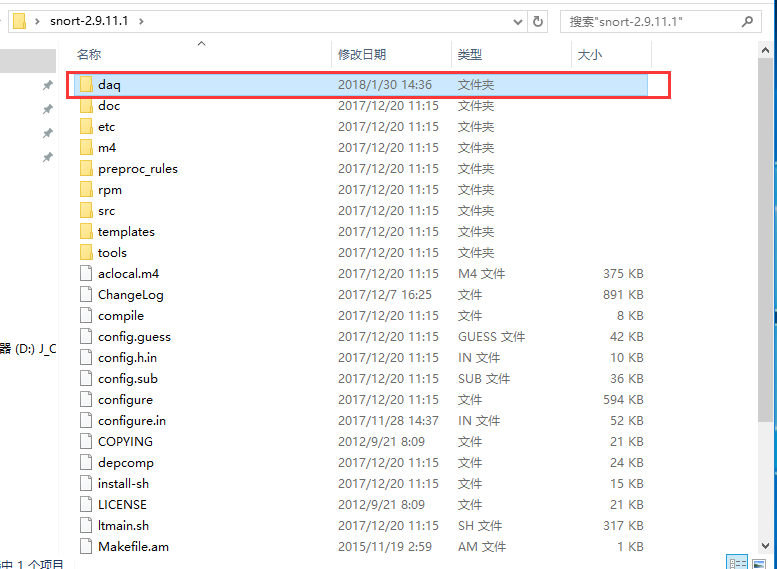
编译环境WIN10-x64,VS2013+Cygwin(要安装bison、flex、sed)

