

Setting up SNMP Trapper for Zabbix.

Receiving SNMP traps is the opposite to querying SNMP-enabled devices. In this case the information is sent from a SNMP-enabled device and is collected or “trapped” by Zabbix. Usually traps are sent upon some condition change and the agent connects to the server on port 162 (as opposed to port 161 on the agent side that is used for queries). Using traps may detect some short problems that occur amidst the query interval and may be missed by the query data.

Receiving SNMP traps in Zabbix is designed to work with snmptrapd and one of the built-in mechanisms for passing the traps to Zabbix - either a perl script or SNMPPTT.

The workflow of receiving a trap:

- snmptrapd receives a trap
- snmptrapd passes the trap to SNMPPTT or calls Perl trap receiver
- SNMPPTT or Perl trap receiver parses, formats and writes the trap to a file
- Zabbix SNMP trapper reads and parses the trap file
- For each trap Zabbix finds all “SNMP trapper” items with host interfaces matching the received trap address. Note that only the selected “IP” or “DNS” in host interface is used during the matching.
- For each found item, the trap is compared to regexp in “snmptrap[regexp]”. The trap is set as the value of all matched items. If no matching item is found and there is an “snmptrap.fallback” item, the trap is set as the value of that. If the trap was not set as the value of any item, Zabbix by default logs the unmatched trap. (This is configured by “Log unmatched SNMP traps” in Administration -> General -> Other.)

Update firewall rules.

Setting up firewall 162 port should be opened. Add the following line in `/etc/sysconfig/iptables`:

```
-A INPUT -p udp -m udp --dport 162 -j ACCEPT
```

Restart Firewall.

```
[ahmed@nms ~]# service iptables restart
```

Setting up Zabbix to receive SNMP traps using `zabbix_trap_receiver.pl`.

Install additional packages

```
[ahmed@nms ~]# yum install -y net-snmp-utils net-snmp-perl
```

We will be using `zabbix_trap_receiver.pl`, File can be downloaded from [HERE](#).

Copy the file to `/usr/bin`

```
[ahmed@nms ~]# cp zabbix_trap_receiver.pl /usr/bin
[ahmed@nms ~]# chmod +x /usr/bin/zabbix_trap_receiver.pl
```

Update snmptrapd.conf

```
[ahmed@nms ~]# vi /etc/snmp/snmptrapd.conf
```

Append below lines to snmptrapd.conf

```
authCommunity execute public
perl do "/usr/bin/zabbix_trap_receiver.pl";
```

Enable Zabbix SNMP trapper in Zabbix server configuration.

```
[ahmed@nms ~]# vi /etc/zabbix/zabbix_server.conf
```

Enable SNMP trap in zabbix_server.conf

```
StartSNMPTrapper=1
```

SNMPTrapperFile should be same as what it is in zabbix_trap_receiver.pl file.

```
SNMPTrapperFile=/tmp/zabbix_traps.tmp
```

Restart Zabbix Server.

```
[ahmed@nms ~]# service zabbix-server restart
```

Setting snmptrapd to start on reboot.

Configure snmptrapd to start automatically:

```
[ahmed@nms ~]# chkconfig snmptrapd on
```

and restart snmptrapd service:

```
[ahmed@nms ~]# service snmptrapd restart
```

SNMP trap transmission file rotation (optional)

Create a directory to store the data.

```
[ahmed@nms ~]# mkdir -p /var/log/zabbix_traps_archive
[ahmed@nms ~]# chmod 777 /var/log/zabbix_traps_archive
```

Add below contents to /etc/logrotate.d/zabbix_traps.

```

/tmp/zabbix_traps.tmp {
    weekly
    size 10M
    compress
    compresscmd /usr/bin/bzip2
    compressoptions -9
    notifempty
    dateext
    dateformat -%Y%m%d
    missingok
    olddir /var/log/zabbix_traps_archive
    maxage 365
    rotate 10
}

```

Testing

Send test trap

```

[ahmed@nms ~]# snmptrap -v 1 -c public 127.0.0.1 '.1.3.6.1.6.3.1.1.5.4' '0.0.0.0' 6 33 '55' \
.1.3.6.1.6.3.1.1.5.4 s "eth0"

```

and check that trap received in the /tmp/zabbix_traps.tmp.

```

PDU INFO:
notificationtype      TRAP
version               0
receivedfrom         UDP: [127.0.0.1]:41840->[127.0.0.1]
errorstatus           0
messageid             0
community             public
transactionid         2
errorindex            0
requestid             0
VARBINDS:
DISMAN-EVENT-MIB::sysUpTimeInstance type=67 value=Timeticks: (55) 0:00:00.55
SNMPv2-MIB::snmpTrapOID.0          type=6  value=OID: IF-MIB::linkUp.0.33
IF-MIB::linkUp                      type=4  value=STRING: "eth0"
SNMP-COMMUNITY-MIB::snmpTrapCommunity.0 type=4  value=STRING: "public"
SNMPv2-MIB::snmpTrapEnterprise.0   type=6  value=OID: IF-MIB::linkUp

```

We are done with setting up SNMP trapper.

