EasySec Firewall SDK
User Manual

iTinySoft EasySec Team

2002-2003

EasySec Firewall SDK User Manual

EasySec Firewall SDK is a professional software kit for developing network firewall, sniffer or analyser appliations for Microsoft Windows. Simple APIs of EasySec Firewall SDK include powerful functions: Double layer packet filter (application layer and kernel layer) can manage and control data packets of all kinds network protocols quickly and correctly. Application auditing avoids back door program leaking sensitive information, and genearates application filter rule intelligentlly. Particular filter rules of net neighborhood can mangage and control the shared resource, prevent information leaking from local network; Lots kinds of filter rules can achieve your requirement for managing network information..

Use EasySec Firewall SDK to add firewall capabilities to applications that will operate on the internet to ensure that your application is safe from various attacks, and that once identified, an intruder can be blocked from accessing the system without incurring high CPU usage.

Supporting OS:

Windows 98 / Me
Windows NT 4.0
Windows 2000
Windows XP

Features:

Application Programming Interface being encapsulated by DLL is simple and powerful.;

Source code of a personal firewall demo using SDK is open and free.

Engine of SDK provide full functions of a professional personal firewall.

Monitors all applications trying to access the Internet, receive data or send an e-mail.

Shared resources of net neighborhood can be managed and controled for unsafe local network.

Double layer packet filter (application layer and kernel layer) can manage and control data packets of all kinds network protocols quickly and correctly;

Supports filtering of packets both incoming (to the local machine) and outgoing (packets attempting to leave the local machine)

ICMP(PING) packet control protect and hide your IP address.

Allows filters to be set up by specifying ranges of IPs and ports

Allows packet filters to be set up to block all traffic by default, or to let all traffic pass by default; rules then operate against this

Multi-threaded design ensures that high rate of packets filtered does not interfere with the main thread of your application

Alert Assistant provides detailed information to help you choose the best course of action

