

Ter itens não suportados no Zabbix é um problema que comumente enfrentamos, pois em muitos casos um determinado template pode possuir sensores não suportados por nossos devices.

O que não é um problema a nível de usuário que apenas usa a ferramenta, agora quando pensamos em performance isso muda completamente o cenário.

Bem então o que devemos fazer? Desabilitar estes itens correto?

A resposta é sim.

Mas observe o seguinte caso desabilitar manualmente pode ser uma solução quando falamos de poucos hosts, agora imagine que a ferramenta tenha 5000 hosts ativos, quanto tempo levaria para entrar em cada um destes e desativar o que não foi suportado?

Seria algo muito inviável.

“Pessoal somente um adendo que foi passado por um dos melhores especialistas Zabbix do país. É interessante tratar os itens suportados, desabilitar na minha visão são em casos extremos, se o device não suporte uma ou outra OID, a solução é filtrar via regexp esse ambiente para não criar os itens ou dividir em templates.

Por Jorge Pretel”

Então que tal criar um simples script em Python para resolver toda essa situação?

Provavelmente você deve se perguntar, mas é possível fazer isso? Minha resposta é sim, o Zabbix permite que diversas ações administrativas sejam realizadas por sua API o que permite em muitos casos um ganho de tempo incrível assim como customizações para facilitar a o máximo a vida do administrador.

Partindo para o código:

Você vai precisar do seu editor preferido para trabalhar com Python, aqui estou usando o Visual Studio Code (ferramenta gratuita e fácil de usar).

Python 3.0

Biblioteca ZabbixAPI (pode efetuar o download em:  
<https://pypi.org/project/zabbix-api/> ou instalar a mesma usando o comando pip # pip install zabbix-api).

O primeiro bloco temos os imports necessários:

```
# -*- coding: utf-8 -*-
from zabbix_api import ZabbixAPI
from datetime import datetime
import urllib.request as urllib2
import time
```

Um print que mostra a versão da ferramenta

No segundo temos o endereço do servidor Zabbix o usuário e senha que serão usados.

E logo em seguida a variável resultado que recebe os valores de zapi.item.get

Observe que na solicitação estou deixando o groupids definido isso porque quero apenas desabilitar valores não suportados neste grupo.

Caso deseje executar para todo o parque basta omitir essa linha (comentar).

Abaixo temos a opção “monitored”: True (ela por si só já tem o significado

declarado, que é apenas verificar hosts ativos no Zabbix). Por fim temos um laço que varre o resultado e imprime no console o id dos itens desabilitados.

```
zapi = ZabbixAPI("https://seu_servidor_zabbix")
# informações de acesso
zapi.login("usuário_zabbix", "senha_do_usuario")
# mostra versão do zabbix
print("Connected to Zabbix API Version %s" % zapi.api_version())

resultado = zapi.item.get({
    "groupids": "38", # O grupo deve ser informado nesta parte do código.
    "filter": {
        "state": 1
    },
    "output": [
        "hostid",
        "name",
        "itemid"
    ],
    "monitored": True
})

for x in resultado:
    zapi.item.update({'itemid': x['itemid'], 'status': 1})
    print(x['itemid'])
```

Se desejar poderá incluir outros dados na saída para uma melhor identificação.

Então com estas poucas linhas um trabalho que poderia ser muito oneroso se torna extremamente simples.

Espero ter ajudado com essa pequena dica.

E logo estaremos de volta com mais novidades.