Create a Template called “Template SNMP trap fallback”

Creating Item called “SNMP trap fallback” in template “Template SNMP trap fallback”

- Name: SNMP trap fallback
- Type: SNMP trap
- Key: snmptrap.fallback
- Type of information: Log

This item will collect all unmatched traps. Create trigger which will inform administrator about new unmatched traps:

- Name: Unmatched SNMP trap received from {HOST.NAME}
- Expression: {Template SNMP trap fallback:snmptrap.fallback.nodata(300)}=0

Complete zabbix_trap_receiver.pl File.

You can find the latest file from the link below.

[ZABBIX TRAPPER FILES HERE](#)

```
#!/usr/bin/perl

#
# Zabbix
# Copyright (C) 2000-2011 Zabbix SIA
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
#

#####
#### ABOUT ZABBIX SNMP TRAP RECEIVER ####
#####

# This is an embedded perl SNMP trapper receiver designed for sending data to the server.
# The receiver will pass the received SNMP traps to Zabbix server or proxy running on the
# same machine. Please configure the server/proxy accordingly.
#
# Read more about using embedded perl with Net-SNMP:
# http://net-snmp.sourceforge.net/wiki/index.php/Tut:Extending_snmpd_using_perl

#####
#### ZABBIX SNMP TRAP RECEIVER CONFIGURATION ####
#####

### Option: SNMPTrapperFile
# Temporary file used for passing data to the server (or proxy). Must be the same
# as in the server (or proxy) configuration file.
#
# Mandatory: yes
# Default:
$SNMPTrapperFile = '/tmp/zabbix_traps.tmp';
```

```

### Option: DateTimeFormat
# The date time format in strftime() format. Please make sure to have a corresponding
# log time format for the SNMP trap items.
#
# Mandatory: yes
# Default:
$DateTimeFormat = '%H:%M:%S %Y/%m/%d';

#####
#### ZABBIX SNMP TRAP RECEIVER ####
#####

use Fcntl qw(O_WRONLY O_APPEND O_CREAT);
use POSIX qw(strftime);

sub zabbix_receiver
{
    my (%pdu_info) = %{$_[0]};
    my (@varbinds) = @{$_[1]};

    # open the output file
    unless (sysopen(OUTPUT_FILE, $SNMPTrapperFile, O_WRONLY|O_APPEND|O_CREAT, 0666))
    {
        print STDERR "Cannot open [$SNMPTrapperFile]: $!\n";
        return NETSNMPTRAPD_HANDLER_FAIL;
    }

    # get the host name
    my $hostname = $pdu_info{'receivedfrom'} || 'unknown';
    if ($hostname ne 'unknown') {
        $hostname =~ /\[(.*?)\].*/;          # format: "UDP: [127.0.0.1]:41070->[127.0.0.1]"
        $hostname = $1 || 'unknown';
    }

    # print trap header
    #     timestamp must be placed at the beginning of the first line (can be omitted)
    #     the first line must include the header "ZBXTRAP [IP/DNS address] "
    #         * IP/DNS address is the used to find the corresponding SNMP trap items
    #         * this header will be cut during processing (will not appear in the item value)
    printf OUTPUT_FILE "%s ZBXTRAP %s\n", strftime($DateTimeFormat, localtime), $hostname;

    # print the PDU info
    print OUTPUT_FILE "PDU INFO:\n";
    foreach my $key(keys(%pdu_info))
    {
        printf OUTPUT_FILE "  %-30s %s\n", $key, $pdu_info{$key};
    }

    # print the variable bindings:
    print OUTPUT_FILE "VARBINDS:\n";
    foreach my $x (@varbinds)
    {
        printf OUTPUT_FILE "  %-30s type=%-2d value=%s\n", $x->[0], $x->[2], $x->[1];
    }
}

```

```
    close (OUTPUT_FILE);

    return NETSNMPTRAPD_HANDLER_OK;
}

NetSNMP::TrapReceiver::register("all", \&zabbix_receiver) or
    die "failed to register Zabbix SNMP trap receiver\n";

print STDOUT "Loaded Zabbix SNMP trap receiver\n";
```