Modules

Kernel layer driver: ESPFNDIS.VXD or ESPFNDIS.SYS

Application layer hook module: ESPFSPI.DLL

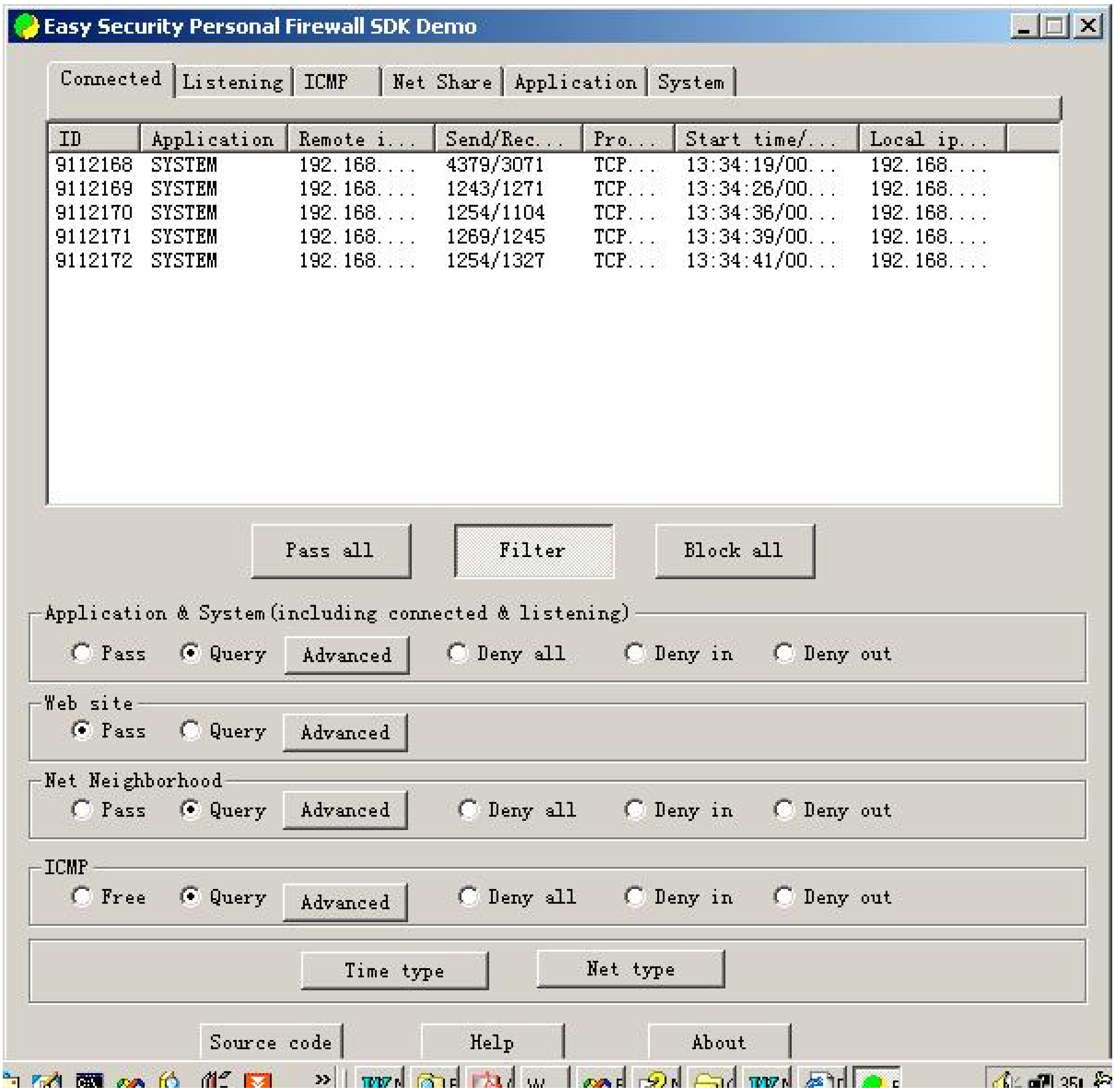
EasySec Firewall API: ESPFSDK.DLL

ESPfSdkR.h API C++ header

ESPfSdk.lib API C++ import library

xacl.cfg Filter rule file

ESPfSdkDemo.exe Demo



Easy Security Personal Firewall SDK Demo

ID	Application	Remote i...	Send/Rec...	Pro...	Start time/...	Local ip...
9112168	SYSTEM	192.168....	4379/3071	TCP...	13:34:19/00...	192.168....
9112169	SYSTEM	192.168....	1243/1271	TCP...	13:34:26/00...	192.168....
9112170	SYSTEM	192.168....	1254/1104	TCP...	13:34:36/00...	192.168....
9112171	SYSTEM	192.168....	1269/1245	TCP...	13:34:39/00...	192.168....
9112172	SYSTEM	192.168....	1254/1327	TCP...	13:34:41/00...	192.168....

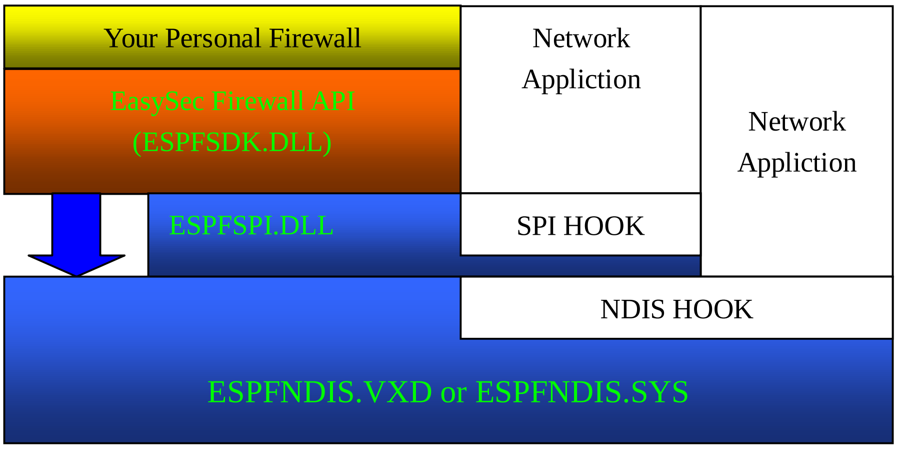
Application & System(including connected & listening)
○ Pass ● Query Advanced ○ Deny all ○ Deny in ○ Deny out

Web site
● Pass ○ Query Advanced

Net Neighborhood
○ Pass ● Query Advanced ○ Deny all ○ Deny in ○ Deny out

ICMP
○ Free ● Query Advanced ○ Deny all ○ Deny in ○ Deny out

ESPfSdkDemo Source code directory of ESPfSdkDemo.exe



How the engine of SDK Works

EasySec Firewall packet filtering intercepts IP packets at the NDIS (Network Device Interface Specification) layer with the driver ESPFNDIS.VXD or ESPFNDIS.SYS and at the SPI(Service Provider Interface) layer with DLL ESPFSPI.DLL . Each packet is checked against the filtering rules that define what kind of traffic is allowed to pass. Allowed incoming packets are forwarded to the TCP/IP stack and the networking applications. Similarly, allowed outgoing packets are sent out on the network interface.

