Overview
This article shows you several methods of sending a trap to your Nagios server to test SNMP Trap functionality.

Sometimes when troubleshooting an SNMP Trap issue, it can be very helpful to remove the actual device that could be causing problems and use the `snmptrap` command instead. This troubleshooting method will confirm if your Nagios server is correctly receiving SNMP Traps and is configured correctly.

You will be executing the command on the Nagios host itself, this is why you see `localhost` in the commands below.

Uptime
When you send a trap, it must of course conform to a set of standards. The options are explained in each section below however there is one option that is common and needs explained.

Every trap needs an uptime value. Uptime is how long the system has been running since boot. Sometimes this is the operating system, other devices might use the SNMP engine uptime. So what value should you type in the commands below? Oddly enough, simply supplying no value by using two single quotes "" will instruct the command to obtain the value from the operating system you are executing it on.

For those who dig deeper and look at the spooled trap before it's processed will want to understand what type of format it is. Here is an example:

```
DISMAN-EVENT-MIB::sysUpTimeInstance 36:2:40:51.67
```

This equates to 36 days, 2 hours, 40 minutes and 51.67 seconds.

The key point to this section is that you now know why the commands below have two single quotes "" for the uptime value.

SNMP v2 Trap
The command below takes the form of:

```
snmptrap -v <snmp_version> -c <community> <destination_host> <uptime> <OID_or_MIB> <object> <value_type> <value>
```

Using a MIB:

```
snmptrap -v 2c -c public localhost '' NET-SNMP-EXAMPLES-MIB::netSnmpExampleHeartbeatNotification netSnmpExampleHeartbeatRate i 123456
```

Shortening the MIB:

```
snmptrap -v 2c -c public localhost '' netSnmpExampleHeartbeatNotification netSnmpExampleHeartbeatRate i 123456
```

Using OID's instead of MIB:

```
snmptrap -v 2c -c public localhost '' 1.3.6.1.4.1.8072.2.3.0.1 1.3.6.1.4.1.8072.2.3.2.1 i 123456
```

The commands above required the following settings in `/etc/snmp/snmptrapd.conf`

```
disableAuthorization yes
traphandle default /usr/sbin/snmptthandler
```

SNMP v3 Trap
The command below takes the form of:

```
snmptrap -v <snmp_version> -e <engine_id> -u <security_username> -a <authentication_protocal> -A ... -X <privacy_protocol_pass_phrase> <destination_host> <uptime> <OID_or_MIB> <object> <value_type> <value>
```

Using a MIB:

```
snmptrap -v 3 -e 0x090807060504030201 -u the_user_name -a SHA -A the_SHA_string -x AES -X the_AES_string localhost '' NET-SNMP-EXAMPLES-MIB::netSnmpExampleHeartbeatNotification netSnmpExampleHeartbeatRate i 123456
```

Shortening the MIB:

```
snmptrap -v 3 -e 0x090807060504030201 -u the_user_name -a SHA -A the_SHA_string -x AES -X the_AES_string localhost '' netSnmpExampleHeartbeatNotification netSnmpExampleHeartbeatRate i 123456
```

Using OID's instead of MIB:

```
snmptrap -v 3 -e 0x090807060504030201 -u the_user_name -a SHA -A the_SHA_string -x AES -X the_AES_string localhost '' 1.3.6.1.4.1.8072.2.3.0.1 1.3.6.1.4.1.8072.2.3.2.1 i 123456
```

The commands above required the following settings in `/etc/snmp/snmptrapd.conf`

```
disableAuthorization yes
traphandle default /usr/sbin/snmptthandler
createUser -e 0x090807060504030201 the_user_name SHA the_SHA_string AES the_AES_string
authUser log,execute,net the_user_name
```
SNMP Trap Definition

The following trap definition can be placed in /etc/snmp/snmptt.conf which will allow the test traps sent above to be passed through to Nagios:

```
EVENT netSnmpExampleHeartbeatRate .1.3.6.1.4.1.8072.2.3.0.1 "netSnmpExampleHeartbeatRate" Normal
FORMAT SNMP netSnmpExampleHeartbeatRate
EXEC /usr/local/bin/snmptraphandling.py "$r" "SNMP Traps" "$s" "$g" "netSnmpExampleHeartbeatRate"
```

Trap vs Inform

From the net-snmp website:

TRAP

A TRAP is a SNMP message sent from one application to another (which is typically on a remote host). Their purpose is merely to notify the other application that something has happened.

INFORM

SNMPv2 PDUs fixed this by introducing the notion of an INFORM, which is nothing more than an acknowledged TRAP. IE, when the remote application receives the INFORM it sends back a "I got it" message. This is nice because then the person sending the traps can keep trying until the trap gets through.

All of the commands above can be changed from snmptrap to snmpinform which will allow you to send a test inform.

SNMP Trap Configuration

More detailed information on configuring your server to accept SNMP TRAP’s can be found in the following KB articles:

- Nagios XI - SNMP Trap Hardening
- Nagios XI - SNMP Trap v3 Configuration

Final Thoughts

For any support related questions please visit the Nagios Support Forums at:

http://support.nagios.com/forum/

Posted by: tlea - Fri, Apr 1, 2016 at 2:49 AM. This article has been viewed 18639 times.

Online URL: https://support.nagios.com/kb/article/snmp-trap-how-to-send-a-test-trap-493.html