)-lineLen])+":5038")
    defer conn.Close()
    if err != nil {
        println("Error : ", err.String())
    }
    conn.Write([]byte("Action: login\r\nUserName: "+
        string(usrName[0:len(usrName)-lineLen])+"\r\nSecret: "+
        string(pwd[0:len(pwd)-lineLen])+"\r\n\r\n"))
    for {
        p := make([]byte, bufLen)
        sz, _ := bufio.NewReader(conn).Read(p)
        println(string(p[0:sz]))
    }
}
```