Licensing Requirements

The component is licensed on a CPU basis, and is not Royalty Free. For each developer that will be developing concurrently with EasySec Firewall SDK you must purchase one license. Each license allows one developer to develop with the SDK, as well as deploy the component on one (1) CPU. This means that if you have one developer developing, and are deploying to a dual processor machine (or to two separate machines), you must purchase two licenses. Similarly, if you have two developers developing but are deploying to one CPU, you must still purchase two licenses. Corporate and Site Licenses are available. Please email market@itinysoft.com if you have any questions regarding pricing or licensing. Licenses can be purchased securely online directly from our web site at www.itinysoft.com/easysec/buynow.htm. Licensing terms and conditions are as per the License Agreement.

License Agreement

EASYSEC FIREWALL SOFTWARE DEVELOPMENT KIT

END-USER LICENSE AGREEMENT FOR iTinySoft SOFTWARE TEAM

IMPORTANT-READ CAREFULLY: This End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and iTinySoft Software Team. for the software product identified above , which includes computer software and may include associated media, printed materials and "online" or electronic documentation ("SOFTWARE PRODUCT"). By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, do not install or use the SOFTWARE PRODUCT.

SOFTWARE PRODUCT LICENSE
The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

1. GRANT OF LICENSE. This EULA grants you the following limited, non-exclusive rights:

Software Product. You may install and use one (1) copy of the SOFTWARE PRODUCT to design, develop, and test your software application ("Application"), and then you may deploy to 1 machine (computer). This constitutes one (1) license.

Sample Code. You may modify any sample source code located in the SOFTWARE PRODUCT's "samples" directories ("Sample Code") if provided, to design, develop, and test your Application. You may also reproduce and distribute the Sample Code in object code form only along with any modifications you make to the Sample Code, provided that you comply with the Deployment Requirements described below. For purposes of this section, "modifications" shall mean enhancements to the functionality of the Sample Code.

Deployable Code. You may deploy SDK files ("Deployable Code") to one machine (computer). You may not otherwise copy or redistribute this code. This SOFTWARE PRODUCT is not royalty free.

Deployment Requirements. You may deploy any Sample Code and/or Deployable Code to one machine (collectively "DEPLOYABLE COMPONENTS") as described above, provided that (a) you deploy the DEPLOYABLE COMPONENTS only in conjunction with, and as a part of, your Application; (b) your Application adds significant and primary functionality to the DEPLOYABLE COMPONENTS; (c) you do not permit redistribution of the DEPLOYABLE COMPONENTS; (e) any deployment of Deployable Code is only in conjunction with your Application and includes each and every file contained therein deployed as a single set. The SDK files may not be individually reproduced or distributed.; (f) you include a valid copyright notice on your Application; and (g) you agree to indemnify, hold harmless, and defend iTinySoft and it's distributors from and against any claims or lawsuits, including attorneys' fees, that arise or result from the use or deployment of your Application (h) you do not use the same application names, filenames, or binary compilations as those that are deployed with the SOFTWARE PRODUCT (i) any Sample Code or Deployable Code, whether enhanced and/or modified, may only be deployed in compiled form.

ITinySoft Team. reserves all rights not expressly granted to you.

2. COPYRIGHT. All rights, title, and copyrights in and to the SOFTWARE PRODUCT (including, but not limited to, any names, images, photographs, animations, video, audio, music, text, and "applets"

incorporated into the SOFTWARE PRODUCT) and any copies of the SOFTWARE PRODUCT are owned by iTinySoft Inc. The SOFTWARE PRODUCT is protected by copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE PRODUCT like any other copyrighted material, except that you may either (a) make one copy of the SOFTWARE PRODUCT solely for backup or archival purposes, or (b) install the SOFTWARE PRODUCT on a single computer, provided you keep the original solely for backup or archival purposes. You may not copy printed materials (if any) accompanying the SOFTWARE PRODUCT.

3. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.

LIMITATIONS ON REVERSE-ENGINEERING, DECOMPILATION, AND DISSASSEMBLY. You may not reverse- engineer, decompile, or disassemble the SOFTWARE PRODUCT, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

RENTAL. You may not rent, lease or lend the SOFTWARE PRODUCT.

SOFTWARE TRANSFER. You may permanently transfer all of your rights under this EULA, provided you retain no copies, you transfer all of the SOFTWARE PRODUCT (including all component parts, the media and printed materials, any upgrades, this EULA, and the recipient agrees to the terms of this EULA. If the SOFTWARE PRODUCT is an upgrade, any transfer must include all prior versions of the SOFTWARE PRODUCT.

