



链滴

zabbix 监控端口自动发现功能

作者: [cuijianzhe](#)

原文链接: <https://ld246.com/article/1556177969287>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

现如今，无监控，不运维。能想到的尽量监控，拿到数据说话。这里的话，一套脚本拿下，直接输出json格式的数据，让系统自动发现。

1. 首先脚本功能要实现，怎么写出自动发现端口呢？

```
#!/bin/python
```

```
import os
import json
cmd=os.popen("""netstat -nltpl| grep -v -w -|grep -v rpc|awk -F "[:]+" '{if($4 ~ /0.0.0.0/ || $4 ~ /127.0.0.1/)print $5}""")
```

```
ports=[]
for port in cmd.readlines():
    r=port.strip()
    ports += [r]
```

```
print json.dumps({'data':ports},sort_keys=True,indent=4,separators=(',',':'))
```

此脚本可以简单的实现端口发现，其实就是用的命令，然后切割拿到自己想要的。可在linux中使用

```
netstat -nltpl| grep -v -w -|grep -v rpc|awk -F "[:]+" '{if($4 ~ /0.0.0.0/ || $4 ~ /127.0.0.1/)print $5}
```

```
[root@zabbix ~]# netstat -nltpl| grep -v -w -|grep -v rpc|awk -F "[:]+" '{if($4 ~ /0.0.0.0/ || $4 ~ /127.0.0.1/)print $5}'
```

```
3306
139
1617
5203
25
445
10050
514
10051
9600
```

但是呢，这只是可以看到起端口功能，如有需求可把端口对应名称采集出来。

可以如下：

```
[root@zabbix ~]# netstat -pntpl | awk '{print $4,$7}'|grep [0-9] |egrep -vw '%s'
```

```
0.0.0.0:3306 187405/mysqld
0.0.0.0:139 46349/smbd
0.0.0.0:1617 188444/rsyslogd
0.0.0.0:5203 2024/sshd
127.0.0.1:25 3189/master
0.0.0.0:445 46349/smbd
0.0.0.0:10050 21978/zabbix_agentd
0.0.0.0:514 188444/rsyslogd
0.0.0.0:10051 156956/zabbix_serve
:::139 46349/smbd
```

```
:::80 18117/httpd
192.168.51.202:9200 22644/java
:::1617 188444/rsyslogd
:::10514 195495/java
:::5203 2024/sshd
192.168.51.202:9300 22644/java
:::21 168385/vsftpd
:::3000 65015/grafana-serve
:::1:25 3189/master
:::8443 18117/httpd
:::445 46349/smbd
127.0.0.1:9600 195495/java
:::10050 21978/zabbix_agentd
:::514 188444/rsyslogd
:::10051 156956/zabbix_serve
```

实现命名采集后，用正则采集对应名称：

用shell正则去取值是这样的：

```
[root@zabbix ~]# netstat -pntl | awk '{print $4,$7}'|grep [0-9] | sed -e 's/127.0.0.1://g' -e 's/0.0.0.0://g' -e 's/:://g' -e 's/::://g' -e 's/1://g' -e 's/192.168.51.202://g' | sed 's/[0-9]*.\\//g'
```

```
3306 mysqld
139 smbd
1617 rsyslogd
5203 sshd
25 master
445 smbd
10050 zabbix_agentd
514 rsyslogd
10051 zabbix_server
139 smbd
80 httpd
9200 java
1617 rsyslogd
10514 java
5203 sshd
9300 java
21 vsftpd
3000 grafana-serve
25 master
8443 httpd
445 smbd
9600 java
10050 zabbix_agentd
514 rsyslogd
10051 zabbix_server
```

1.2 如下python不解释了。

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
# 使用python2 commands模块
```

```

import re
import commands
import json

DROP_LIST = ['22','25','111'] # 排除端口

def filterList():
    DROP_str = "|".join(DROP_LIST)
    CMD="sudo netstat -pntl | awk '{print $4,$7}'|grep [0-9] |egrep -vw '%s'" % (DROP_str)
    Result_Str = commands.getoutput(CMD)
    #print (Result_Str)
    tmp_list = Result_Str.split("\n") #每行加入列表
    new_dict = {}
    for line in tmp_list:
        # print (line)
        PORT_REG = re.search(r"(127.0.0.1|:::|0.0.0.0)(\d+).\d+/\(S+)",line)
        if PORT_REG is not None:
            match_line = (PORT_REG.groups())
            new_dict[ match_line[-1]] = match_line[-2]
    return new_dict

if __name__ == "__main__":
    Results = filterList()

