SendDataToClient ，SendRequestLocal和SendOperationLocal 的区别

enum{

ProtoTypeNotify=0, // 提醒通知

ProtoTypeProxy, // 代理模式

ProtoTypeRequest, // 请求资源

ProtoTypeOperation, // 操作请求

};

SendDataToClient ProtoHead->dwProtoType = ProtoTypeProxy

SendRequestLocal ProtoHead->dwProtoType = ProtoTypeRequest

SendOperationLocal ProtoHead->dwProtoType = ProtoTypeOperation

发送的数据 数据 头+ 数据包

LPINTER\_PROTO\_HEAD ProtoHead=(LPINTER\_PROTO\_HEAD)xBuffer;

ProtoHead->dwClientAddr=dwClientAddr; //客户端IP

ProtoHead->dwCommandId=dwCommandId; //命令ID

ProtoHead->dwDataLen=dwBufferLen; //数据包长度

ProtoHead->dwProtoType=dwProtoType; //发送消息类型

RtlCopyMemory(xBuffer+sizeof(INTER\_PROTO\_HEAD),Buffer,dwBufferLen );

// 与服务交互

//获取配置信息

SendRequestLocal(0, REQUEST\_QUERY\_SERVER\_CONFIG);//请求获取配置信息

//接受的返回消息包

REQUEST\_QUERY\_SERVER\_CONFIG,//查询服务器设置

typedef

struct \_ServerSetConfigInfo

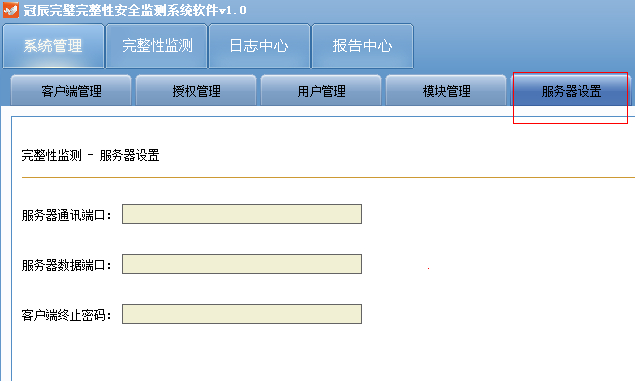
{

DWORD dwDataPort;

DWORD dwCommandPort;

WCHAR szClientPassword[21];

}ServerSetConfigInfo, \*PServerSetConfigInfo;



SendRequestLocal(0, REQUEST\_CLIENT\_IS\_ONLINE, &ip, sizeof(ip));

REQUEST\_CLIENT\_IS\_ONLINE,//查询客户端是否在线

typedef

struct \_ClientIP

{

DWORD dwClientIP;

}ClientIP, \*PClientIP;

//接受的返回数据包

typedef

struct \_ClientConnStatus

{

DWORD dwClientIP;

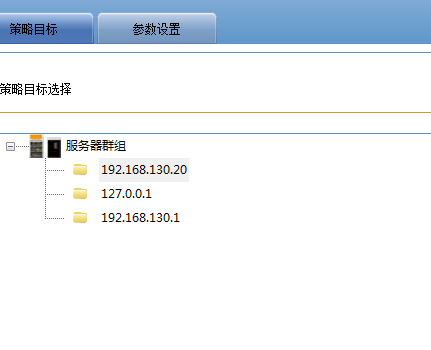
DWORD dwIPStatus;

}ClientConnStatus, \*PClientConnStatus;

//发送运行策略前，检测客户端是否在线



//获取客户端目录前 检测客户端是否在线



SendOperationLocal(0, OPERATION\_SET\_SERVER\_CONFIG, &SConfig, sizeof(SConfig));

OPERATION\_SET\_SERVER\_CONFIG,//更改服务器设置

typedef

struct \_ServerSetConfigInfo

{

DWORD dwDataPort;

DWORD dwCommandPort;

WCHAR szClientPassword[21];

}ServerSetConfigInfo, \*PServerSetConfigInfo;



SendOperationLocal(0, SERVER\_START\_RESTORE, &copyinfo, sizeof(copyinfo));

SERVER\_START\_RESTORE,//开始还原

typedef

struct \_CopyInfo

{

DWORD dwClientIP;//客户端IP

WCHAR szStrategyName[64];//策略名

WCHAR szCopyFilePath[MAX\_PATH];

}CopyInfo, \*PCopyInfo;



SendOperationLocal(0, OPERATION\_CONNECT\_CLIENT, &clientip, sizeof(clientip));

OPERATION\_CONNECT\_CLIENT,//连接客户端

typedef

struct \_ClientIP

{

DWORD dwClientIP;