RUN-TIME DEPLOYMENT. You may deploy the run-time modules of the Software to one (1) computer provided that: (a) you deploy the run-time modules only in conjunction with and as a part of your software product; (b) you include valid copyright notices on your software product; (c) you agree to indemnify, hold harmless, and defend iTinySoft Inc. and its suppliers and distributors from and against any claims or lawsuits, including attorneys' fees, that arise or result from the use or deployment of your software product; and (d) you do not embed the run-time modules in a toolkit which allows users to build and use or distribute applications containing the run-time modules; (e) your Application adds significant and primary functionality to the DEPLOYABLE COMPONENTS. The "run-time modules" refers to the ESPFNDIS.VXD, ESPFNDIS.SYS, ESPFSPI.DLL,ESPFSDK.DLL files that are required for execution of your software program. The run-time modules are limited to run-time files and install files.

TERMINATION. Without prejudice to any other rights, iTinySoft Inc. may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE PRODUCT and all of its component parts.

4. EXPORT RESTRICTIONS. You agree that neither you nor your customers intend to or will, directly or indirectly, export or transmit (a) the SOFTWARE PRODUCT or related documentation and technical data, or (b) your Application as described in Section 1 of this EULA (or any part thereof), or process, or service that is the direct product of the SOFTWARE PRODUCT to any country to which such export or transmission is restricted by any applicable government regulation or statute, without the prior written consent, if required, by such governmental entity as may have jurisdiction over such export or

transmission.

MISCELLANEOUS. If any provision of this Agreement is found to be unlawful, void or unenforceable, then that provision shall be severed from this Agreement and will not affect the validity and enforceability of any of the remaining provisions.

NO WARRANTIES. To the maximum extent permitted by applicable law, iTinySoft Inc. expressly disclaims any warranty for the SOFTWARE PRODUCT. The SOFTWARE PRODUCT and any related documentation are provided "as is" without warranty of any kind, either express or implied, including, without limitation, the implied warranties of merchantability or fitness for a particular purpose. The entire risk arising out of use or performance of the SOFTWARE PRODUCT remains with you.

LIMITATION OF LIABILITY. iTinySoft Inc.'s entire liability and your exclusive remedy under this EULA shall not exceed five dollars ($5.00 CDN).

NO LIABILITY FOR CONSEQUENTIAL DAMAGES. To the maximum extent permitted by applicable law, in no event shall iTinySoft Inc. or its suppliers or distributors be liable for any damages whatsoever (including, without limitation, damages for loss of business profit, business interruption, loss of business information, or any other pecuniary loss) arising out of the use of, or inability to use, this product, even if iTinySoft Inc. has been advised of the possibility of such damages. Because some states/provinces/jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

RIGHT OF PUBLICITY. You agree that iTinySoft Inc. is hereby granted the right to promote SOFTWARE PRODUCT and your use of it in it's online portfolio, it web site, its press kits, its press releases, and any other promotional materials.

EasySec Firewall SDK is a trademark of iTinySoft Inc.

Distribution Requirements

Modules:

Kernel layer driver:

ESPFNDIS.VXD(WIN98/WINME) (win98/system)
ESPFNDIS.SYS (WINNT/WIN2000/WINXP) (winnt/system32)

Application layer hook module: ESPFSPI.DLL
EasySec Firewall API: ESPFSDK.DLL

Registry

WIN98/WinMe
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\XPACKET]

```
"StaticVxD"="espfndis.vxd"
"Start"=hex:00
```

