NAME

dynfad.conf – Foreign Agent configuration file

DESCRIPTION

Upon execution, **dynfad** reads its configuration information from a configuration file which, by default, is */usr/local/etc/dynfad.conf* .

The configuration file has the following format:

Empty lines and lines starting with ´#´ are comments.

Otherwise a line is of the format "keyword arguments".

The configuration file must end in the keyword **END**.

The following keywords may be used:

INTERFACES

A list of interfaces to be used for Mobile IP services. Each line contains parameters for one interface listed: <Interface> <Type> <Agentadv> <Interval> <Force_IP_addr>. **Interface** is the name of the interface (e.g. eth0) **Type** can be one of the following: **1** = mobile services offered both upper and lower directions. **2** = only upper direction (to upper FA / HA). **3** = only lower direction (to lower FA / MN). Note: Only one interface can be used for upper direction (hierarchy), but multiple interfaces can be used for lower direction. **Agentadv** is either 0 (no broadcast agent advertisement) or 1 (send periodic agent advertisements). **Interval** tells the delay in seconds between the ICMP mobility agent advertisement messages. Its default value is **30**. **Force_IP_addr** is a dotted decimal notation of the interface address. If not entered the primary address of the interface is used. The list must start with a keyword INTERFACES_BEGIN and end with a keyword INTERFACES_END.

NetworkAccessIdentifier

Network Access Identifier (NAI) of this FA that is a unique identifier for this FA. A macro [interface] can be used to get the hardware address of an interface in dot-separated format.

HighestFAIPAddress

The highest Foreign Agent address, given in "dotted decimal notation". Home Agent is expected to communicate with this machine. When running a non-hierarchical FA network, this should be set to the FA's own IP address. This should be the public (i.e., the address that is toward HAs) in a multi-homed FA.

HighestFAPubKeyHash

Hex string. This highest FA public key extension hash code is used to protect the public key from active man-in-the-middle style attacks. Its use is optional, but recommended. If this hash is configured, the FA broadcasts it in the agent advertisements and the MNs send it in their registration requests to the HA (protected with MN-HA authentication extension). The hash code is printed by the rsakeygen utility when the public key is generated and it can also be calculated with rsakeyhash from a key file. If this hash is used, it must be the hash of this organization's highest FA.

UpperFAIPAddress

The parent Foreign Agent address, given in "dotted decimal notation". Foreign Agent communicates with this machine. If this address is same as the own IP address, then this FA is the highest Foreign Agent.

HighestFA

Boolean value < TRUE | FALSE >. HighestFA parameter tells whether this FA is the highest FA (i.e., it does not have any upper FAs and it communicates directly with the Home Agents). Defaults to TRUE.

UDPPort

The UDP port that this FA listens to for registration requests and replies. Defaults to **434**

UpperFAUDPPort

The UDP port that the upper FA listens to for registration requests and replies. Defaults to **434**

HAUDPPort

The UDP port that the HA listens to for registration requests and replies. This is used only in highest FA. Defaults to **434**

RegistrationTTLCheck

RFC 2344 style tunnel hijacking protection requires that the MN uses TTL value 255 on all registration request messages and the FAs check this. Since RFC 2002 compliant MN implementations do not necessarily set the TTL to 255, this may limit the access of those MNs. This option can be used to change the checking of the TTL field in the IP header. **0** = no TTL checking (i.e., accept any value). **1** = check the TTL only in registration requests that ask for reverse tunneling (i.e., the MN should be RFC 2344 compliant and use the TTL 255). **2** = check the TTL on every registration requests (this might deny the access of some MN implementations).

SocketPriority

Socket priority for signaling sockets (UDP) can be set with SO_PRIORITY to allow easier QoS configuration. If this argument is set, the given value is used as a priority for the signaling socket. E.g., CBQ class can be used to make sure that signaling is not disturbed by other traffic on a congested link.

TunnelDevice

The tunnel interface name beginning (e.g. FA may use TUNL0, TUNL1, ...). Defaults to **TUNL**

RoutingTableStart

The start of the range of routing tables that this FA can use. Linux kernel has 256 routing table (0 .. 255), but 0, 254, and 255 are reserved. Defaults to **1**

RoutingTableEnd

The end of the range of routing tables that this FA can use. Defaults to **253**

AUTHORIZEDNETWORKS

A list of pairs of authorized network addresses and netmasks [networkaddress]/[netmask] (one per row) from which other FAs can connect to this FA. The list must start with a keyword: AUTHORIZEDNETWORKS_BEGIN, and end with a keyword: AUTHORIZEDNETWORKS_END.

AllowMobileNodes

Whether this FA allows MNs to connect directly or not (lowest FAs must allow MNs to connect). Defaults to **TRUE**

FA_SECURITY

A Foreign Agent may have optional security associations with other Foreign Agents. If the security association exists the session key can be encrypted with the help of shared secret and thus man-in-the-middle style attacks can be prevented. If no security association is set for a certain Foreign Agent - Foreign Agent pair, public key encryption (RSA) is used. Again keywords FA_SECURITY_BEGIN and FA_SECURITY_END begin and end the list. The required information is available in the example config file of the distribution.

