csstel

AN-862102

SNMP Trap Collection & Monitoring

Document No.: AN-862102, Rev 1.00
Applicable Products: ComView NX(S/M/L/X)
Contact: support@csstel.com
Web: www.csstel.com

Introduction

ComView offers a suite of network-based applications. One of which is SNMP Trap Receiver app, developed based on Net-SNMP that consists of applications used to implement SNMP v1, SNMP v2c and SNMP v3 using both IPv4 and IPv6.

SNMP Trap Receiver lets users collect and monitor SNMP traps (v1/2c/3)/informs (v2c/3) from SNMP-capable network devices (i.e., SNMP agents). The app can filter incoming traps/informs and only reports alarms on those that meet user-definable alarm conditions.

SNMP Trap Receiver app can be an ideal solution for users who manage servers and network devices to get real-time alerts on conditions that matter the most to their operations, while having the ability to archive all system traps/informs for network diagnostics and troubleshooting.

This application note is intended to provide an overview of SNMP Trap Receiver app and how to configure it for collecting and monitoring SNMP traps/informs from your network.

This application note does not provide detailed description of how to use ComView, its connectivity and configuration, and other supporting information, as these are beyond the scope of this document. Refer to other resources for more details.

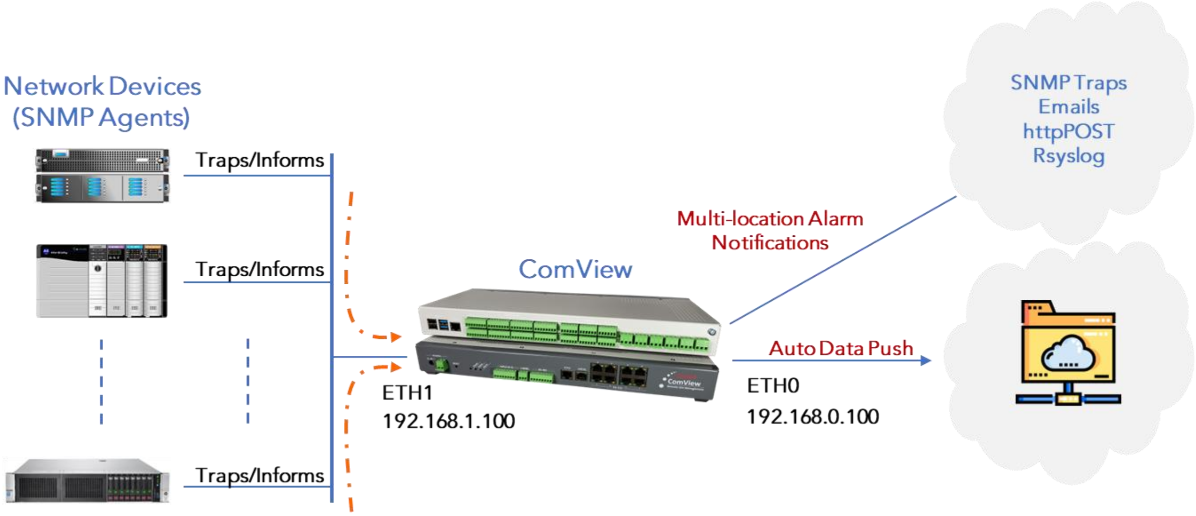
References:

[1]. ComView - User Guide

[2]. Net-SNMP - http://www.net-snmp.org/

Setup Overview

The diagram below illustrates the setup used in this application note.



From the above,

- Network devices are connected to ComView Ethernet 1 (ETH1, at IP address 192.168.1.100)
- Network devices must support SNMP and be configured to send SNMP traps/informs to ComView

In this application note, our goal is to configure SNMP Trap Receiver app to receive traps/informs from network devices and to detect signatures in these traps/informs for alarm conditions.