WINNT/WIN2000/WINXP

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\XPacket]
"Type"=dword:00000001
"Start"=dword:00000000
"Group"="Extended Base"
"ErrorControl"=dword:00000001
65, 00, 73, 00, 70,00    66, 00, 6E,00, 64,00, 69,00, 73,00, 2E,00 73,00, 79,00, 73,00
"ImagePath"=hex(2):53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00, 65, 00,\
   73, 00, 70,00    66, 00, 6E,00, 64,00, 69,00, 73,00, 2E,00 73,00, 79,00, 73,00,00,00
"DisplayName"="XFilter Packet"
"DependOnService"=hex(7):4e,00,44,00,49,00,53,00,00,00,00,00
"DependOnGroup"=hex(7):4e,00,44,00,49,00,53,00,20,00,57,00,72,00,61,00,70,00,\
   70,00,65,00,72,00,00,00,00,00
```

Callback function

typedef int (CALLBACK PASCAL * DOACTIONPFORQUERY)(PSESSION pSession);
Query user whether pass or deny this session

Parameters
PSESSION pSession   Pointer to a **SESSION** structure
Return:
pSession->bAction = 0;   //0 pass 1 deny
pSession->bStatus = SESSION_STATUS_FREE; //release this session

typedef int (CALLBACK PASCAL * NOTIFYMONITORSTREAMINFO)(int, PACKET_LOG *);
The EasySec Firewall SDK engine is receiving a packet inside or sending a packet outside, notify this packet information to the user.

Parameters
int   PacketType
MON_STREAM_ICMP   ICMP packet information
MON_STREAM_NNB   Network neighborhood packet information
MON_STREAM_APP   Application packet information
PACKET_LOG * pPacketInfo
Pointer to a **PACKET_LOG** structure

typedef int (CALLBACK PASCAL * NOTIFYMONITORLISTENINFO)(int, PSESSION);
Information of applications listening and waiting for connect outside

Parameters

int   PacketType   Reserved

PSESSION pSession   Pointer to a **SESSION** structure

typedef int (CALLBACK PASCAL * NOTIFYMONITORSESSIONINFO)(int, PSESSION);
Information of applications having connected and communicating

Parameters

int   Type

MON_SESSION_ADD   Add a connected session

MON_SESSION_REMOVE   Remove a connected session

PSESSION pSession   Pointer to a SESSION structure

Function

BOOL ESPfSdkInit(
DOACTIONPFORQUERY   FuncDoActionForQuery,
NOTIFYMONITORSTREAMINFO   FuncNotifyMonitorStreamInfo,
NOTIFYMONITORLISTENINFO   FuncNotifyMonitorListenInfo,
NOTIFYMONITORSESSIONINFO FuncNotifyMonitorSessionInfo)

SDK init and transfer callback function to the engine of SDK

Parameters

DOACTIONPFORQUERY   FuncDoActionForQuery

Callback function DOACTIONPFORQUERY address

NOTIFYMONITORSTREAMINFO   FuncNotifyMonitorStreamInfo

Callback function NOTIFYMONITORSTREAMINFO address

NOTIFYMONITORLISTENINFO   FuncNotifyMonitorListenInfo

Callback function DOACTIONPFORQUERY address

NOTIFYMONITORSESSIONINFO   FuncNotifyMonitorSessionInfo

Callback function NOTIFYMONITORSESSIONINFO address

BOOL ESPfSdkExit()

SDK Exit and release some context of SDK

BOOL ESPfStartMonitor()

FIREWALL engine start monitoring

BOOL ESPfSetWorkMode(unsigned char ucWorkMode);

Set firewall working mode

Parameters

unsigned char ucWorkMode

ACL_PASS_ALL	Do not block any network packets
ACL_QUERY	Normal mode, query the filter rule of firewall

ACL_DENY_ALL	Deny all network packets

int ESPfGetSecurityLevel();

Get current firewall security level

return:

ACL_SECURITY_HIGH	0x00
ACL_SECURITY_NORMAL	0x01
ACL_SECURITY_LOWER	0x02

void ESPfSetSecurityLevel(BYTE bSecurity)

Set current firewall security level

Parameters

ACL_SECURITY_HIGH	0x00
ACL_SECURITY_NORMAL	0x01
ACL_SECURITY_LOWER	0x02

int ESPfEasyGetRule(int RuleType, int *pRuleAction, int *pActionForRule)

Acquire main rule processing information

int ESPfEasySetRule(int RuleType, int RuleAction, int ActionForRule)

