Coletando dados utilizando a API do Zabbix com PHP e JSON

INTRODUÇÃO:

O Zabbix possui uma limitação quanto à montagem de painéis a serem utilizados por pessoas que não estão acostumadas a utilizar a sua interface, como operadores de monitoramento e gerentes.

Pensando nisso, comecei a montar painéis de monitoramento utilizando a API do Zabbix com PHP e JSON, como o exemplo abaixo.

Neste artigo irei demonstrar como coletar o último valor de um determinado item.

UTILIZAÇÃO DA API PARA COLETA DOS DADOS:

De acordo com o manual do Zabbix, para pegar a última informação coletada para o item 123456, deveríamos fazer uma chamada JSON para a API, como o exemplo abaixo:

```
{

    "jsonrpc": "2.0",

    "method": "history.get",

    "params": {

        "output": "extend",    //   Obrigatório. A saída só pode ser alterada para host e
                               // trigger, para os demais tem que ser 'extend'.

        "history": 0,          //   Tipo de valor a ser retornado. O padrão é 3 - inteiro.

        "itemids": "123456",   //   Itemid do item desejado.

        "sortfield": "clock",  //   Campo de ordenação, neste caso a data da coleta.

        "sortorder": "DESC",   //   Tipo de ordenação.

        "limit": 1             //   Limite na quantidade dos valores coletados.
```

```
    },


    // Campos referentes à autenticação da API
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}
```

A saída para esta chamada seria algo como:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "123456",
            "clock": "~~#00f:1351090996~~",
            "value": "0.0850",
            "ns": "563157632"
        }
       ],
       "id": 1
}
```

O exemplo acima não pode ser aplicado diretamente porque a API necessita que a requisição seja feita por POST.

Assim, necessitamos de alguma linguagem de programação, como o PHP, que passe o post no formato JSON para fazer a requisição desejada.

UTILIZANDO PHP PARA FAZER A REQUISIÇÃO ACIMA.

Em primeiro lugar, devemos ter um usuário no Zabbix para ser utilizado pela API. Este usuário deve ter permissão para utilizar o frontend e poderes suficientes para fazer as consultas/alterações desejadas.

A seguir, temos o código comentado para fazer a chamada à API do Zabbix, que deverá ser colocado em um arquivo à parte, por exemplo "coleta_ultimo_valor.php".

```
<?php

/*--------------------------------------------------------------------------------

Função para executar o JSON no Zabbix

Ela espera três parâmetros:

$query:  Parte do post a ser enviado para o Zabbix que contém os
     Parâmetros de execução da API.

$method: Qual é a requisição que será utilizada. Neste exemplo utilizamos
     user.login para a autenticaÃ$ e history.get para recuperar o
     valor do item.

$auth:   Token de autenticação.

--------------------------------------------------------------------------------*/

function execJSON($query,$method,$auth) {

    try {

        // URL do Zabbix - Esse campo precisa ser alterado
```

```
    $url = "http://SEU_SERVIDOR/zabbix/api_jsonrpc.php";

/* Monta o post para a API do Zabbix
Para utilizar o JSON com PHP, substituímos {} por array associativo.
Este array será posteriormente convertido para o formato JSON por uma
função interna do PHP (a partir da versão 5.2). Desta forma não corremos
o risco de passar caracteres incorretos durante a utilização do JSON.  */
    $post = array('jsonrpc' => '2.0',
                  'method' => $method,
                  'params' => $query,
                  'auth' => $auth,
                  'id' => 1
            );

    // Utiliza a função json_encode() para converter o post para o padrão JSON.
    $data = json_encode($post);

    // Pega o caminho do CURL
    $curl = exec("which curl");

/*    Monta a string para execução do CURL, onde:
$curl = caminho do curl;
-X POST = define o método de requisição a ser passado para o servidor.
```

```
    Neste caso, o POST

    -H 'Content-Type: application/json' = Define o cabeçalho (header) da requisição.
                      Neste caso, estamos informando que o tipo
                      de informação é do tipo JSON.

    -d '$data' = define quais dados serão enviados para a API.

    $url = Endereço da API do Zabbix.     */

        $curlStr = "$curl -X POST -d '$data' -H 'Content-Type: application/json' '$url'";



        // Executa o CURL

        $execCurl = exec($curlStr);



    /*   Pega a saída da execução do CURL e decodifica de JSON para o padrÃ£do PHP, ex.:

    JSON: {"jsonrpc":"2.0","result":"5b97745abb3274383719ea854f176b77","id":1}

    PHP: stdClass Object ( [jsonrpc] => 2.0 [result] => 5b97745abb3274383719ea854f176b77 [id] => 1) */

        $curlOutput = json_decode($execCurl);



        // Retorna o resultado do CURL, apresentando apenas o campo 'result'.

        return $curlOutput->result;



    } catch (Exception $e) {

      echo "Exceção pega: ",  $e->getMessage(), "\n";

    }
```

```
}


/*-------------------------------------------------------------------------------
    PRINCIPAL
-------------------------------------------------------------------------------*/


/* Monta o post com o usuário e senha para autenticação no Zabbix
   Esses campo precisam ser alterados */
$queryAuth = array('user' => 'admin',
                   'password' => 'ZABBIX'
                  );


/* Executa a Query para efetuar o login no Zabbix o retorno é uma string de 32 caracteres.
   Padrão para token de autenticação do Zabbix. */
$auth = execJSON($queryAuth,'user.login','');



// Verifica se a autenticação retornou uma string de 32 caracteres.
if (strlen($auth) === 32) {


    /* Hora local menos 10 minutos no formato EPOCH
    A utilização de um período de tempo definido melhora o desempenho da busca no banco de dados. */
```

```
$horaInicial = time() - 600;


/* Item a ser pesquisado normalmente utilizo get: $_GET["item"]
   Esse campo precisa ser alterado */

$item = 123456;


// Monta o post para a busca de host.
$queryGetHistory = array('output' => 'extend',
                         'itemids' => $item,
                         'history' => 0,
                         'sortfield' => 'clock',
                         'sortorder' => 'DESC',
                         'time_from' => $horaInicial,
                         'limit' => 1
                        );


// Executa a query de busca de host
$outputGetHistory = execJSON($queryGetHistory,'history.get',$auth);


/* Executa uma sequência de comandos para cada item encontrado. No caso, exibe o valor coletado.
```

```
        Tratamos como array porque o resultado de uma consulta dessas é um array com cada item retornado.*/

    foreach($outputGetHistory as $history){

        echo "<br>Valor: $history->value";

    }


}

?>
```

Criado por: Roberto Flavio Rezende