SNMP Trap Receiver App

SNMP Trap Receiver app is based on SNMP trap daemon (snmptrapd), a component of Net-SNMP. SNMP Trap Receiver uses '/usr/cvconf/snmptrapd.conf' for the snmptrapd configuration file. The following is the default file content that users may edit for their specific operational requirements:

```
# Authorizing SNMPv1/2c users with logging
# For security reason, change community name 'public' below
authCommunity log public

# SNMPv3 Trap vs Inform (source: http://www.net-snmp.org/)
# SNMPv3 with the User-Based Security Model (USM) makes use of an EngineID
# identifier for the SNMPv3 application that is authoritative
# (meaning the one who controls the flow of information)
# - SNMPv3 TRAPs: authoritative engine is the engine that sends the trap
# - SNMPv3 INFORMs: authoritative engine is the engine that receives the trap
#
# Configuring SNMPv3 TRAP User - add entry in the following format:
#  createUser -e ENGINEID myuser SHA "AuthPass" AES "KeyPass"
#    - ENGINEID: engine ID of the trap sender
#    - myuser: username that sends trap notification
#    - SHA: selected SHA from [SHA,MD5] options for authentication type
#    - AuthPass: authentication passphrase to generate authentication key
#    - AES: selected AES from [AES,DES] options for encryption type
#    - KeyPass: encryption passphrase to generate encryption key
#
#  For example:
#  createUser -e 0x8000744003dca632be3acf cvSNMPv3 SHA cvSNMPv3Auth AES cvSNMPv3Key
#
# To receive SNMPv3 traps from network devices, add 'createUser' line for each device.
# ENGINEID is the engine ID of the network device.
#
# For simplicity, the remaining parameters can be the same for all devices:
#    - myuser = cvSNMPv3
#    - AuthPass = cvSNMPv3Auth
#    - KeyPass = cvSNMPv3Key
#
createUser -e 0x8000744003dca632be3acf cvSNMPv3 SHA cvSNMPv3Auth AES cvSNMPv3Key

# Configuring SNMPv3 INFORM User - add entry in the following format:
#  createUser  myuser SHA "AuthPass" AES "KeyPass"
#
#  For example:
#  createUser cvSNMPv3 SHA cvSNMPv3Auth AES cvSNMPv3Key
#
# To receive SNMPv3 informs from network devices, add 'createUser' line for each device.
#
createUser cvSNMPv3 SHA cvSNMPv3Auth AES cvSNMPv3Key

# Authorizing SNMPv3 users - add entry for each user created in the following format:
#  authUser log,execute,net myuser
#   - log: logging
#   - execute: executing commands
#   - net: forwarding notification received
#   - myuser: username that sends notification
#  For example:
#  authUser log cvSNMPv3
authUser log cvSNMPv3

# To accept all traps
disableAuthorization yes

# Format SNMPv1 with header/trailer pair
format1 \n---Begin---\nAgent_Address = %A\nDate = %y-%02.2m-%02.2l %02.2h:%02.2j:%02.2k\n \
Security_Info = %P\nPDU_Attribute_Value_Pair_Array:\n%V\n%v\n---End---\n

# Format SNMPv2 and SNMPv3 with header/trailer pair
format2 \n---Begin---\nAgent_Address = %A\nDate = %y-%02.2m-%02.2l %02.2h:%02.2j:%02.2k\n \
Security_Info = %P\nPDU_Attribute_Value_Pair_Array:\n%V\n%v\n---End---\n
```

To describe the entries in the above file, we shorten it by removing comments and assign a line number to each entry. This results in the following:

```
Line 1: authCommunity log public

Line 2: createUser -e 0x8000744003dca632be3acf cvSNMPv3 SHA cvSNMPv3Auth AES cvSNMPv3Key

Line 3: createUser cvSNMPv3 SHA cvSNMPv3Auth AES cvSNMPv3Key

Line 4: authUser log cvSNMPv3

Line 5: disableAuthorization yes

Line 6: format1 \n---Begin---\nAgent_Address = %A\nDate = %y-%02.2m-%02.2l
%02.2h:%02.2j:%02.2k\n \
Security_Info = %P\nPDU_Attribute_Value_Pair_Array:\n%V\n%v\n---End---\n

Line 7: format2 \n---Begin---\nAgent_Address = %A\nDate = %y-%02.2m-%02.2l
%02.2h:%02.2j:%02.2k\n \
Security_Info = %P\nPDU_Attribute_Value_Pair_Array:\n%V\n%v\n---End---\n
```

Line 1:

This line supports SNMPv1/v2c traps and SNMPv2c informs with default community name 'public'. For security reason, edit the file and update this community name. All SNMP agents must use this community name for successful delivery of SNMPv1/v2c traps and SNMPv2c informs.

Line 2:

This line creates a USM (User-based Security Model) entry for an SNMPv3 trap sender/user; i.e., each network device that sends SNMPv3 trap to SNMP Trap Receiver. This line uses the following default values:

- Trap sender engine ID: 0x8000744003dca632be3acf
- Trap sender username: cvSNMPv3
- Authentication protocol: SHA
- Authentication passphrase: cvSNMPv3Auth
- Encryption key protocol: AES
- Encryption key passphrase: cvSNMPv3Key

If each SNMPv3 trap sender uses its own unique set of credentials, an associated USM entry must be created for it. For successful delivery of SNMPv3 trap, the sender credentials must match with those of USM entry pre-defined for that sender. To help simplify USM management for a large base of trap senders, it is advisable to use a common set of credentials for all senders.

Line 3:

This line creates a USM entry for an SNMPv3 inform sender/user; i.e., each network device that sends SNMPv3 inform to SNMP Trap Receiver. This line uses the following default values:

- Trap sender username: cvSNMPv3
- Authentication protocol: SHA

- Authentication passphrase: cvSNMPv3Auth
- Encryption key protocol: AES
- Encryption key passphrase: cvSNMPv3Key

If each SNMPv3 inform sender uses its own unique set of credentials, an associated USM entry must be created for it. For successful delivery of SNMPv3 inform, the sender credentials must match with those of USM entry pre-defined for that sender. To help simplify USM management for a large base of inform senders, it is advisable to use a common set of credentials for all senders.