Set main rule processing information

Parameters: RuleType , RuleAction

RuleType 0 Application 1 Web site 2 Net neighborhood 3 ICMP

RuleType== 0

RuleAction 0 pass

RuleAction 1 deny linking in

RuleAction 2 deny linkout in

RuleAction 3 deny bilinking

RuleAction 4 According to filter rules

if RuleAction== 4

RuleAction 0 pass 1 deny 2 query

RuleType== 1

RuleAction 0 pass

RuleAction 2 According to filter rules

RuleAction 0 pass 1 deby 2 query

RuleType== 2

RuleAction 0 pass

RuleAction 1 deny visiting your shareing information

RuleAction 2 deny visiting other peoples' shareing information

RuleAction 3 deny all

RuleAction 4 According to filter rules

RuleAction 0 pass 1 deby 2 query

RuleType== 3
RuleAction 0 pass
RuleAction 1 deny linking in
RuleAction 2 deny linkout in
RuleAction 3 deny bilinking
RuleAction 4 According to filter rules

int ESPfAddOneRule(void *pAddRule, int length, int RuleType)
Add a filter rule

Parameters:
void *pAddRule
Pointer to the structure of XACL or XACL_IP or XACL_TIME or XACL_WEB or XACL_NNB or XACL_ICMP
int RuleType, length
ACL_TYPE_ACL: PXACL pAddRule ,length = sizeof(XACL)
ACL_TYPE_DISTRUST_IP: PXACL_IP pAddRule, length = sizeof(XACL_IP)
ACL_TYPE_TRUST_IP: PXACL_IP pAddRule, length = sizeof(XACL_IP)
ACL_TYPE_CUSTOM_IP: PXACL_IP pAddRule, length = sizeof(XACL_IP)
ACL_TYPE_INTRANET_IP: PXACL_IP pAddRule, length = sizeof(XACL_IP)
ACL_TYPE_WEB: XACL_WEB pAddRule, length = sizeof(XACL_WEB)
ACL_TYPE_NNB: XACL_NNB pAddRule, length = sizeof(XACL_NNB)
ACL_TYPE_ICMP: XACL_ICMP pAddRule, length = sizeof(XACL_ICMP)
ACL_TYPE_TIME: XACL_TIME pAddRule, length = sizeof(XACL_TIME)

int ESPfUpdateOneRule(void *pAddRule, int RuleType)
Update a filter rule

Parameters:
void *pAddRule
Pointer to the structure of XACL or XACL_IP or XACL_TIME or XACL_WEB or XACL_NNB or XACL_ICMP
int RuleType
ACL_TYPE_ACL: PXACL pAddRule
ACL_TYPE_DISTRUST_IP: PXACL_IP pAddRule
ACL_TYPE_TRUST_IP: PXACL_IP pAddRule
ACL_TYPE_CUSTOM_IP: PXACL_IP pAddRule
ACL_TYPE_INTRANET_IP: PXACL_IP pAddRule
ACL_TYPE_WEB: XACL_WEB pAddRule

ACL_TYPE_NNB: XACL_NNB pAddRule
ACL_TYPE_ICMP: XACL_ICMP pAddRule
ACL_TYPE_TIME: XACL_TIME pAddRule

int ESPfDelOneRule(DWORD dwRuleId, int RuleType)

Delete a filter rule with the rule id and ruletype

Parameters:

DWORD dwRuleId

It's the first item of structure of XACL or XACL_IP or XACL_TIME or XACL_WEB or XACL_NNB or XACL_ICMP

int RuleType

ACL_TYPE_ACL, ACL_TYPE_DISTRUST_IP, ACL_TYPE_TRUST_IP, ACL_TYPE_CUSTOM_IP, ACL_TYPE_INTRANET_IP, ACL_TYPE_WEB, ACL_TYPE_NNB, ACL_TYPE_ICMP,ACL_TYPE_TIME

PVOID ESPfFindRuleFromId(DWORD dwRuleId, int RuleType)

Find a filter rule pointer through it's id

Parameters:

DWORD dwRuleId

It's the first item of structure of XACL or XACL_IP or XACL_TIME or XACL_WEB or XACL_NNB or XACL_ICMP

int RuleType

ACL_TYPE_ACL, ACL_TYPE_DISTRUST_IP, ACL_TYPE_TRUST_IP, ACL_TYPE_CUSTOM_IP, ACL_TYPE_INTRANET_IP, ACL_TYPE_WEB, ACL_TYPE_NNB, ACL_TYPE_ICMP,ACL_TYPE_TIME

PVOID ESPfGetNextRule(int RuleType, void *pCurrent)

Walk through all filter

Parameters:

int RuleType

ACL_TYPE_ACL, ACL_TYPE_DISTRUST_IP, ACL_TYPE_TRUST_IP, ACL_TYPE_CUSTOM_IP, ACL_TYPE_INTRANET_IP, ACL_TYPE_WEB, ACL_TYPE_NNB, ACL_TYPE_ICMP,ACL_TYPE_TIME

void *pCurrent

pCurrent == NULL find the first filter rule
pCuurent == Current filter rule address, acquire next filter rule

int ESPfSaveRuleConfigFile()

Save all rules to the rule file xacl.cfg

Structure and Macro

PACKET_LOG is the structure of network data info, including type MON_STREAM_ICMP, MON_STREAM_NNB, MON_STREAM_APP

