Gestion de l'authentification

Sommaire

Problématiques rencontrées avec l'authentification NagVis

Lorsqu'un utilisateur utilise une instance de NagVis, il utilise des identifiants propres à cette installation de NagVis. Il est possible de gérer les utilisateurs et leur droit directement dans NagVis.

Lorsque NagVis est installé pour être utilisé de manière transparente avec Shinken Entreprise, cette fonctionnalité devient un problème puisqu'il devient nécessaire de synchroniser les bases d'utilisateurs de Shinken et de NagVis. Dans Shinken, une base d'utilisateurs est déjà présente.

Pour simplifier la gestion de l'authentification entre NagVis et Shinken, plusieurs modules ont été ajoutés dans NagVis.

Fonctionnement de l'authentification de NagVis

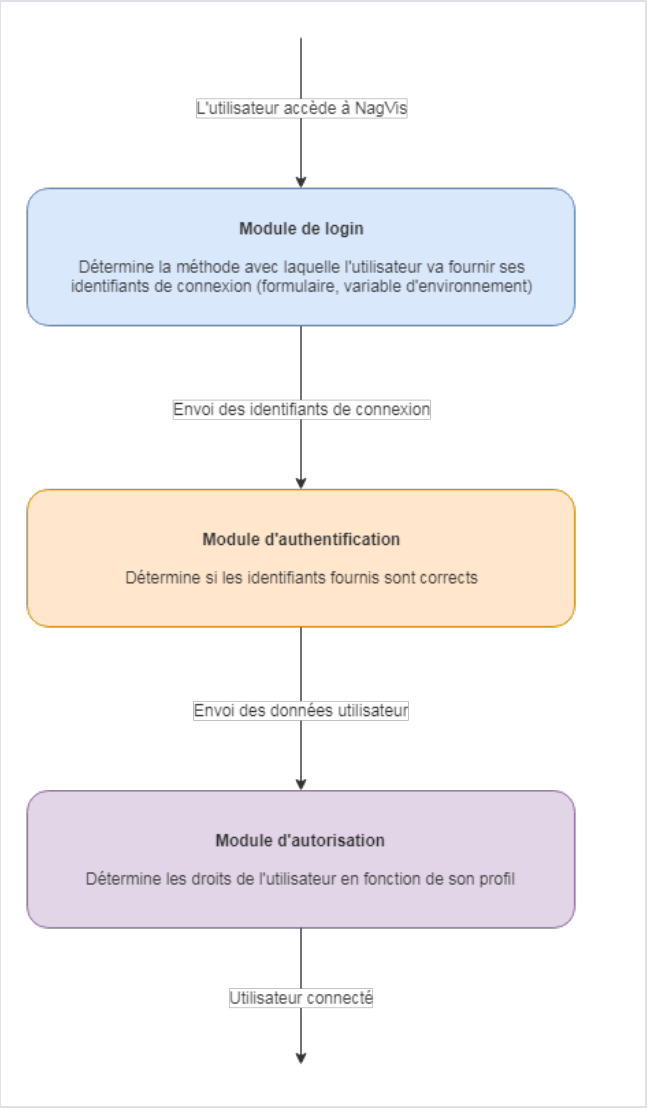
Dans une installation NagVis classique, la gestion des utilisateurs est gérée avec 3 types de modules différents :

- **Un module de login**
  Définit la manière avec laquelle l'utilisateur fournit ses identifiants de connexion. Selon les modules, les identifiants peuvent être passés par un formulaire ou par variable d'environnement.
- **Un module d'authentification**
  Utilise les identifiants de connexion fournis par le module de login et vérifie si ces identifiants sont corrects. Le module d'authentification par défaut vérifie les identifiants de connexion dans la base d'utilisateur propre à NagVis.
- **Un module d'autorisation**
  Utilise les données de l'utilisateur (*son profil et ses réglages*) pour lui attribuer les droits nécessaires (*droits d'administration de NagVis, droits et vue et d'édition des cartes, etc.*)

Les modules par défaut dans l'installation NagVis utilisée pour l'export de l'architecture sont les suivants :

- **Module de login :** LogonShinkenMixed
  Récupère les informations de connexion via des entêtes HTTP. Si aucun entête d'authentification n'est passé, les identifiants de connexion sont récupérés depuis le cookie des interfaces Web Shinken. Sinon, un formulaire de connexion classique est utilisé.
- **Module d'authentification :** CoreAuthModShinken
  Vérifie la validité des identifiants de connexion avec les informations stockées dans la base d'utilisateurs de Shinken.
- **Module d'autorisation :** CoreAuthorisationModShinken
  Définit des droits par défaut (*non modifiables*).

