

3 Пассивные и активные проверки агента

Обзор

Этот раздел описывает детали пассивных и активных проверок, которые выполняются [Zabbix агентом](#).

Zabbix использует протокол на основе JSON для взаимодействия с Zabbix агентом.

Для получения более подробных сведений об определении длины заголовка и данных, пожалуйста, обратитесь к [деталям протокола](#).

Пассивные проверки

Пассивной проверкой является простой запрос данных. Zabbix сервер или прокси запрашивает какие-либо данные (к примеру, Загрузку CPU) и Zabbix агент отправляет обратно результат серверу.

Запрос сервера

```
<ЗАГОЛОВОК><ДЛИНАДААННЫХ><ключ элемента данных>
```

Ответ агента

```
<ЗАГОЛОВОК><ДЛИНАДААННЫХ><ДААННЫЕ>[\0<ОШИБКА>]
```

Часть в квадратных скобках выше опциональна и отправляется только по неподдерживаемым элементам данных.

Например:

1. Сервер открывает TCP соединение
2. Сервер отправляет **<ЗАГОЛОВОК><ДЛИНАДААННЫХ>agent.ping\n**
3. Агент читает запрос и отвечает с **<ЗАГОЛОВОК><ДЛИНАДААННЫХ>1**
4. Сервер обрабатывает данные, чтобы извлечь значение, в нашем случае 1
5. TCP соединение закрывается

Для неподдерживаемых элементов данных:

1. Сервер открывает TCP соединение
2. Сервер отправляет **<ЗАГОЛОВОК><ДЛИНАДААННЫХ>vfs.fs.size[/nono]\n**
3. Agent читает запрос и отвечает с **<ЗАГОЛОВОК><ДЛИНАДААННЫХ>ZBX_NOTSUPPORTED\0Cannot obtain filesystem information: [2] No such file or directory**
4. Сервер обрабатывает данные, меняет состояние элемента данных на неподдерживаемое с указанным сообщением об ошибке

5. TCP соединение закрывается

Активные проверки

Активные проверки требуют более сложной обработки. Агент должен сначала получить с сервера список элементов данных для независимой обработки.

Сервера, которые отдают активные проверки, перечислены в параметре "ServerActive" [файла конфигурации](#) агента. Частота запросов проверок настраивается параметром 'RefreshActiveChecks' в том же файле конфигурации. Однако, если обновление активных проверок завершится с ошибкой, запрос повторится через, жестко задано в коде, 60 секунд.

Затем агент периодически отправляет новые значения на сервер(a).

Получение списка элементов данных

Запрос агента

```
<ЗАГОЛОВОК><ДЛИНАДААННЫХ>{  
  "request":"active checks",  
  "host": "<имя хоста>"  
}
```

Ответ сервера

```
<ЗАГОЛОВОК><ДЛИНАДААННЫХ>{  
  "response": "success",  
  "data": [  
    {  
      "key": "log[/home/zabbix/logs/zabbix_agentd.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  
    }  
  ]  
}
```

```
    }  
  ]  
}
```

Сервер должен ответить положительно. У каждого полученного элемента данных обязательно должны быть указаны параметры **key**, **delay**, **lastlogsize** и **mtime**, независимо от того элемент данных является журнальным или нет.

Например:

1. Агент открывает TCP соединение
2. Агент запрашивает список проверок
3. Сервер отвечает списком элементов данных (ключ элемента данных, интервал обновления)
4. Агент анализирует ответ
5. TCP соединение закрывается
6. Агент начинает периодический сбор данных

Возьмите на заметку, что (чувствительные) данные конфигурации могут стать доступными лицам, имеющим доступ к порту траппера Zabbix сервера, когда используются активные проверки. Это возможно так как любой может претендовать на возможность быть активным агентом и запросить данные конфигурации элементов данных; аутентификация не производится, если вы не используете опции [шифрования](#).

Отправка собранных данных

Отправка агента

```
<ЗАГОЛОВОК><ДЛИНАДААННЫХ>{  
  "request":"agent data",  
  "session": "12345678901234567890123456789012",  
  "data":[  
    {  
      "host":"<hostname>",  
      "key":"agent.version",  
      "value":"2.4.0",  
      "id": 1,  
      "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).",  
      "id": 2,  
      "clock":1400675595,  
      "ns":77053975  
    }  
  ]  
}
```

```
    },  
    {  
      "host": "<hostname>",  
      "key": "vfs.fs.size[/nono]",  
      "state": 1,  
      "value": "Cannot obtain filesystem information: [2] No such file  
or directory",  
      "id": 3,  
      "clock": 1400675595,  
      "ns": 78154128  
    }  
  ],  
  "clock": 1400675595,  
  "ns": 78211329  
}
```

Виртуальный ID назначается каждому значению. Значением ID является простой счётчик с инкрементом, уникальный в пределах одной сессии передачи данных (идентифицируется при помощи токена сессии передачи данных). Этот ID используется, чтобы отбрасывать дубликаты значений, которые могут быть отправлены в средах с плохой связью.

Ответ сервера

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

Если отправка некоторых значений завершилась неудачей на сервере (например, по причине того, что узел сети или элемент данных был деактивирован или удален), агент не будет повторять отправку этих элементов данных

Например:

1. Агент открывает TCP соединение
2. Агент отправляет список значений
3. Сервер обрабатывает данные и отправляет статус обратно
4. TCP соединение закрывается

Обратите внимание на то, как в примере выше указывается неподдерживаемое состояние для `vfs.fs.size[/nono]` при помощи значения `"state"` равного 1 и сообщения об ошибке в свойстве `"value"`.

Сообщение об ошибке будет обрезано до 2048 символов на стороне сервера.

Более старый XML протокол

Zabbix воспринимает до 16 MB XML данных закодированных в Base64, но одиночное декодированное значение должно быть не длиннее 64 KB, в противном случае значение будет

обрезано до 64 КБ в процессе декодирования.

From:

<https://www.zabbix.com/documentation/4.0/> - **Zabbix Documentation 4.0**

Permanent link:

<https://www.zabbix.com/documentation/4.0/ru/manual/appendix/items/activepassive>

Last update: **2018/08/06 03:11**