从snort官网上下载 daq-2.0.6.tar.gz\snort-2.9.11.1.tar.gz

<https://www.snort.org/downloads#snort-downloads>



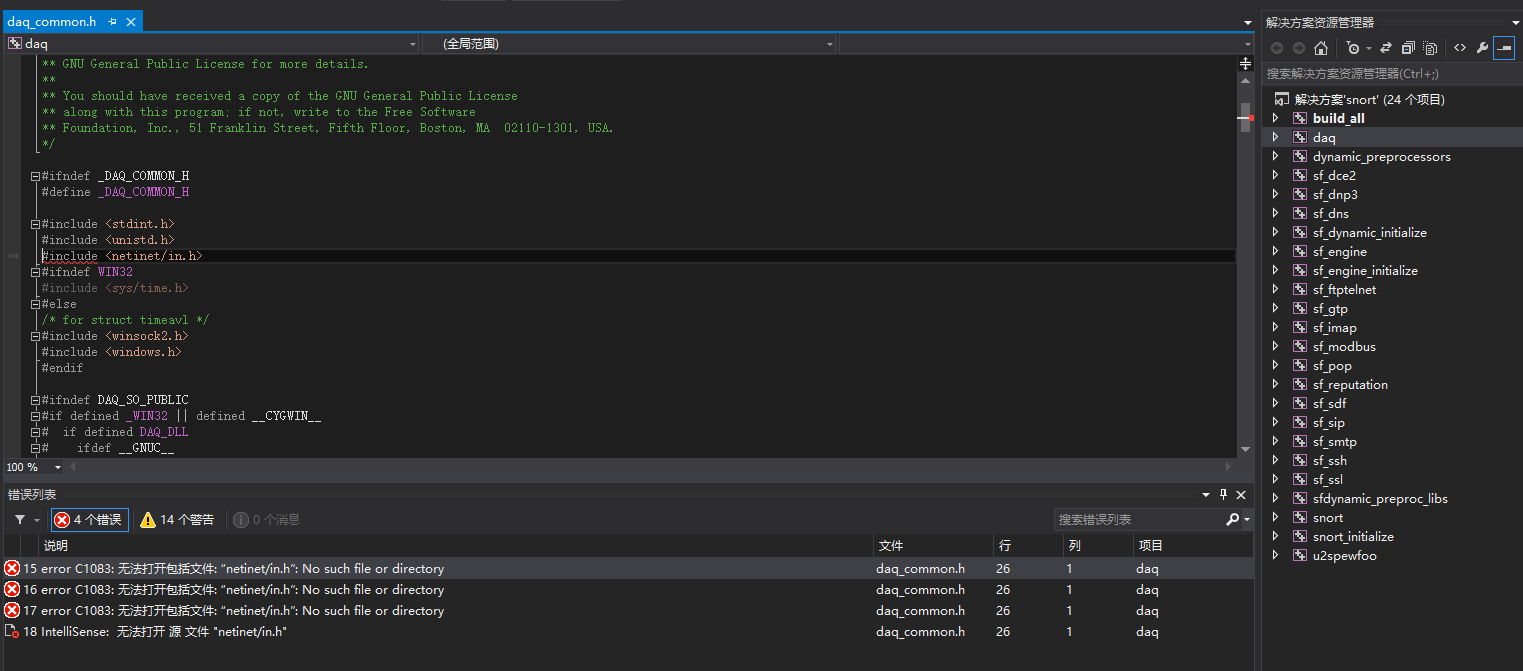
分别解压，并将daq-2.0.6重命名为daq并放在snort的第一层文件夹中



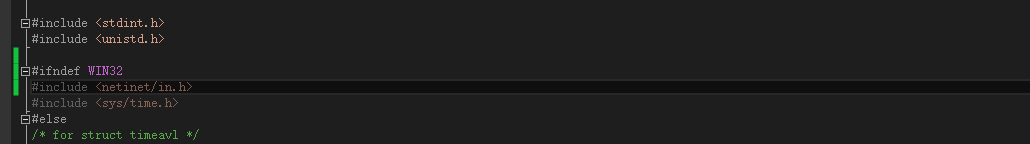
用VS2013打开snort.dsw

snort-2.9.11.1\src\win32\WIN32-Prj（目录）

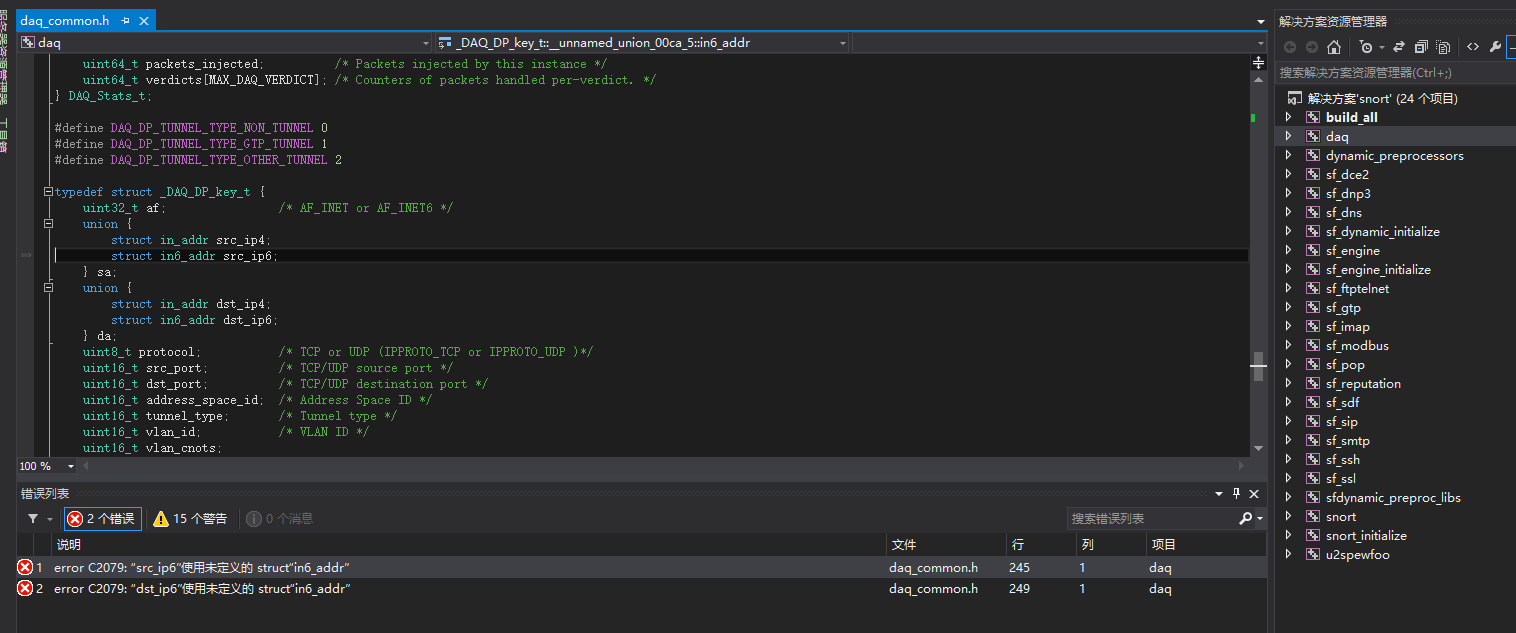
先编译daq报如下错误



修改如下



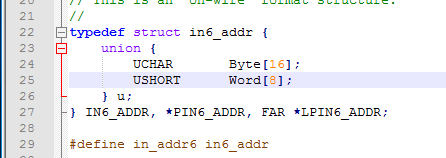
再次编译daq，报如下错误



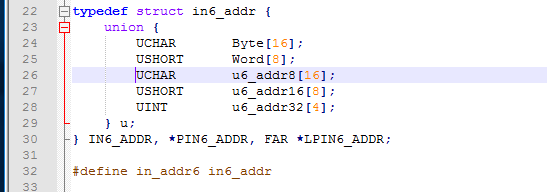
