isnsadm.conf, isnsd.conf, isnsdd.conf

Fichiers de configuration iSNS

Tous les utilitaires Open-iSNS lisent leur configuration dans un fichier dans /etc/isns. Il y a un fichier par application, isnsd, isnsadm, et isnsdd. Des valeurs de délai acceptent une unité : d (jour), h (heure), m (minute), ou s (seconde).

Options génériques

HostName Par défaut, les applications Open-iSNS récupèrent le nom d'hôte de la machine avec gethostname(3), et utilisent une recherche DNS pour retrouver le nom canonique. Cette option remplace ce mode

SourceName Obligatoire pour toutes les applications Open-iSNS. Devrait être le nom qui identifie le client de manière unique. Il y a 2 lectures de la rfc4171, une exige qun nom iSCSI tel que 'iqn.2001-04.com.example.host', et une autre suggère un format plus simple, tel que fqdn du client.

IQNPrefix Spécifie le préfixe de nom qualifié iSCSI ; doit être sous la forme iqn.YYYY-MM, YYYY étant l'année et MM le mois.

ServerAddress (client) Spécifie le nom d'hôte ou l'adresse du serveur iSNS à contacter.

SLPRegister (serveur) À 1, le service s'enregistre lui-même avec le service SLP. Cela permet aux clients de contacter le serveur sans avoir à configurer d'adresse statiquement.

PIDFile (serveur) Spécifie le nom du fichier pid. Défaut : /var/run/isnsd.pid

Options liée à la base de données

Database Spécifie comment la base est stockée. Vide, la base est conservée en mémoire, sinon, c'est le répertoire où isnsd conserve sa base.

DefaultDiscoveryDomain iSNS définis la visibilité des autres nœuds en utilisant les Discovery Domains. Un nœud de stockage A ne verra que le nœud de stockage B, s'ils sont membres du même domaine de découverte. À 1, indique à isnsd de créer un domaine de découverte virtuel, qui maintient tous les nœuds qui ne font pas partie d'un domaine de découverte configuré administrativement.

RegistrationPeriod Le serveur iSNS peut purger les entités enregistrées après une certaine période d'inactivité. Les clients qui s'enregistrent sont supposés rafraîchir leur enregistrement dans cette période.

ESIRetries Open-iSNS est capable de monitorer l'accessibilité des nœuds de stockage et leur portails en utilisant une fonctionnalité de protocole appelé ESI (Entity status inquiry). Les clients demandent à superviser ESI en enregistrant un port ESI avec chaque portail. Le serveur envoie des messages ESI à ces portails à interval régulier. Si le portail échoue à répondre plusieurs fois, il est considéré mort, et sera supprimé de la base. ESIRetries spécifie le nombre max de tentative. Défaut : 3

ESIMinInterval interval ESI minimum. Défaut : 60 secondes

ESIMaxInterval Interval ESI maximum. Défaut : 10 minutes

SCNRetries Les clients iSNS peuvent enregistrer leur message SCN (State Change Notification) reçus pour connaître les changements dans la base iSNS. Spécifie le nombre de tentatives de transmission d'une message SCN par le serveur avant abandon. Défaut : 3

SCNCallout Chemin du programme helper que isnsdd invoque quand il traite un SCN d'un serveur. Ce programme est invoqué avec un argument indiquant le type d'événement (add, update, ou remove), suivi par une liste d'attributs name=value, en utilisant les noms et conventions décris dans isnsadm.

Options liées à la sécurité

Le standard iSNS définis une méthode d'authentification basée sur l'algorithme DSA. Les participants dans l'échange authentifient les messages en ajoutant un bloc d'authentification contenant un horodatage, une chaîne identifiant la clé utilisée, et une signature numérique du message. La même méthode est également utilisée par SLP.

La chaîne contenue dans la bloc d'authentification est référée au SPI (Security Policy Index). Cette chaîne peut être utilisée par le serveur pour rechercher la clé publique du client ; donc la chaîne peut être utilisée comme nom du fichier de clé publique dans un répertoire, ou pour récupérer un certificat dans LDAP.

Pour les applications clientes Open-iSNS, il y a seulement 2 clés : la clé privée du client, utilisée pour signer les message qu'il envoie au serveur, et la clé publique du serveur, utilisée pour vérifier la signature des messages du serveur.

Le serveur iSNS doit, en plus de sa clé privée, accéder à toutes les clés publiques des clients qui vont communiquer avec lui.

Security Active l'authentification DSA. à 1, le client signe tous les message, et s'attend à des messages serveur signés.

AuthName Chaîne utilisée come SPI dans tous les messages sortants qui ont un bloc auth. Défaut : le nom d'hôte (voir l'option HostName)

AuthKeyFile Chemin de la clé DSA encodé PEM. Défaut : /etc/isns/auth_key

ServerKeyFile (client), fichier contenant la clé publique DSA encodé PEM du serveur. Défaut : /etc/isns/server_key.pub

KeyStore (serveur) spécifie l'emplacement des clés.

Auth.ReplayWindow Pour compenser le décalage d'horloge entre 2 hôtes, Open-iSNS applique un léger flou en comparant les horodatages contenus dans le message et l'horloge système. Défaut : 5m

Auth.TimestampJitter En vérifiant les messages entrant, Open-iSNS vérifie que les horodatages envoyés par le pair sont augmenté monotoniquement. Pour compenser le réordonnancement des messages par le réseaux, un jitter est accepté. Si un horodatage d'un message entrant n'est pas avant cette durée en seconde, le message est accepté. Défaut : 1s

Stockage de clé et stratégie

L'implémentation actuelle supporte 2 types de stockage de clé. Le premier utilise un simple répertoire pour stocker les clés publiques, chaque clé a un fichier PEM et nommé par le SPI du client. Ce type n'est pas recommandé puisqu'il ne stock aucune information de stratégie.

L'approche recommandé est d'utiliser une base de donnée. Celà utilise des objets de stratégie spécifique aux vendeurs pour gérer les chaîne SPI, clé publique, noms d'entité, nom de source et d'autres bits de statégie. La base de données est configurée en définissant l'option KeyStore à la valeur DB :. Les objets de stratégie Open-iSNS ont les attributs suivants, en plus du SPI :

Source Nom de nœud source que le client doit utiliser. Défaut : la chaîne SPI

Functions bitmap détaillant quelles fonctions le client peut invoquer. Les noms de bit correspondent aux noms cours utilisé dans la rfc4711, tel que DevAttrReg, DevAttrQry, etc. Défaut : autorise l'enregistrement, requête et désenregistrement, et SCNRegister

Entity name C'est le nom de l'entité assigné au client. Si définit, un enregistrement par le client n'est pas autorisé avec un autre nom. Si le client s'enregistre sans identifiant d'entité, le serveur le nom donné par la stratégie. Défaut : pas de restriction

Object access Champ de bits de permission d'accès pour chaque type d'objet (lecture, écriture).

Node types Champ de bits décrivant quels types de nœud de stockage un client peut enregistrer. (target, initiator ou control). Défaut : initiator uniquement.

Options liées au réseau

Network.MaxSockets Nombre de connexion entrantes acceptées. Défaut : 1024.

Network.ConnectTimeout timeout d'attente d'établissement de connexion TCP. Défaut : 60s

Network.ReconnectTimeout Si une connextion échoue, délai d'attente avant de retenter. Défaut : 10s

Network.CallTimeout Délai d'attente de timeout d'un appel au serveur iSNS. Défaut : 60s.