Firewall Plugin Modules

https://campus.barracuda.com/doc/43847226/

Some applications, as for example FTP, do not use just simple communication between two IPs over well defined ports. An example for this type of service is FTP: After an initial control dialog over port 21, the client and the server use another random port from 1024 through 65535 to send and receive data. The firewall has two possibilities to handle this: either it opens all higher ports, which is not really suitable for a secure firewall, or it listens to the two FTP partners and opens the dynamic port agreed upon in the initial control dialog. The firewall services uses plugin modules to listen for these dynamically allocated ports for the following services:

FTP

- **Protocol Family: TCP, Syntax with parameters: ftp (same port)**
- Using this module indicates that no PAT (Port Address Translation) is performed for ftp data sessions even if the firewall session is NATed. This way it can be guaranteed that the source port for an active FTP data session remains port 20.

For more information, see: How to Use the FTP Plugin Module.

RSH

- **Protocol Family: TCP, Syntax with parameters: rsh**
- The RSH module ensures that rsh works properly.

ICA Browser

- **Protocol Family: UDP, Syntax with parameters: ip-address-1, ip-address-2, ip-address-3, … ip-address-n**
- This module is used for the ICA browser application (mapping, redirecting). The pairs of IPs are mapped: IP/real IP. If no NAT is involved, you must declare the IPs as pairs as well.

Oracle SQL*Net

- **Protocol Family: TCP, Syntax with parameters: ora hostname = ip-address**
- This module is needed when the Oracle SQL*Net application uses dynamic ports. It is also used in the context of destination NAT (mapping, redirecting). The Oracle server usually uses domain name resolution. Hence you must give the IP/name pair to the module.

For more information, see: How to Use the Oracle SQL*Net (ora) Plugin Module.

Trivial FTP

- **Protocol Family: UDP, Syntax with parameters: tftp**
- This module can be used for all UDP applications, which maintain their connection on a different

port than their initial starting port; trivial FTP is the most common example.

For more information, see: How to Use the Trivial FTP Plugin Module.

ONCRPC

- **Protocol Family: UDP & TCP, Syntax with parameters: oncrpc**
- This module is used in context with RPC handling.

For more information, see: How to Use the RPC Plugin Module and How to Configure the ONCRPC Plugin Module.

DCERPC

- **Protocol Family: UDP & TCP, Syntax with parameters: dcerpc**
- This module is used in context with RPC handling.

For more information, see: How to Use the RPC Plugin Module and How to Configure the DCERPC Plugin Module.

Skinny

- **Protocol Family: TCP, Syntax with parameters: -**
- The plugin monitors the skinny signalling connection between the phone and the Cisco callmanager. The default signalling port for SCCP is TCP 2000. When the plugin intercepts a Skinny packet that establishes an RTP connection for an audio transmission for VoIP a pinhole for the voice stream in the firewall will be opened. A call release packet or the termination of the skinny signalling connection closes the pinhole in the firewall.

For more information, see How to Configure VOIP Connections with the Skinny (SCCP) Firewall Plugin; How to Configure the SIP Plugin Module.

SIP

- **Protocol Family: UDP, Syntax with parameters: sip**
- The SIP plugin supports SIP signalling over UDP/IP packets. The default port for SIP signalling connection is UDP port 5060.

For more information, see: How to Configure the SIP Plugin Module.

DNS

- **Protocol Family: UDP, Syntax with parameters: dns**
- The DNS plugin is used to replace the result of a DNS query, according to a predefined IP address translation table.

For more information, see: How to Configure DNS Translation Using the DNS Plugin Module.