此时是因为VS2013自带的系统文件in6addr.h

（路径C:\Program Files (x86)\Windows Kits\8.1\Include\shared）

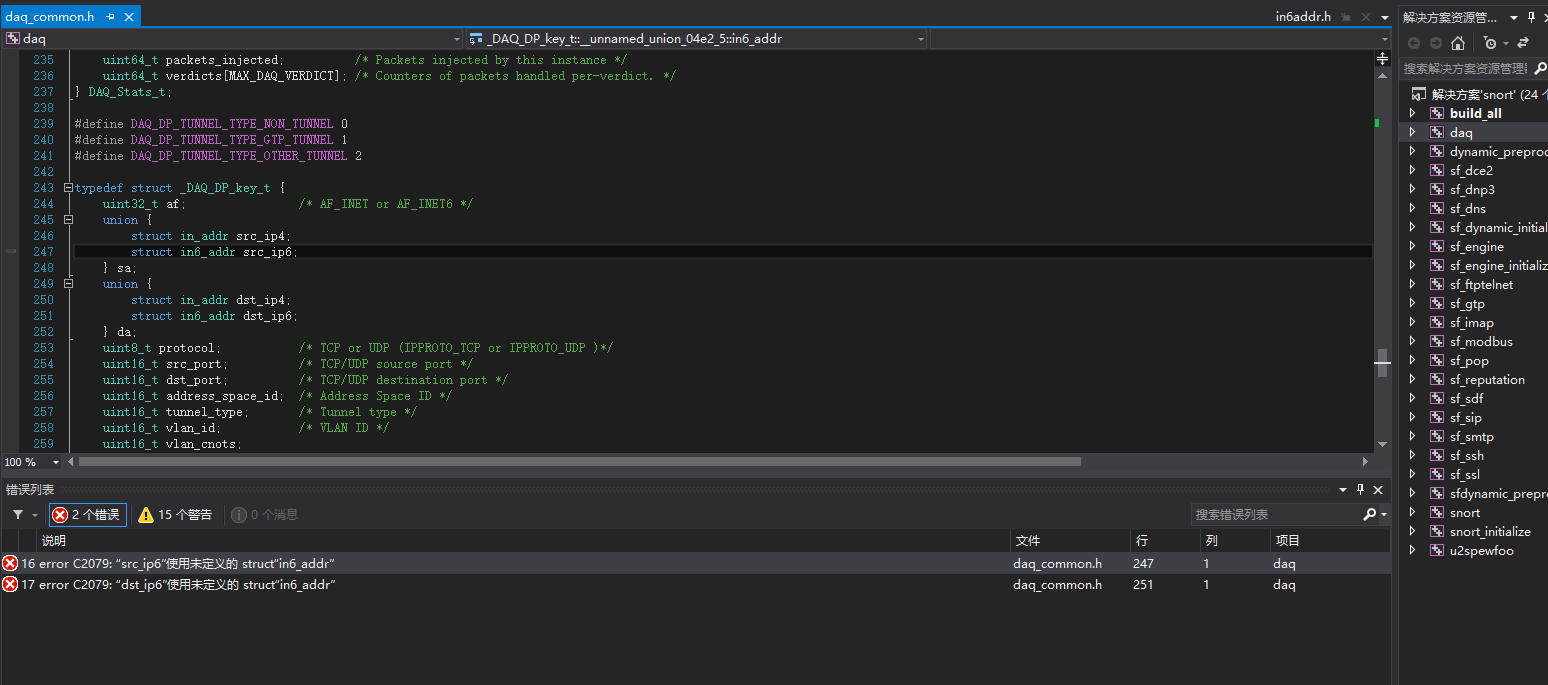
的联合体成员数不够



修改如下



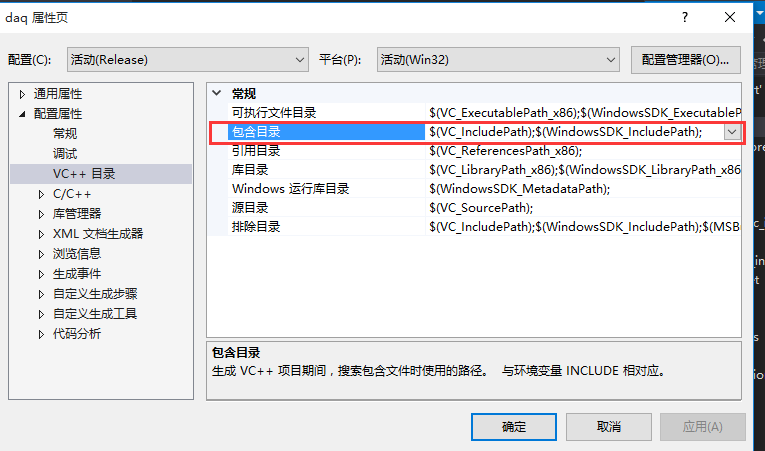
再次编译daq，还是出错

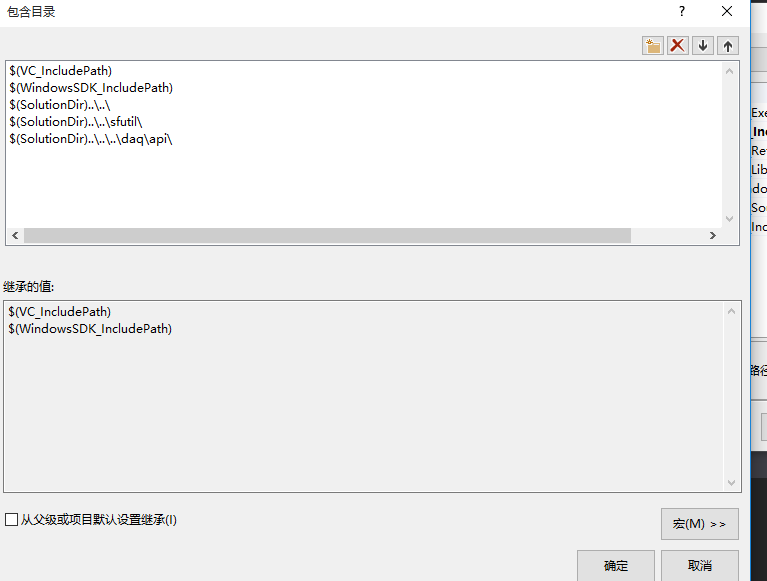


修改方法（步骤有点多，请仔细看）

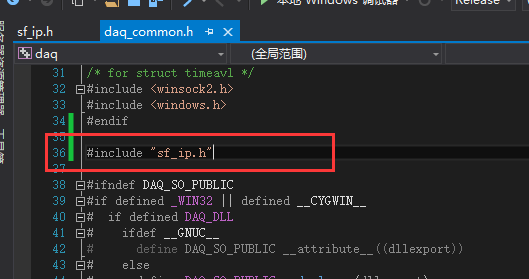
右键daq的属性 添加VC++目录—>包含目录

$(VC\_IncludePath);$(WindowsSDK\_IncludePath);$(SolutionDir)..\..\;$(SolutionDir)..\..\sfutil\;$(SolutionDir)..\..\..\daq\api\;