FAKeyFile

The RSA public key file generated with **rsakeygen**(8). Defaults to **/usr/local/etc/dynfad.key**

Mobile IPv4 Challenge/Response (RFC 3012)

EnableChallengeResponse

Dynamics supports Mobile IPv4 Challenge/Response protocol as an optional addition to the Mobile IP registration. This is not used in a default setup because the extensions used for challenges will prevent MNs that do not support this addition from using the FA at all. Setting this item to TRUE enables challenge/response usage.

ChallengeWindow

Number of last advertised challenges that will be accepted (default value given in RFC 3012: 2).

ChallengeLength

Length of the challenge to be used (in bytes; 0 .. 256)

RequireChallenge

Whether the FA requires the challenge to be present in every registration request (if not present, request will be denied with MISSING_CHALLENGE error). If 'EnableChallengeResponse' is TRUE, the challenge is required from MNs which do not have a security association with the FA. With have the security association.

ChallengeInRegReply

Whether the FA adds new challenge to all the registration replies.

RequireMNFASecAssoc

AAA Keys for Mobile IP (draft-ietf-mobile-aaa-key-07.txt): The FA can be configured to deny registration replies that do not have an Unsolicited MN-FA Key Material From AAA extension for an MN that does not have a security association with the FA.

MaxBindings

The number of maximum simultaneous bindings allowed. The value defaults to **20**.

MaxPending

The maximum number of pending registration requests (unconfirmed bindings) the FA is willing to maintain (typically 5 according to rfc2002-bis draft). Additional registrations will be rejected until at least one of the pending registrations has been completed or has timed out. Zero means no limit on pending registration requests.

DeletePendingAfter
The number of seconds after which pending registration requests MAY be deleted. Zero means do not force pending registration request deletion before their requested lifetime has expired.

EnableFADecapsulation
A boolean to tell whether we let the Foreign Agent decapsulate the IP-within-IP encapsulated packets, if a Mobile Node requests such tunneling operation mode. Note! Not fully tested yet. Defaults to **FALSE**.

EnableTriangleTunneling
Triangle tunnel means that the packages to Mobile Nodes are send via the HA, but packages from Mobile Node are routed directly (i.e., FA uses normal IP routing). Defaults to **TRUE**

EnableReverseTunneling
Reverse tunnel means bi-directional tunneling in which both the packages from and to Mobile Node are send via HA. Defaults to **TRUE**

ForceReverseTunneling
Forces FA to use reverse tunneling even if the Mobile Node asks Triangle Tunnel mode.

RegistrationRequired
FA may require registration even from those MNs which have acquired a co-located care-of address. This option selects whether the agent advertisements messages have 'Registration required' flag or not

FADefaultTunnelLifetime
The maximum mobility binding lifetime. This value is the lifetime advertised for this FA. If a request has a larger lifetime value it will be denied. This value should not be greater than any of the maximum lifetimes configured for upper FAs (i.e., best to use the same maximum for whole FA organization). Special case: 65535 (or more) seconds mean unlimited time (the binding will not expire). The default value is **400**.

PacketSocketMode
FA uses a packet socket for raw L2 header access. When sending registration messages, this is only used between the lowest FA and the MN. Current code does not implement fragmentation and packets larget than the used MTU are thus probably dropped. FA can be configured to not use packet socket when sending frames, but this may require broadcast ARP for MN's home address in the foreign network. This is against RFC 2002, so it should be used only if the packet socket does not work. Possible values for PacketSocketMode:
0 = use packet socket when sending registration replies to MN (default)
1 = do not use packet socket at all for sending registration messages

SyslogFacility
The log messages are written through syslog service. The facility to be used defaults to **LOG_LOCAL0**, but it can be set with this parameter to any of the possible facilities (**LOG_AUTHPRIV**, **LOG_DAEMON**, and so on). The processing of log messages is defined in the */etc/syslog.conf* file.

FAAPIReadSocketPath
Path for the read only socket for API calls given as a text string. Read only socket means that through this socket no modifications cannot be made to FA; only statistical data can be read. An example path for read socket: **/var/run/dynamics_fa_read**.

FAAPIReadSocketGroup
Group for the read only socket for API calls given as a text string. For example **root**.

FAAPIReadSocketOwner
Owner for the read only socket for API calls given as a text string. For example **root**.

FAAPIReadSocketPermissions
Read socket permissions given in octal form like in **chmod**(1). For example **0766**.

FAAPIAdminSocketPath
Path for the administration socket. Calls through this socket can modify the dynamic data in FA. For example **/var/run/dynamics_fa_admin**.

FAAPIAdminSocketGroup
Group for the administration socket given as a text string. Look in /etc/group file. For example **root**.

FAAPIAdminSocketOwner
Owner for the administration socket given as a text string. Look in /etc/passwd. For example **root**.

FAAPIAdminSocketPermissions
Permissions for the administration socket given in octal format like in **chmod**(1). For example **0700**.

FILES

/usr/local/etc/dynfad.conf
The system wide configuration file for Foreign Agent.

AUTHOR AND COPYRIGHT

Dynamics group <dynamics@cs.hut.fi>

SEE ALSO

dynfad(8), **dynhad**(8), **dynmnd**(8), **rsakeygen**(8), **dynamics**(8)