```
typedef struct _PACKET_LOG
{
    BYTE        AclType;
    BYTE        bDirection;
//Direction including(0-4) 0 _T("link in") 1_T("link out") 2_T("Bidirection") 3_T("Broadcast")
4_T("Listen")
    BYTE        bProtocol;
//(0-9) 0_T("any protocol") 1_T("TCP") 2_T("UDP") 3_T("FTP") 4_T("TELNET") 5_T("HTTP")
6_T("NNTP") 7_T("POP3") 8_T("SMTP") 9_T("ICMP")

    BYTE        bAction;  //Action   0_T("Pass"),   1_T("Reject"), 2_T("Query")

    union
    {
        struct
        {
            BYTE    TcpCode       : 6;
            BYTE    Reserved1     : 2;
        };
        struct
        {
            BYTE    TcpFin        : 1;// Link over
            BYTE    TcpSyn        : 1;// attempt to link
            BYTE    TcpRst        : 1;//link init
            BYTE    TcpPsh        : 1;
            BYTE    TcpAck        : 1;//
            BYTE    TcpUrg        : 1;//
            BYTE    SendOrRecv    : 2;// _T("RECV"),   _T("SEND"),   _T("RDSD")
        };
    };
    BYTE        IcmpType;
    BYTE        IcmpSubType;
    BYTE        PacketType;

    DWORD       dwLocalIp;                  //Local IP Address
    DWORD       dwRemoteIp;                 //Remote IP Address
    WORD        wLocalPort;                 //Local port
    WORD        wRemotePort;            //Remote port
    DWORD       tStartTime;                 //Start time
    DWORD       tEndTime;
    DWORD       dwSendData;                 //Send Data(Bytes)
    DWORD       dwRecvData;                 //Receive data(Bytes)
```

```
        TCHAR       sProcessName[MAX_PATH];//Process name and path
        TCHAR       sMemo[MAX_PATH];       //Memo or description
        TCHAR       sLocalHost[64];
        TCHAR       sRemoteHost[64];

} PACKET_LOG, *PPACKET_LOG;
```

The status of item **bStatus** in the structure SESSION

```
#define SESSION_STATUS_FREE                 0
#define SESSION_STATUS_CHANGE           1
#define SESSION_STATUS_OVER                 10
#define SESSION_STATUS_QUERYING_APP    101
#define SESSION_STATUS_QUERYING_WEB  102
#define SESSION_STATUS_QUERY_APP   151
#define SESSION_STATUS_QUERY_WEB 152
#define SESSION_STATUS_QUERY_DRIVER                 200
#define SESSION_STATUS_QUERY_DRIVER_APP         ACL_TYPE_DRIVER_APP            +
SESSION_STATUS_QUERY_DRIVER
#define SESSION_STATUS_QUERY_DRIVER_NNB         ACL_TYPE_NNB                   +
SESSION_STATUS_QUERY_DRIVER
#define SESSION_STATUS_QUERY_DRIVER_ICMP        ACL_TYPE_ICMP                  +
SESSION_STATUS_QUERY_DRIVER
#define SESSION_STATUS_QUERY_MARGIN 50
```

The item **bDirection** in the structure SESSION or all kinds of ACL structure

```
#define ACL_DIRECTION_IN                0
#define ACL_DIRECTION_OUT               1
#define ACL_DIRECTION_IN_OUT            2
#define ACL_DIRECTION_BROADCAST             3
#define ACL_DIRECTION_LISTEN            4
#define ACL_DIRECTION_NOT_SET           255

typedef struct _SESSION
{
        DWORD       dwIndex;
        DWORD       dwPid;
        unsigned int s;//ID
        //SOCKET            s;

        DWORD       dwAclId;

        BYTE        bIsQuery;
        BYTE        bAclType;
        BYTE        bTimeType;
```

```
    BYTE        bNetType;

    BYTE        bStatus;   //SESSION_STATUS_FREE            0
                               //SESSION_STATUS_QUERYING_APP
                               //SESSION_STATUS_QUERYING_WEB
                               //SESSION_STATUS_QUERY_DRIVER_APP
                               //SESSION_STATUS_QUERY_DRIVER_NNB
                               //SESSION_STATUS_QUERY_DRIVER_ICMP
                               //SESSION_STATUS_QUERY_DRIVER

    BYTE        bDirection;
    BYTE        bProtocol;     //0 _T("Any protocol") 1_T("TCP") 2_T("UDP") 3_T("FTP")
4_T("TELNET") 5_T("HTTP")
                               //6_T("NNTP") 7_T("POP3") 8_T("SMTP") 9_T("ICMP")
    BYTE        bAction;