再在daq\_common.h中引入 #include “sf\_ip.h” 如下图所示



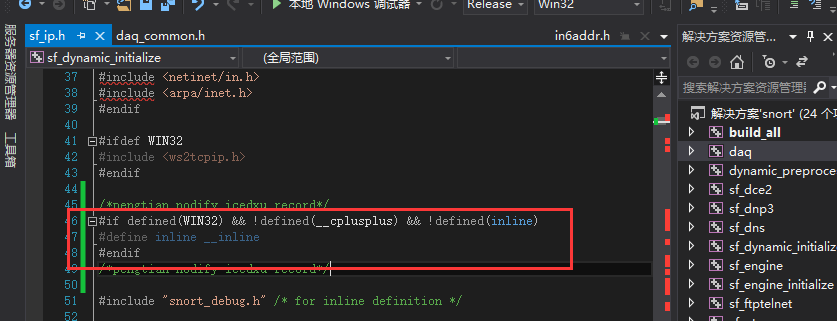
进入sf\_ip.h文件加入

#if defined(WIN32) && !defined(\_\_cplusplus) && !defined(inline)

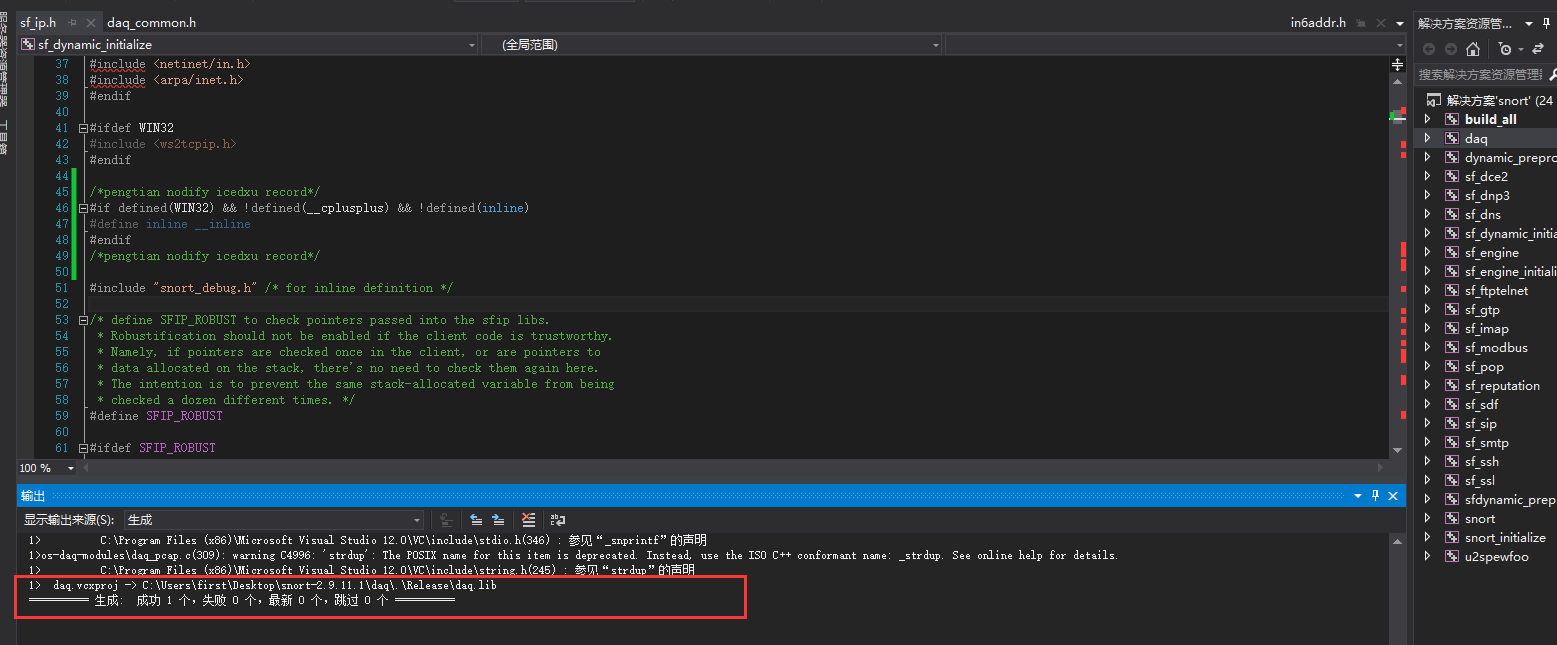
#define inline \_\_inline

#endif

如下图所示

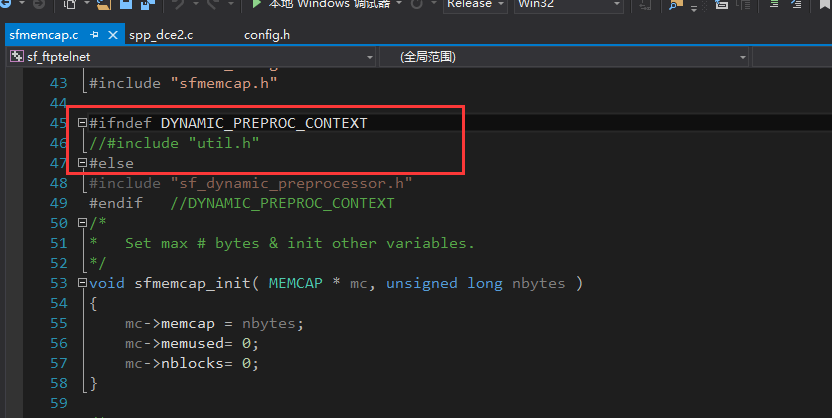


再次编译daq项目，成功如下图

先编译一下dynamic\_preprocessors 会生成一些默认目录（接下来会用到，不编译就不会有include目录）

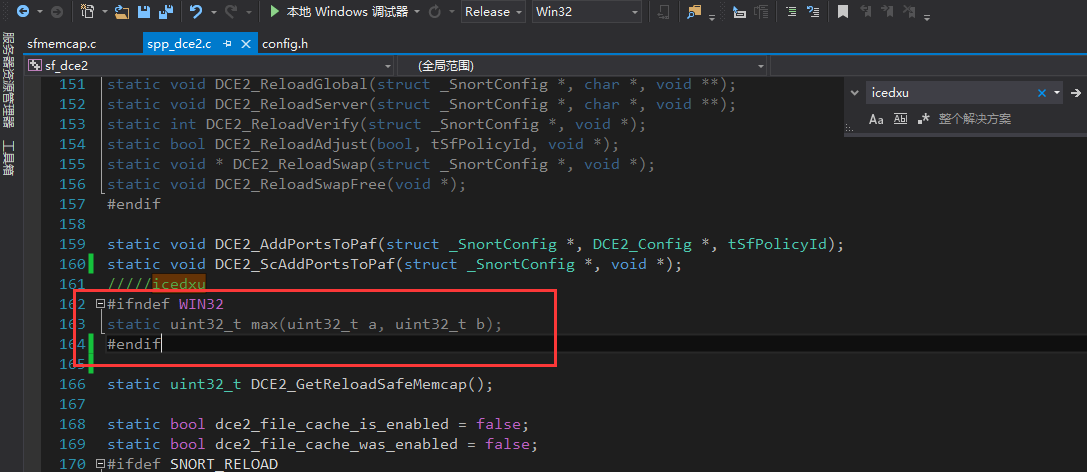
将reload\_api.h 从\snort-2.9.11.1\src\ 复制到 \snort-2.9.11.1\src\dynamic-preprocessors\include\ 目录中