Le fonctionnement de ces modules est décrit de manière détaillée dans les sections suivantes.



Pour plus d'informations sur les modules disponibles par défaut, la documentation NagVis présente un récapitulatif des fonctionnalités disponibles:

- http://docs.nagvis.org/1.9/en_US/index.html

Solutions d'authentification mises en place

Pour permettre une gestion de l'authentification transparente entre Shinken et NagVis, plusieurs modules ont été ajoutés.

Configuration générale

Pour permettre la liaison de l'authentification avec Shinken, les différents modules utilisés pour la connexion ont besoin de savoir quelle est l'adresse de l'installation Shinken avec laquelle il lui faut se connecter.

De manière plus précise, pour se connecter avec Shinken, NagVis utilise le module WebUI (*l'Interface de Visualisation*).

Plusieurs paramètres sont ajoutés pour spécifier l'installation Shinken à contacter :

Nom	Type	Unité	Défaut	Description
`shinken_auth_protocol`	Texte	---	**http**	Protocole à utiliser pour la connexion à Shinken (*http ou https*) Cette valeur est automatiquement renseignée par le module **architecture-export** de l'Arbiter.
`shinken_auth_port`	Entier	---	**7767**	Port de l'interface de Visualisation Shinken à contacter Cette valeur est automatiquement renseignée par le module **architecture-export** de l'Arbiter.

`shinken_auth_address`	Texte	**---**	***Vide***	Adresse du module webui à contacter (*exemple : 192.168.0.3*). Lorsque ce champ est vide, l'adresse du backend par défaut configuré dans NagVis est utilisée. Dans la majorité des cas, il ne sera pas nécessaire de spécifier une adresse particulière. Cette valeur est automatiquement renseignée par le module **architecture-export** de l'Arbiter.
`shinken_auth_restrict_to_shinken_admin`	Booléen	**---**	**1**	Restreint la connexion aux utilisateurs définis comme Administrateurs Shinken dans Shinken
`shinken__authentication__ssl__verify_certificate`	Booléen	**---**	**0**	Activer la vérification du certificat reçu de la WebUI quand elle est configurée en **https** • **0** : non • **1** : oui
`shinken__authentication__ssl__verify_certificate_name`	Booléen	**---**	**1**	Quand la vérification du certificat de la WebUI est activé, vérifier si le nom d'hôte de la WebUI correspond au nom enregistré dans le ceritifcat • **0** : non • **1** : oui Sous CentOS 7 (*ayant une version de PHP < 7*), la vérification du nom du certificat ne fonctionne pas quand ce certificat est dans la chaine de confiance (*paramètre shinken__authentication__ssl__certificate_authority_file*). Il faudra mettre la version de PHP à jour en version 7.2 pour utiliser cette fonctionnalité.
`shinken__authentication__ssl__allow_self_signed_certificate`	Booléen	**---**	**1**	Quand la vérification du certificat de la WebUI est activé, autoriser les certificats auto-signés. • **0** : non • **1** : oui
`shinken__authentication__ssl__certificate_authority_file`	Texte	**---**	***vide***	Définit le certificat d'autorité à utiliser • la valeur système par défaut est "/etc/ssl/certs/ca-bundle.trust.crt" • pour autoriser un certificat auto signé généré pour Shinken, on peut utiliser "/etc/shinken/certs/ca.pem"

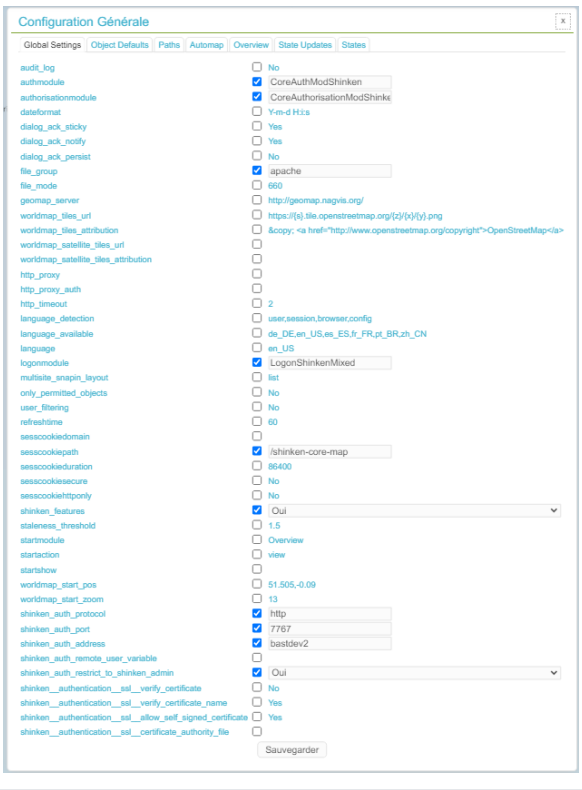
Ces paramètres peuvent être modifiés de 2 manières différentes:

- **Par le fichier de configuration de NagVis**:

/etc/shinken/external/nagvis/etc/nagvis.ini.php

```
; Protocol to use when authenticating with Shinken (http or https) when using the CoreAuthModShinken
authentication module
;shinken_auth_protocol="http"
; Port of broker webui
;shinken_auth_port=7767
; Address of broker webui. If not specified, address of default backend is used instead
;shinken_auth_address=""
; Name of the HTTP header to use to perform SSO authentication with Shinken.
; This value must be the same as the one configured in Shinken. An empty value means authentication by http
header is disabled.
;shinken_auth_remote_user_variable=""
; Authorize authentication into NagVis to Shinken administrators only
;shinken_auth_restrict_to_shinken_admin=1
; Enable verification of SSL cerificate issued by WebUI
;shinken__authentication__ssl__verify_certificate  = 0
; Enable verification of peer name in SSL certificate issued by WebUI
;shinken__authentication__ssl__verify_certificate_name = 1
; Allow WebUI to issue a self signed certificate
;shinken__authentication__ssl__allow_self_signed_certificate = 1
; Set location of certificate authority on local filesystem ( cf. https://www.php.net/manual/en/context.ssl.
php#context.ssl.cafile )
; - system defaults to:
;   - on RedHat / CentOS / Alma / Rocky : "/etc/ssl/certs/ca-bundle.trust.crt"
;   - on Debian : "/etc/ssl/certs/ca-certificates.crt" ; - you can use "/etc/shinken/certs/ca.pem" if you
generated a self signed certificate four your Shinken installation
;shinken__authentication__ssl__certificate_authority_file = ""
```

- **Par l'interface Web de NagVis** (*Options > Configuration générale*).
  Le lien vers l'interface Web de NagVis est présent dans l'interface de visualisation (*menu "Architectures"*), ou directement à l'adresse suivante :
  - http://IP-ARBITER/shinken-core-map/



Modules de login

Utilisation d'entêtes HTTP

Nom du module : CoreLogonShinkenHeader

Lorsque la connexion par SSO est activée dans Shinken (*voir la page* Synchronizer - Authentification unique (SSO)), il est alors possible d'utiliser cette fonctionnalité pour se connecter dans NagVis en utilisant les mêmes entêtes HTTP que ceux utilisés pour Shinken.

L'entête utilisé dans NagVis doit être le même que celui utilisé dans l'interface de Visualisation de Shinken. Pour utiliser l'authentification par SSO dans NagVis avec le module fourni par Shinken, l'authentification par entête HTTP doit également être activée dans Shinken sur l'UI de Visualisation.

Le nom de l'entête contenant le nom d'utilisateur doit être spécifié dans NagVis avec le paramètre "*shinken_auth_remote_user_variable*".

/etc/shinken/external/nagvis/etc/nagvis.ini.php

```
; Name of the HTTP header to use to perform SSO authentication with Shinken.
; This value must be the same as the one configured in Shinken. An empty value means authentication by http
header is disabled.
;shinken_auth_remote_user_variable="X-Remote-User"
```

Lorsque cette variable est vide, l'authentification par entête HTTP est désactivée. Par défaut, l'utilisation des entêtes HTTP est donc désactivée dans NagVis.

Pour des raisons de sécurité, Apache filtre les en-têtes HTTP qui contiennent des caractères invalides, par exemple les underscore (_). Si l'en-tête utilisé contient des caractères underscore (_), elle ne sera pas envoyée à NagVis et l'authentification par en-tête HTTP ne fonctionnera pas.

