

## 3 Verificações passivas e ativas

### Visão geral

Esta seção fornece detalhes sobre as verificações passivas e ativas utilizando o Zabbix Agent.

O Zabbix utiliza o protocolo JSON para se comunicar com o Zabbix Agent.

Existem algumas definições detalhadas nos protocolos utilizados pelo Zabbix:

```
<HEADER> - "ZBXD\x01" (5 bytes)
<DATALEN> - tamanho do dado (8 bytes). 1 será formatado como
01/00/00/00/00/00/00/00 (oito bytes em HEX, número de 64 bit)
```

Para evitar o potencial esgotamento de memória o Zabbix Server é limitado a aceitar no máximo 128MB em uma mesma conexão através do protocolo.

### Verificações passivas

Uma verificação passiva é uma requisição simples de dados. O Zabbix Server/Proxy solicita algum dado (por exemplo a carga de CPU) e o Zabbix Agent retorna com o resultado.

### Requisição do servidor

```
<item key>\n
```

### Resposta do agente

```
<HEADER><DATALEN><DATA> [\0<ERROR>]
```

No exemplo de resposta acima, a parte enviada entre colchetes é opcional e só é enviada para itens (chaves) não suportadas.

Exemplo de fluxo com item suportado:

1. Servidor abre uma conexão TCP
2. Servidor envia **agent.ping\n**
3. Agente lê a requisição e responde com **<HEADER><DATALEN>1**
4. Servidor processa o dado para receber o valor ('1' neste exemplo)
5. Conexão TCP é fechada

Exemplo de fluxo para item não suportado:

1. Servidor abre uma conexão TCP

2. Servidor envia **vfs.fs.size[/nono]\n**
3. Agente lê a requisição e responde com  
**<HEADER><DATALEN>ZBX\_NOTSUPPORTED\0Cannot obtain filesystem information:  
[2] No such file or directory**
4. Servidor processa o dado, modifica o estado do item para 'não suportado' com a mensagem de erro retornada
5. Conexão TCP é fechada

## Verificações Ativas

As verificações ativas requerem um processamento mais complexo. Primeiramente o agente deve obter do servidor a lista com os itens a serem processados de forma independente.

Os servidores que irão fornecer a lista de verificações ativas deverão estar listados no parâmetro 'ServerActive' do arquivo de [configuração do agente](#). A frequência de consulta à estas verificações é definida pelo parâmetro 'RefreshActiveChecks' neste mesmo arquivo. Caso uma requisição da lista de itens ativos falhe, uma nova tentativa será feita a cada 60 segundos com aquele servidor.

O agente envia os dados periodicamente aos servidores.

### Obtendo a lista de itens

### Requisição do agente

```
<HEADER><DATALEN>{  
  "request":"active checks",  
  "host":"<hostname>"  
}
```

### Resposta do servidor

```
<HEADER><DATALEN>{  
  "response":"success",  
  "data":[  
    {  
      "key":"log[/home/zabbix/logs/zabbix_agend.log]",  
      "delay":30,  
      "lastlogsize":0,  
      "mtime":0  
    },  
    {  
      "key":"agent.version",  
      "delay":600,  
      "lastlogsize":0,  
      "mtime":0  
    }  
  ]  
}
```

```
    },
    {
      "key": "vfs.fs.size[/nono]",
      "delay": 600,
      "lastlogsize": 0,
      "mtime": 0
    }
  ]
}
```

O servidor deve responder com sucesso. Para cada item retornado todas as propriedades (**key**, **delay**, **lastlogsize** e **mtime**) deverão estar presentes, mesmo que o item não seja de log.

Exemplo do fluxo:

1. Agente abre uma conexão TCP
2. Agente requisita a lista de verificações ativas
3. Servidor responde com uma lista de itens (item key, delay, lastlogsize e mtime)
4. Agente processa a resposta
5. Conexão TCP é fechada
6. O agente coleta de forma periódica os dados

Note que dados sensíveis de configuração podem se tornar disponíveis para terceiros que tenham acesso à porta de 'trapper' do servidor Zabbix, quando for utilizada a verificação ativa. Isso é possível porque qualquer um pode se passar por um agente ativo e solicitar dados de configuração de itens; não ocorre autenticação neste caso.

### Enviando os dados coletados

### O agente envia os dados

```
<HEADER><DATALEN>{
  "request": "agent data",
  "data": [
    {
      "host": "<hostname>",
      "key": "agent.version",
      "value": "2.4.0",
      "clock": 1400675595,
      "ns": 76808644
    },
    {
      "host": "<hostname>",
      "key": "log[/home/zabbix/logs/zabbix_agentd.log]",
      "lastlogsize": 112,
      "value": " 19845:20140621:141708.521 Starting Zabbix Agent
[<hostname>]. Zabbix 2.4.0 (revision 50000).",
      "clock": 1400675595,
      "ns": 77053975
    }
  ]
}
```

```
    },  
    {  
      "host": "<hostname>",  
      "key": "vfs.fs.size[/nono]",  
      "state": 1,  
      "value": "Cannot obtain filesystem information: [2] No such file  
or directory",  
      "clock": 1400675595,  
      "ns": 78154128  
    }  
  ],  
  "clock": 1400675595,  
  "ns": 78211329  
}
```

## O servidor responde

```
<HEADER><DATALEN>{  
  "response": "success",  
  "info": "processed: 3; failed: 0; total: 3; seconds spent: 0.003534"  
}
```

Se o envio de alguns itens falhar no lado do servidor (por exemplo, por que o host ou o item estão desativados ou excluídos), o agente não irá tentar enviar novamente estes dados.

Exemplo de fluxo:

1. O agente abre a conexão TCP
2. O agente envia a lista de valores
3. O servidor processa os dados e envia o status de volta
4. A conexão TCP é fechada

Observe no exemplo acima que o status de não suportado para a chave `vfs.fs.size[/nono]` é indicado pela propriedade `state` tendo o valor '1' e a mensagem de erro vai na propriedade `value`.

A mensagem de erro será limitada em 2048 caracteres no lado do servidor.

## Protocolo XML antigo

O Zabbix irá tratar até 16 MB de dados XML codificados em Base64, mas um único valor decodificado não poderá ser superior a 64 KB ou será truncado para 64 KB durante a decodificação.

## Veja também

1. [Mais detalhes sobre o protocolo do Zabbix Agent \(inglês\)](#)

From:

<https://www.zabbix.com/documentation/3.2/> - **Zabbix Documentation 3.2**

Permanent link:

<https://www.zabbix.com/documentation/3.2/pt/manual/appendix/items/activepassive>

Last update: **2016/02/16 15:25**