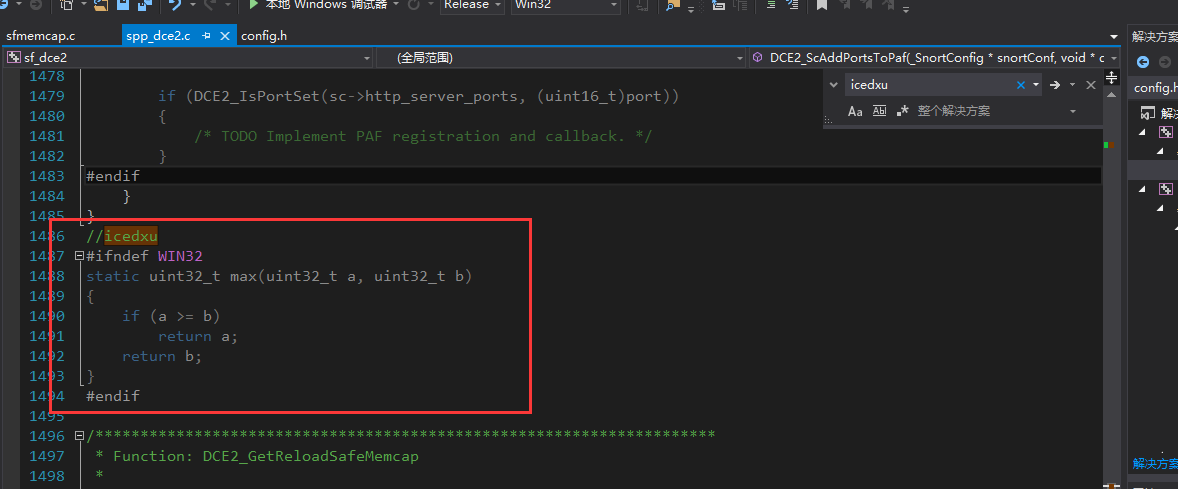
将sfmemcap.h/sfmemcap.c 从\snort-2.9.11.1\src\sfutil\ 复制到 \snort-2.9.11.1\src\dynamic-preprocessors\include\ 之后修改\include\sfmemcap.c文件内容，注释#include”util.h”



将 reg\_test.h/reg\_test.c 从\snort-2.9.11.1\src\ 复制到 \snort-2.9.11.1\src\dynamic-preprocessors\include\ 目录中

修改\snort-2.9.11.1\src\dynamic-preprocessors\dcerpc2\spp\_dce2.c 文件如下图所示



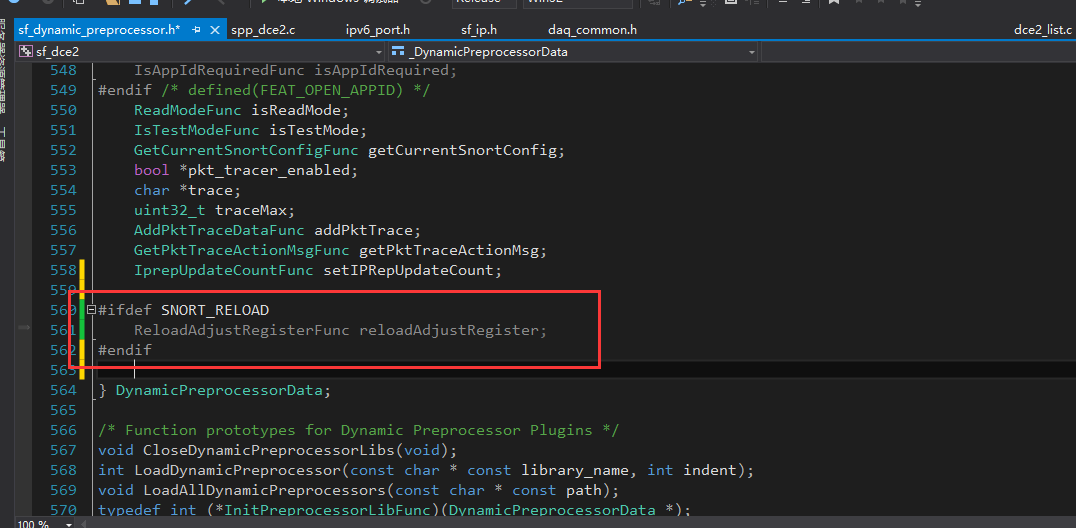


修改\snort-2.9.11.1\src\dynamic-preprocessors\include\sf\_dynamic\_preprocessor.h文件