Utilisation des cookies des interfaces Web Shinken

Nom du module: CoreLogonShinkenCookie

Ce module utilise les cookies des interfaces Web de Shinken Entreprise. Lorsque l'utilisateur est connecté dans une interface (*Interface de Configuration ou Interface de Visualisation*), il sera donc automatiquement connecté dans NagVis. La connexion peut être restreinte aux administrateurs Shinken grâce au paramètre "*shinken_auth_restrict_to_shinken_admin*".

Cette solution ne sera fonctionnelle qui si NagVis est installé sur une machine qui héberge également au moins une interface Shinken (*Con figuration ou Visualisation*). En d'autres mots, cette solution ne sera fonctionnelle qui si le Synchronizer ou un Broker avec le module "webui" sont présents sur la même machine que le démon Arbiter.

Formulaire de connexion

Nom du module: CoreLogonDialog

Ce module est le module par défaut de connexion NagVis. Il affiche un formulaire qui demande à l'utilisateur d'entrer son nom d'utilisateur et son mot de passe pour se connecter dans NagVis.

Lorsqu'il est utilisé avec le module d'authentification "*CoreAuthModShinken"*, les identifiants entrés dans le formulaire seront vérifiés avec Shinken.

Agrégation des modules précédents

Nom du module: CoreLogonShinkenMixed *(par défaut)*

Ce module rassemble les différents modes de connexion précédents en un seul module. Le fonctionnement du module est le suivant :

- Tentative de connexion en utilisant les entêtes HTTP
- Si l'authentification avec entête HTTP échoue (*utilisateur invalide, entête HTTP non défini*), le module tente de connecter l'utilisateur en utilisant le cookie des interfaces Web Shinken.
- Si l'authentification par cookie a échoué, le formulaire de connexion est présenté à l'utilisateur.

Modules d'authentification

Authentification avec Shinken Entreprise

Nom du module : CoreAuthModShinken *(par défaut)*

Ce module utilise les données de connexion fournies par le module de login et vérifie auprès de Shinken (*en particulier l'Interface de Visualisation*) si les identifiants correspondent bien à un utilisateur Shinken existant.

Le comportement de ce module est configurable avec les variables définies dans la section précédente sur la configuration générale des modules d'authentification.

Modules d'autorisation

Définition des droits selon le profil Shinken

Nom du module : CoreAuthorisationModShinken *(par défaut)*

Ce module définit des droits par défaut pour un utilisateur connecté avec Shinken. Ces droits, non configurables, sont les suivants :

- Visualisation en lecture seule de toutes les cartes définies
- Visualisation en lecture seule des rotations définies dans NagVis (*plus d'information sur les rotations dans la documentation NagVis :* http://docs.nagvis.org/1.9/en_US/nagvis_config_format_description.html#rotation)
- Modification autorisée de la configuration générale de NagVis

Définition des droits selon les groupes d'utilisateurs

Nom du module: CoreAuthorisationModShinken

Ce module est très similaire au module "*CoreAuthorisationModGroups*" présent dans une installation NagVis classique. Il permet de définir les droits utilisateurs en fonction des groupes d'utilisateur Shinken dans lequel l'utilisateur est présent.

La configuration des droits se fait grâce au fichier perms.db présent par défaut dans "*/etc/shinken/external/nagvis/etc/*".

Le chemin de ce fichier est configurable dans la configuration de NagVis avec le paramètre "*authorisation_group_perms_file*".

Dans l'exemple, les groupes sont répartis comme suivant:

- Les utilisateurs du groupe "*admins*" ont les droits administrateur dans NagVis
- Les utilisateurs du groupe "*it_admins*" ont accès en lecture et écriture sur toutes les cartes
- Les utilisateurs du groupe "*users*" ont accès en lecture seule à toutes les cartes
- Les utilisateurs du groupe "*users_site1*" ont accès en lecture et écriture seulement sur les cartes "*site1*" et "*site1_bis*"

/etc/shinken/external/nagvis/etc/perms.db

```
{
  "admins": {
    "admin": 1
  },
  "it_admins": {
    "view": [ "*" ],
    "edit": [ "*" ]
  },
  "users": {
    "view": [ "*" ]
  },
  "users_site1": {
    "view": [ "site1", "site1_bis" ],
    "edit": [ "site1", "site1_bis" ]
  }
}
```

La différence de ce module avec celui livré par défaut dans NagVis (*CoreAuthorisationModGroups*) se situe sur le paramètre "admin".

Dans ce module, "admin: 1" aura pour effet de donner tous les droits à l'utilisateur, sauf les droits des gestions des utilisateurs, puisque ceux-ci sont gérés dans Shinken et non dans NagVis.