    #格式化适合zabbix lld的json数据
    ports = []
    for key in Results:
        ports += [{'#PNAME}':key,{'#PPORT}':Results[key]]
    print json.dumps({'data':ports},sort_keys=True,indent=4,separators=(',',':'))

```

1.3 python3 端口自动发现功能:

```
#!/bin/env python3
```

```

import subprocess
import json
import re

def PortList():
    CMD = "sudo netstat -pntl | awk '{print $4,$7}'|grep [0-9] |egrep -vw '%s'"
    Result_str = subprocess.getoutput(CMD)
    #print(Result_str)
    tmp_list = Result_str.split("\n")
    #print(tmp_list)
    port_dict = {}
    for line in tmp_list:
        # print(line)
        PORT_REG = re.search(r"(127.0.0.1|:::|0.0.0.0)(\d+).\d+/\(S+)",line)
    #    print(PORT_REG)
        if PORT_REG is not None:
            match_line = (PORT_REG.groups())
            port_dict [ match_line[1]] = match_line[2]
    return port_dict

```

```

if __name__ == "__main__":
    Results = PortList()
    ports = []
    for key in Results:
        ports += [{'#PNAME}':key,{'#PPORT}':Results[key]]
    print(json.dumps({'data':ports},sort_keys=True,indent=4,separators=(',',':')))

```

以上脚本功能实现。然后在zabbix_agentd.d中添加key，以及调用脚本功能目录。

UserParameter=discovery.ports,/usr/bin/python /usr/lib/zabbix/externalscripts/zabbix_ports_discovery.py

最后在zabbix前端添加自动发现：

- 添加模板：

The screenshot shows the Zabbix template configuration interface. The template name is 'discovery port' and its visible name is 'A_discovery ports on server'. It is assigned to the 'Linux servers' group. The description field is empty. At the bottom, there are buttons for '更新' (Update), '克隆' (Clone), '全克隆' (Full Clone), '删除' (Delete), '删除并清除' (Delete and Clear), and '取消' (Cancel).

- 在相应模板添加自动发现规则

The screenshot shows the configuration page for an automatic discovery rule. The rule name is 'discovery ports on server' and its type is 'Zabbix 客户端'. The key value is 'discovery.ports' and the update interval is '30s'. There is a table for custom time intervals with one entry: '灵活' (Flexible) type, '调度' (Schedule) interval, '50s' interval, and '1-7,00:00-24:00' period. The resource cycle is '1d'. The rule is checked as '已启用' (Enabled). Buttons for '更新' (Update), '克隆' (Clone), '删除' (Delete), and '取消' (Cancel) are at the bottom.

- 添加监控项原型

监控项原型

所有模板 / A discovery ports on server 自动发现清单 / discovery ports on server 监控项原型 1 触发器类型 1 图形原型 主机模板

监控项原型 进程

* 名称

类型

* 键值

信息类型

单位

* 更新间隔

自定义时间间隔

● 添加触发器

触发器类型

所有模板 / A discovery ports on server 自动发现清单 / discovery ports on server 监控项原型 1 触发器类型 1 图形原型 主机模板

触发器类型 标记 依赖关系

* 名称

严重性

* 表达式

[表达式构造器](#)

事件成功条件

● 在相应主机添加模板,

主机

所有主机 / Server solo 已启用 ZBX SNMP JMX IPMI 应用集 13 监控项 54 触发器 18 图形 11 自动发现规则 3 Web 场景

主机 模板 IPMI 标记 宏 资产记录 加密

链接的模板	名称	动作
	<input type="text" value="A_discovery ports on server"/>	取消链接 取消链接并清理
	Template App Nginx	取消链接 取消链接并清理
	Template OS Linux	取消链接 取消链接并清理

链接指示器

● 查看主机发现数据:

discovery ports on server: discovery service zabbix_agentd on port:10050

discovery ports on server: discovery service sshd on port:5203

discovery ports on server: discovery service nginx: on port:80

discovery ports on server: discovery service mysqld on port:3306

discovery ports on server: discovery service java on port:8080