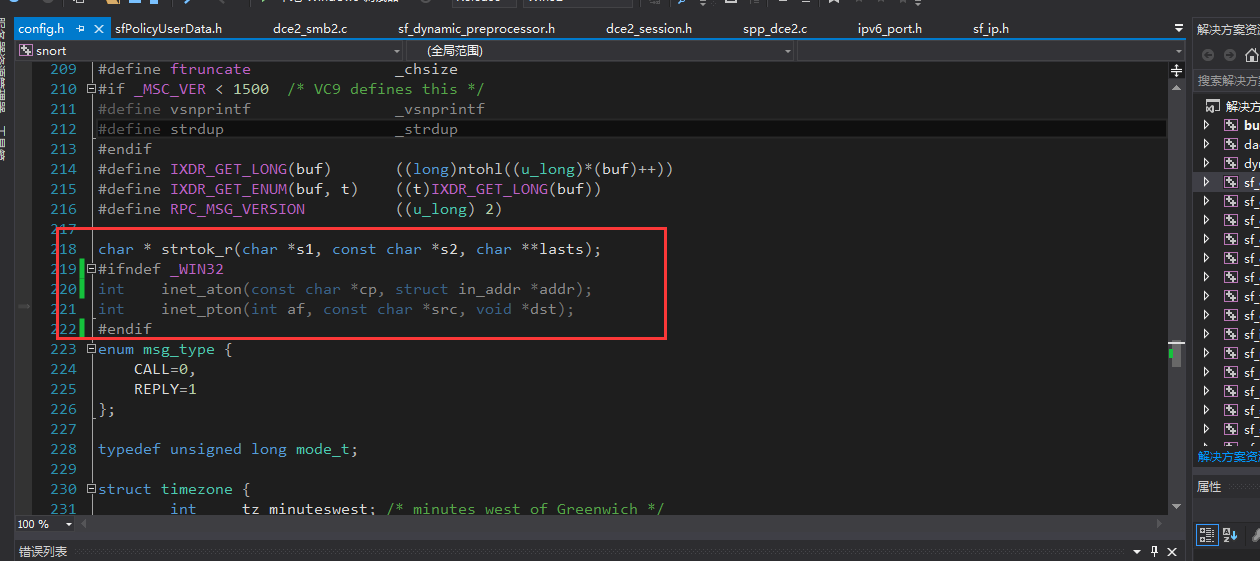
及

\snort-2.9.11.1\src\dynamic-plugins\sf\_dynamic\_preprocessor.h文件 （相同的文件）

成下图所示



修改\snort-2.9.11.1\src\win32\WIN32-Includes\config.h文件如下图所示



将\snort-2.9.11.1\src\ 中的 reload.c/.h和pkt\_tracer.c/.h添加到snort项目中

并修改pkt\_tracer.c 文件

添加

/\*inet\_ntop\*/

static const char \*

inet\_ntop\_v4(const void \*src, char \*dst, size\_t size)

{

const char digits[] = "0123456789";

int i;

struct in\_addr \*addr = (struct in\_addr \*)src;

u\_long a = ntohl(addr->s\_addr);

const char \*orig\_dst = dst;

if (size < INET\_ADDRSTRLEN) {

errno = ENOSPC;

return NULL;

}

for (i = 0; i < 4; ++i) {

int n = (a >> (24 - i \* 8)) & 0xFF;

int non\_zerop = 0;

if (non\_zerop || n / 100 > 0) {

\*dst++ = digits[n / 100];

n %= 100;

non\_zerop = 1;

}

if (non\_zerop || n / 10 > 0) {

\*dst++ = digits[n / 10];

n %= 10;

non\_zerop = 1;

}

\*dst++ = digits[n];

if (i != 3)

\*dst++ = '.';

}

\*dst++ = '\0';

return orig\_dst;

}

#ifdef HAVE\_IPV6

static const char \*

inet\_ntop\_v6(const void \*src, char \*dst, size\_t size)

{

const char xdigits[] = "0123456789abcdef";

int i;

const struct in6\_addr \*addr = (struct in6\_addr \*)src;

const u\_char \*ptr = addr->s6\_addr;

const char \*orig\_dst = dst;

int compressed = 0;

if (size < INET6\_ADDRSTRLEN) {

errno = ENOSPC;

return NULL;

}

for (i = 0; i < 8; ++i) {

int non\_zerop = 0;

if (compressed == 0 &&

ptr[0] == 0 && ptr[1] == 0 &&

i <= 5 &&

ptr[2] == 0 && ptr[3] == 0 &&

ptr[4] == 0 && ptr[5] == 0) {

compressed = 1;

if (i == 0)

\*dst++ = ':';

\*dst++ = ':';

for (ptr += 6, i += 3;

i < 8 && ptr[0] == 0 && ptr[1] == 0;

++i, ptr += 2);

if (i >= 8)

break;

