

3 Verificações ativas e passivas de agentes

Visão geral

Esta seção apresenta detalhes sobre verificações passivas e ativas realizadas por agentes Zabbix.

O Zabbix usa um protocolo baseado em JSON para comunicação com o agente Zabbix.

Há algumas definições usadas nos detalhes dos protocolos usados pelo Zabbix:

```
<HEADER> - "ZBXD\x01" (5 bytes)
<DATALEN> - comprimento dos dados (8 bytes). 1 Será formatado como
01/00/00/00/00/00/00/00 (oito bytes em hexadecimal, número de 64 bits)
```

Para não exaurir (potencialmente) a memória, o Zabbix server está limitado a aceitar apenas 128MB em uma conexão usando o protocolo Zabbix.

Verificações passivas

Uma verificação passiva é uma simples requisição de dados. O servidor Zabbix server ou proxy pede algum dado (por exemplo, carga de CPU) and e o agente Zabbix manda o resultado de volta para o servidor.

Requisição do servidor

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

Resposta do agente

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

Acima, a parte entre colchetes é opcional e só será enviada para itens não suportados.

Por exemplo, para itens suportados:

1. Servidor abre uma conexão TCP
2. Servidor envia **agent.ping \ n**
3. Agente lê o pedido e responde com **<header> <DataLen> 1**
4. Servidor processa os dados valor, '1' nesse caso
5. Conexão TCP é fechada

Para itens não suportados:

1. Servidor abre uma conexão TCP

2. Servidor envia **vfs.fs.size [/nono]\n**
3. Agente lê o pedido e responde com **<header><DataLen>ZBX_NOTSUPPORTED\0Cannot obtain filesystem information: [2] No such file or directory**
4. Servidor processa os dados, muda o status do item para não suportado com a mensagem de erro especificada
5. Conexão TCP é fechada

Verificações ativas

Verificações ativas requerem um processamento mais complexo. O agente deve primeiro obter do servidor(es) uma lista de itens para processamento independente.

Os servidores dos quais obter verificações ativas estão listados no parâmetro 'ServerActive' [arquivo de configuração](#) do agente. A frequência da requisição destas verificações é definida pelo parâmetro 'RefreshActiveChecks' no mesmo arquivo de configuração. No entanto, se verificações ativas falharem, só serão retentadas após 60 segundos (valor fixo, hardcoded).

O agente, a partir de então, envia periodicamente os novos valores para o(s) servidor(es).

Obtendo a lista de itens

Requisição do agente

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

Resposta do servidor

```
<HEADER><DATALENn> {  
  "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** devem existir, independente do tipo de item ser log ou não.

Por exemplo:

1. Agente abre uma conexão TCP
2. Agente pede a lista de verificações
3. Servidor responde com uma lista de itens (key item, delay)
4. Agente analisa a resposta
5. Conexão TCP é fechada
6. Agent inicia coleta periódica de dados

Observe que dados de configuração (confidenciais) podem tornar-se disponíveis para quem tiver acesso à porta do Zabbix trapper quando se usa verificações ativas. Isso é possível porque qualquer pessoa pode fingir ser um agente ativo e requisitar dados de configuração de itens; não há autenticação.

Enviando dados coletados

Agente envia

```
<HEADERr><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: 141.708,521 Starting Zabbix Agent
[<hostname>]. Zabbix 2.4.0 (revision 50000).",
      "clock": 1400675595,
      "ns": 77053975
    },
  ],
}
```

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

Resposta do servidor

```
<code> <HEADERr> <DATALEN> {
```

```
  "response": "success",
  "info": "processed: 3; failed: 0; total: 3; seconds spent: 0.003534"
```

```
} </code>
```

Se o envio de alguns valores falhar no servidor (por exemplo, porque o host ou item foi desativado ou excluído), o agente não vai repetir o envio desses valores.

Por exemplo:

1. Agente abre uma conexão TCP
2. Agente envia uma lista de valores
3. Servidor processa os dados e envia o status de volta
4. Conexão TCP é fechada

Note como, no exemplo acima, o estado não suportado para `vfs.fs.size [/nono]` é indicado pelo valor de "state" de 1 e a mensagem de erro na propriedade "value".

A mensagem de erro será truncada em 2048 símbolos no lado do servidor.

Protocolo XML antigo

O Zabbix vai aceitar até 16 MB de dados XML codificados em Base64, mas um único valor decodificado não pode ter mais de 64 KB, pois caso contrário ele será truncado para 64 KB durante a decodificação.

From:
<https://www.zabbix.com/documentation/2.4/> - **Zabbix Documentation 2.4**

Permanent link:
<https://www.zabbix.com/documentation/2.4/pt/manual/appendix/items/activepassive>

Last update: **2014/12/02 15:33**