}ClientIP, \*PClientIP;



SendOperationLocal(0, OPERATION\_DISCONNECT\_CLIENT, &clientip, sizeof(clientip));

OPERATION\_DISCONNECT\_CLIENT,//断开客户端

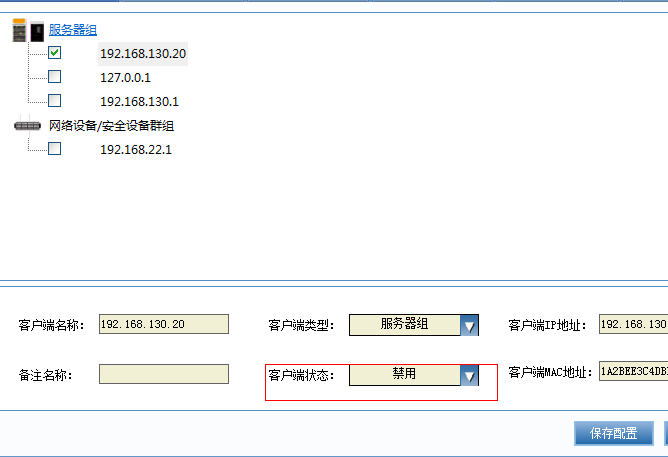
typedef

struct \_ClientIP

{

DWORD dwClientIP;

}ClientIP, \*PClientIP;



// 与客户端交互

// 公用

SendDataToClient(inet\_addr(W2A(m\_csIP.GetBuffer())), SERVER\_LIST\_DRIVE);

SERVER\_LIST\_DRIVE, //列出盘符

\*\*\*\*没有数据包\*\*\*\*\*

//点击客户端IP 获取客户端盘符



//客户端返回数据包

头文件 +数据包

//SendDataToClient(ProtoTypeProxy,dwCommandId,RemoteAddr,Buffer,dwBufferLen);

头文件的dwProtoType = ProtoTypeProxy

<DriverInfo>

<Item Dir="252" />

</DriverInfo>

SERVER\_LIST\_FILEDIRECTORY, //列出目录下所有文件和文件夹

SendDataToClient(0, SERVER\_LIST\_DIRECTORY, &dir, sizeof(dir));

\*\*\*\*没有数据包\*\*\*\*\*

//客户端返回数据包

头文件 +数据包

头文件的dwProtoType = ProtoTypeProxy

//NAME 目录名

<FileDirInfo>

<Item Name="20130725" IsDir="1" />

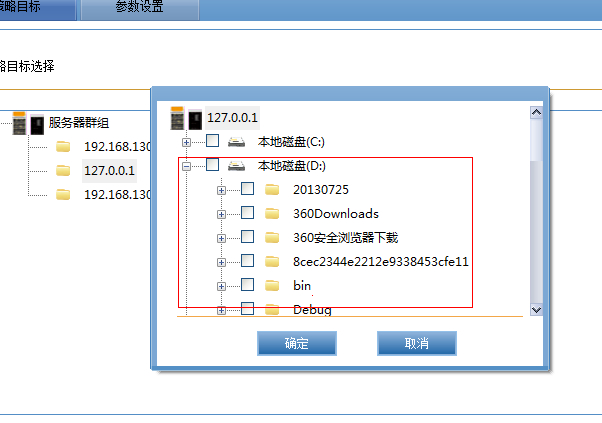
<Item Name="360Downloads" IsDir="1" />

<Item Name="淘宝" IsDir="1" />

<Item Name="签名目录" IsDir="1" />

<Item Name="送检前程序" IsDir="1" />

</FileDirInfo>



//因为之前的结构数据太大了，调整了服务的实现，策略消息统一用数据包统一用STRATEGY\_BUFFERDATA

typedef struct \_STRATEGY\_BUFFERDATA

{

//策略名称

WCHAR StrategyName[64];

}STRATEGY\_BUFFERDATA,\*LPSTRATEGY\_BUFFERDATA;

SendOperationLocal(0, SERVER\_STRATEGYGUARD\_ADD,&strategyBufferData,sizeof(STRATEGY\_BUFFERDATA))

SERVER\_STRATEGYGUARD\_START, //运行策略

SERVER\_STRATEGYGUARD\_PAUSE, //暂停策略

SERVER\_STRATEGYGUARD\_STOP, //停止策略

SERVER\_STRATEGYGUARD\_ADD, //新增策略 （新加）

SERVER\_STRATEGYGUARD\_DELETE, //删除策略

SERVER\_STRATEGYGUARD\_MODIFY, //修改策略