}

if (non\_zerop || (ptr[0] >> 4)) {

\*dst++ = xdigits[ptr[0] >> 4];

non\_zerop = 1;

}

if (non\_zerop || (ptr[0] & 0x0F)) {

\*dst++ = xdigits[ptr[0] & 0x0F];

non\_zerop = 1;

}

if (non\_zerop || (ptr[1] >> 4)) {

\*dst++ = xdigits[ptr[1] >> 4];

non\_zerop = 1;

}

\*dst++ = xdigits[ptr[1] & 0x0F];

if (i != 7)

\*dst++ = ':';

ptr += 2;

}

\*dst++ = '\0';

return orig\_dst;

}

#endif /\* HAVE\_IPV6 \*/

const char \*

inet\_ntop(int af, const void \*src, char \*dst, size\_t size)

{

switch (af) {

case AF\_INET:

return inet\_ntop\_v4(src, dst, size);

#ifdef HAVE\_IPV6

case AF\_INET6:

return inet\_ntop\_v6(src, dst, size);

#endif

default:

errno = EAFNOSUPPORT;

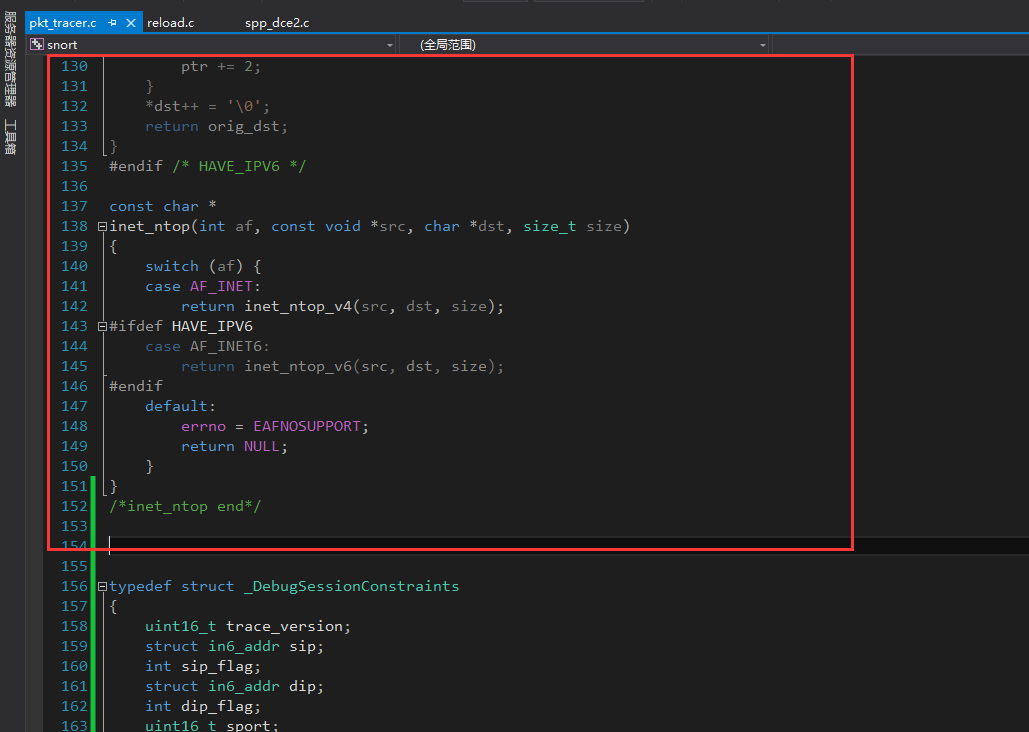
return NULL;

}

}

/\*inet\_ntop end\*/

到如下位置



接下来为每个文件添加 VC++目录-🡪包含目录

dynamic\_preprocessors 添加

$(VC\_IncludePath);$(WindowsSDK\_IncludePath);$(SolutionDir)..\..\;$(SolutionDir)..\..\sfutil\;$(SolutionDir)..\..\..\daq\api\

sf\_dce2、sf\_dnp3、sf\_dns、sf\_dynamic\_initialize 、sf\_engine、sf\_engine\_initialize、sf\_gtp、sf\_pop、sf\_reputation、sf\_sdf、sf\_sip、sf\_smtp、sf\_ssh、sf\_ssl、snort\_initialize、u2spewfoo、sf\_modbus添加

$(VC\_IncludePath);$(WindowsSDK\_IncludePath);$(SolutionDir)..\..\dynamic-preprocessors\include\;$(SolutionDir)..\..\dynamic-preprocessors\ssl\_common\;$(SolutionDir)..\..\dynamic-preprocessors\libs\;$(SolutionDir)..\..\dynamic-preprocessors\include\;$(SolutionDir)..\..\..\;

sf\_ftptelnet

$(VC\_IncludePath);$(WindowsSDK\_IncludePath);$(SolutionDir)..\..\dynamic-preprocessors\include\;$(SolutionDir)..\..\dynamic-preprocessors\ssl\_common\;$(SolutionDir)..\..\dynamic-preprocessors\libs\;$(SolutionDir)..\..\dynamic-preprocessors\include\;$(SolutionDir)..\..\..\daq\;$(SolutionDir)..\..\..\

