

Script d'interprétation des traps avec le module named-pipe

Sommaire

[Script interpréteur des traps](#)
[Pour un hôte](#)
[Pour un check](#)

Voir la mise en place de la configuration du module : [Module named-pipe](#)

Script interpréteur des traps

Le script va se charger d'interpréter les futurs traps SNMP reçus pour les envoyer à Shinken.

Pour un hôte

Ajouter le script suivant que l'on appellera **submit_host_result_to_receiver** 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)
# ${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
echocmd="/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="[${datetime}] PROCESS_HOST_CHECK_RESULT:${1};${2};${3}"

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

On le rend exécutable et on le donne à l'utilisateur Shinken

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

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

```
/var/lib/shinken-user/libexec/submit_host_result_to_receiver HÔTE 2 "test envoi trap - CRITIQUE"
```



Les arguments sont :

- \$1 = Le nom de la machine concerné par la trap,
- \$2 = Le code de retour (`0=OK, 1=WARNING, 2=CRITICAL, 3=UNKNOWN`),
- \$3 = Un message texte correspondant à la sortie de la commande.

L'hôte 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*).



Attention

Si vous ne voyez pas votre hôte passer en critique, il est possible que vous n'avez pas désactivé la vérification active sur votre hôte. Pour modifier cela, rendez-vous dans l'interface de configuration, cliquez sur votre hôte et dans supervision, mettez à Faux l'actif activé comme sur l'image ci-dessous :

Staging > Hôte

Hôte > Validé disco_switch

Propriété	Valeur
Période de maintenance planifiée	-- Par défaut [Aucun] --
Vérification du statut de l'élément (ACTIF et PASSIF peuvent être combiné)	
Actif (Les commandes de vérifications sont ordonnancées et lancées par Shinken)	
Actif activé	<input checked="" type="checkbox"/> Vrai <input type="checkbox"/> Faux <small>Par défaut [Vrai]</small>
Vivant (Commande de vérification)	-- Par défaut [check-host-alive (ping)] --

Pour un check

Ajouter le script suivant que l'on appellera **submit_check_result_to_receiver** 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
echocmd="/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
${echocmd} "${cmdline}" >> "${commandfile}"
```

On le rend exécutable et on le donne à l'utilisateur Shinken

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

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_to_receiver HÔTE CHECK 2 "test envoi trap - CRITIQUE"
```

i Les arguments sont :

- \$1 = Le nom de la machine concerné par la trap,
- \$2 = Le nom du check (*doit correspondre au nom donnée dans la définition du check Shinken. dans cet exemple : TRAP*),
- \$3 = Le code de retour (*0=OK, 1=WARNING, 2=CRITICAL, 3=UNKNOWN*),
- \$4 = Un message texte correspondant à la sortie de la commande.

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).

! **Attention**

Si vous ne voyez pas votre check passer en critique, il est possible que vous n'ayez pas désactivé votre check en mode actif. Pour modifier cela, rendez-vous dans l'interface de configuration, cliqué sur votre check et dans supervision, mettez à Faux l'actif activé comme sur l'image ci-dessous :

The screenshot shows the configuration page for a host check in Nagios Core. The breadcrumb is "Staging > Check appliqué au modèle d'hôte". The page title is "Check appliqué au modèle d'hôte > CPU SNMP". The "Général" tab is active, showing a table with columns "Propriété" and "Valeur". The "Actif activé" property is set to "Faux".

Propriété	Valeur
Période de maintenance planifiée	-- Par défaut [Même comportement que son hôte] --

Vérification du statut de l'élément (ACTIF et PASSIF peuvent être combiné)

Actif (Les commandes de vérifications sont ordonnancées et lancées par Shinken)

Actif activé **Faux** Forcé par défaut: [Vrai] Pas d'héritage