

Process SNMP (windows)

Sommaire

- [Contexte](#)
- [Exemple](#)
 - [Exemple de résultat](#)
- [Données et métriques](#)
 - [Données](#)
 - [Métriques](#)

Exemple d'utilisation du module

Création d'un check passif

Dans l'interface de configuration, il faut :

- Créer un check passif à associer à un hôte, non actif et volatile, car si nous souhaitons recevoir une notification dès un changement d'état. L'expiration de l'état de fraîcheur doit également être paramétrée.
- Appliquer le modèle d'hôte à un hôte accessible sur le réseau (*le paramètre "adresse" doit être rempli avec un nom d'hôte ou IPv4 accessible*).

Voici un exemple de fichier de configuration d'un check passif et d'un modèle d'hôte :

```
define service{
    service_description      TRAP
    check_command             check-host-alive
    host_name                 TRAP-modele
    is_volatile               1
    passive_checks_enabled   1
    active_checks_enabled    0
    check_freshness          1
    freshness_threshold       300
    register                  0
    check_interval            1
    retry_interval            1
}

define host{
    name                     TRAP-modele
    register                  0
}
```

Maintenant, il faut pousser ces éléments en production afin de pouvoir observer le changement d'état de notre hôte.

Création d'un script pour envoyer des traps SNMP

- Maintenant, il faut un script qui va se charger d'écrire les futurs traps SNMP dans le pipe précisé dans la configuration du module (*dans notre cas, /var/lib/shinken/shinken.cmd*).
- Ajouter le script suivant que l'on appellera "submit_check_result" dans le dossier des plugins Shinken (*/var/lib/shinken-user/libexec/*):

```
#!/bin/bash

# Arguments:
# ${1} = host_name (Short name of host that the service is associated with)
# ${2} = svc_description (Description of the service)
# ${3} = return_code (An integer that determines the state of the service check, 0=OK, 1=WARNING, 2=CRITICAL,
3=UNKNOWN).
# ${4} = plugin_output (A text string that should be used as the plugin output for the service check)

# Ensuring we use the correct commands by using their full absolute path
echo cmd="/bin/echo"
commandfile="/var/lib/shinken/shinken.cmd"

# get the current date/time in seconds since UNIX epoch
datetime="$(date +%s)"

# create the command line to add to the command file
cmdline="[${datetime}] PROCESS_SERVICE_CHECK_RESULT;${1};${2};${3};${4}"

# append the command to the end of the command file
${echo cmd} "${cmdline}" >> "${commandfile}"
```



Ne pas oublier de rendre le script exécutable et de donner les droits du fichier à l'utilisateur Shinken :

```
chown shinken:shinken /var/lib/shinken-user/libexec/submit_check_result
chmod +x /var/lib/shinken-user/libexec/submit_check_result
```

Pour tester le script et simuler une réception d'un trap traduit au format Shinken, il suffit d'exécuter la commande suivante qui va faire passer le service en état critique :

```
/var/lib/shinken-user/libexec/submit_check_result TRAP 2 "test envoi trap - CRITIQUE"
```

Le check devrait passer en critique, et si au bout de la période du seuil de fraîcheur, aucun nouveau trap n'a été reçu, alors la commande check-host-alive fera repasser le check à OK (*si bien sûr l'hôte est accessible via le réseau*).