sf\_imap

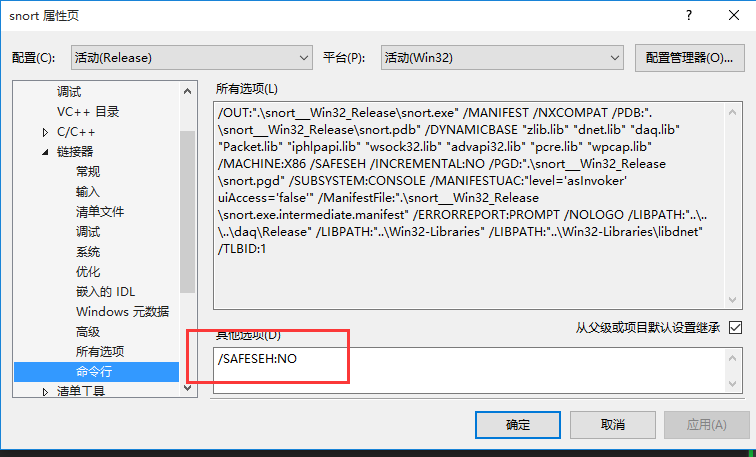
$(VC\_IncludePath);$(WindowsSDK\_IncludePath);$(SolutionDir)..\..\dynamic-preprocessors\include\;$(SolutionDir)..\..\dynamic-preprocessors\ssl\_common\;$(SolutionDir)..\..\dynamic-preprocessors\libs\;$(SolutionDir)..\..\dynamic-preprocessors\include\;$(SolutionDir)..\..\..\;$(SolutionDir)..\..\dynamic-examples\dynamic-preprocessor\;

sfdynamic\_preproc\_libs

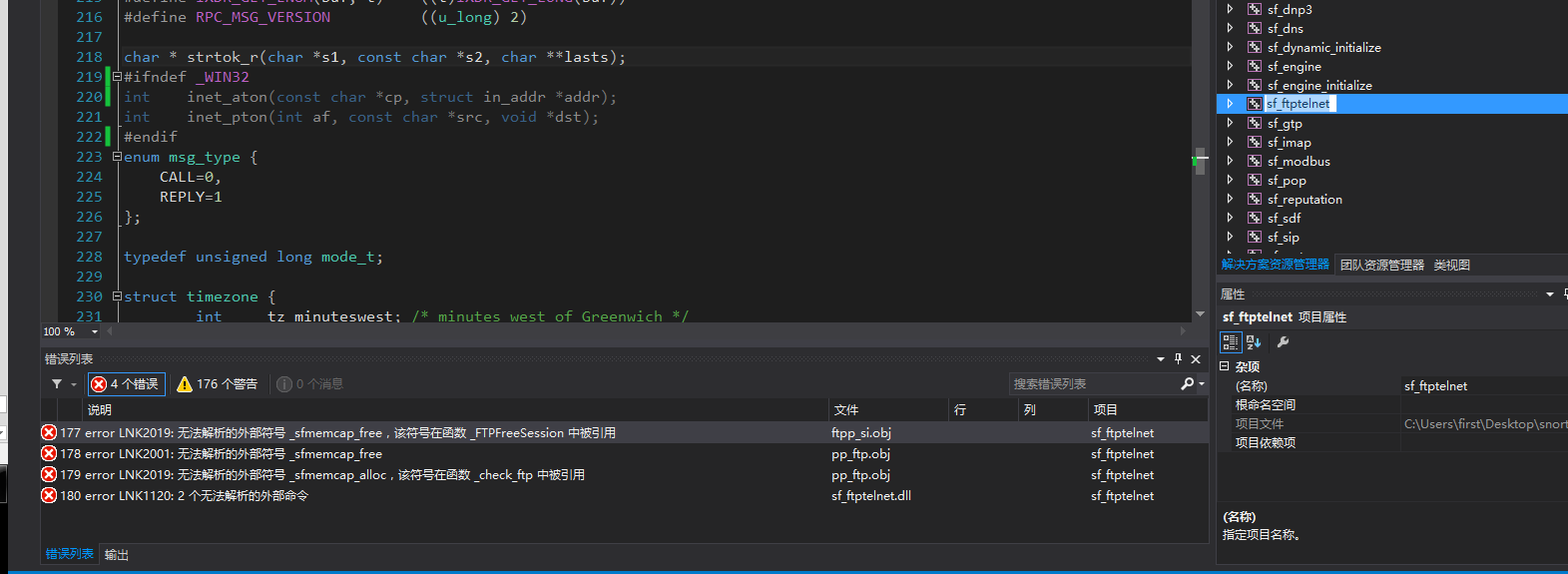
$(VC\_IncludePath);$(WindowsSDK\_IncludePath);$(SolutionDir)..\..\dynamic-preprocessors\ssl\_common\;$(SolutionDir)..\..\..\;$(SolutionDir)..\..\dynamic-examples\dynamic-preprocessor\;

Snort

$(VC\_IncludePath);$(WindowsSDK\_IncludePath);$(SolutionDir)..\..\..\;$(SolutionDir)..\..\..\src\;$(SolutionDir)..\..\..\src\sfutil;$(SolutionDir)..\..\..\src\output-plugins ;$(SolutionDir)..\..\..\src\detection-plugins ;$(SolutionDir)..\..\..\src\dynamic-plugins ;;$(SolutionDir)..\..\..\src\target-based ;$(SolutionDir)..\..\..\src\control ;$(SolutionDir)..\..\..\src\file-process ;$(SolutionDir)..\..\..\src\file-process\libs ;$(SolutionDir)..\..\..\src\side-channel ;$(SolutionDir)..\..\..\src\side-channel\plugins ;$(SolutionDir)..\..\..\src\reload-adjust\;$(SolutionDir)..\..\dynamic-preprocessors\libs\;$(SolutionDir)..\..\dynamic-examples\dynamic-preprocessor\;$(SolutionDir)..\..\..\daq\api\;



在添加完成编译sf\_ftptelnet 时出现



说明sfmemcap.c/.h文件没有被sf\_ftptelnet项目识别，此时手动添加此文件即可