    DWORD       dwLocalIp;                 //Local IP Address
    DWORD       dwRemoteIp;                //Remote IP Address
    WORD        wLocalPort;                //Local port
    WORD        wRemotePort;           //Remote port
    DWORD       tStartTime;                //Start time
    DWORD       tEndTime;
    DWORD       dwSendData;                //Send Data(Bytes)
    DWORD       dwRecvData;                //Receive data(Bytes)
    TCHAR       sPathName[MAX_PATH];       //Application name and path
    TCHAR       sMemo[MAX_PATH];           //Memo or description
} SESSION, *PSESSION;

typedef struct _XACL          XACL,         *PXACL;
typedef struct _XACL_IP           XACL_IP,      *PXACL_IP;
typedef struct _XACL_TIME XACL_TIME, *PXACL_TIME;
typedef struct _XACL_WEB  XACL_WEB,  *PXACL_WEB;
typedef struct _XACL_NNB  XACL_NNB,  *PXACL_NNB;
typedef struct _XACL_ICMP XACL_ICMP, *PXACL_ICMP;
```

XAL is the structure of Application filter rule

```
typedef struct _XACL
{
    DWORD       ulAclID;//ID
    TCHAR       sApplication[MAX_PATH];   //Application name and path

    BYTE        bRemoteNetType;            //Remote net type
    BYTE        bAccessTimeType;       //Access time type
    BYTE        bAction;               //Action for this rule
    BYTE        bDirection;                 //protocol dirction

    BYTE        bServiceType;           //protocol      type      _T("Any      protocol")
_T("TCP")_T("UDP") _T("FTP") _T("TELNET")
                                            //_T("HTTP")    _T("NNTP")    _T("POP3")
_T("SMTP") _T("ICMP")

    BYTE        bReserved[3];           //Reserved

    WORD        uiServicePort;          //Remote port
    WORD        wLocalPort;                 //Local port
    DWORD       dwProcessId;            //Process ID
    TCHAR       sMemo[56];                  //Memo
}XACL,      *PXACL;
```

XAL_IP is the structure of IP filter rule

```
typedef struct _XACL_IP
{
    DWORD       dwId;
    DWORD       ulStartIP;
    DWORD       ulEndIP;

    BYTE        bNotAllowEdit;
    BYTE        bReserved[3];
}XACL_IP,   *PXACL_IP;
```

XAL_ TIME is the structure of time type

```
typedef struct _XACL_TIME
{
    DWORD       dwId;
    DWORD       tStartTime;
    DWORD       tEndTime;

    BYTE        bWeekDay;               //Day of a week
    BYTE        bNotAllowEdit;
    BYTE        bReserved[2];
}XACL_TIME,     *PXACL_TIME;
```

XAL_ WEB is the structure of web site filter rule

```
typedef struct _XACL_WEB
{
    DWORD       dwId;
    TCHAR       sWeb[64];
    BYTE        bAction;
```

```
        BYTE            bReserved[3];
        TCHAR           sMemo[56];
}XACL_WEB,      *PXACL_WEB;
```

XAL_NNB is the structure of network neighborhood filter rule

```
typedef struct _XACL_NNB
{
        DWORD           dwId;
        TCHAR           sNnb[64];
        DWORD           dwIp;

        BYTE            bDirection;
        BYTE            bTimeType;
        BYTE            bAction;
        BYTE            bReserved;

        TCHAR           sMemo[56];
}XACL_NNB,*PXACL_NNB;
```

XAL_ICMP is the structure of ICMP packet filter rule

```
typedef struct _XACL_ICMP
{
        DWORD           dwId;

        BYTE            bNetType;
        BYTE            bDirection;
        BYTE            bTimeType;
        BYTE            bAction;

        TCHAR           sMemo[56];
}XACL_ICMP,     *PXACL_ICMP;
```

Three securtiy level:

```
#define ACL_SECURITY_HIGH               0x00
#define ACL_SECURITY_NORMAL                     0x01
#define ACL_SECURITY_LOWER                      0x02
```

Filter rule type:

```
#define ACL_TYPE_TIME                   0
#define ACL_TYPE_ALL_IP                 1
#define ACL_TYPE_INTRANET_IP            2
#define ACL_TYPE_DISTRUST_IP            3
#define ACL_TYPE_TRUST_IP               4
#define ACL_TYPE_CUSTOM_IP                      5
```

```
#define ACL_TYPE_ACL                6
#define ACL_TYPE_APP                ACL_TYPE_ACL
#define ACL_TYPE_WEB                7
#define ACL_TYPE_NNB                8
#define ACL_TYPE_ICMP               9
#define ACL_TYPE_DRIVER_APP             10
```

PACKET_LOG type

```
#define MON_STREAM_APP     1
#define MON_STREAM_NNB     2
#define MON_STREAM_ICMP    3
```

Connected SESSION processing method

```
#define MON_SESSION_ADD       1
#define MON_SESSION_REMOVE    2
```

Listening SESSION processing method

```
#define MON_LISTEN_ADD        1
#define MON_LISTEN_REMOVE     2
```

1