Line 4:

This line defines what action(s) snmptrapd is authorized to take on receiving traps and informs from the specified SNMPv3 sender. In the default setting, snmptrapd is authorized to ‘log’ traps and informs from user ‘cvSNMPv3’.

Line 5:

This line disables all finer access control checks (defined by ‘authAccess’ and ‘setAccess’) and accepts all incoming traps and informs which are then processed on successful verification of sender credentials.

Line 6:

This line defines fields in formatting incoming SNMPv1 traps with Begin/End encapsulation

Line 7:

This line defines fields in formatting incoming SNMPv2c/v3 traps and informs with Begin/End encapsulation

Alarm Configuration

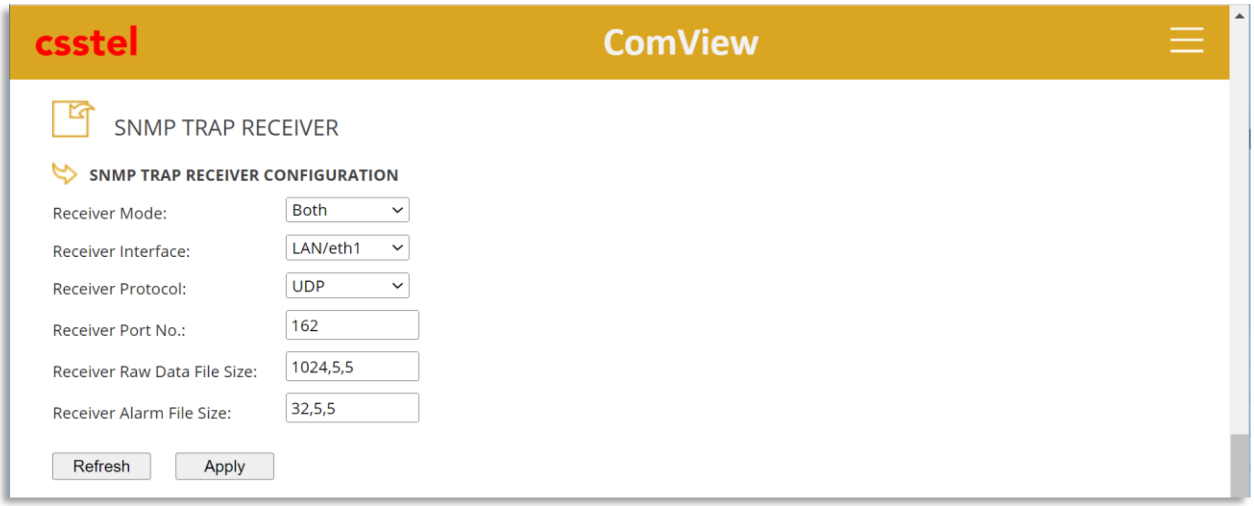
You can configure alarm conditions for SNMP Trap Receiver app using one of the following:

- Online web interface
- Online editing of text-based device configuration file
- Uploading device configuration file

In this application note, we use web interface to define sample alarm condition 0 and 1 to detect SNMP traps/informs that contain 'Trap' and 'Inform' signatures. Once detected, we want to be notified by ComView in real-time so corrective action can be taken.

To do this, you log on ComView via its web interface and navigate to 'CONFIGURATION -> NET APP'. For ease of use, an app is configured on one web page only.

SNMP Trap Receiver has two sections of configuration. The first section lets you define operating mode, port settings, raw data file size, and alarm file size, as shown in the screenshot below:



Receiver Mode: Both
Receiver Interface: LAN/eth1
Receiver Protocol: UDP
Receiver Port No.: 162
Receiver Raw Data File Size: 1024,5,5
Receiver Alarm File Size: 32,5,5

Receiver Mode:

Description: To define operating mode of SNMP Trap Receiver

Usage: Select from dropdown list [Raw, Alarm, Both, None]:
- Raw: Data collection mode, data received is logged in raw data file
- Alarm: Alarm monitoring mode, lines of data that meet alarm conditions are logged in common alarm file
- Both: Data collection and Alarm monitoring mode
- None: SNMP Trap Receiver not used

Receiver Interface:

Description: To define the Ethernet interface of SNMP Trap Receiver

Usage: Select from dropdown list [WAN/eth0, LAN/eth1]

Receiver Protocol:

Description: To define network protocol of SNMP Trap Receiver

Usage: Select from dropdown list [TCP/UDP]

Receiver Port No.:

Description: To define network port of SNMP Trap Receiver

Usage: Enter a valid network port number

Receiver Raw Data File Size:

Description: To define raw data file size in kilo bytes (kB), the percentage (%) to remove data in FIFO manner when file overflows its limit, and the time interval in minutes (min) to check the file size.

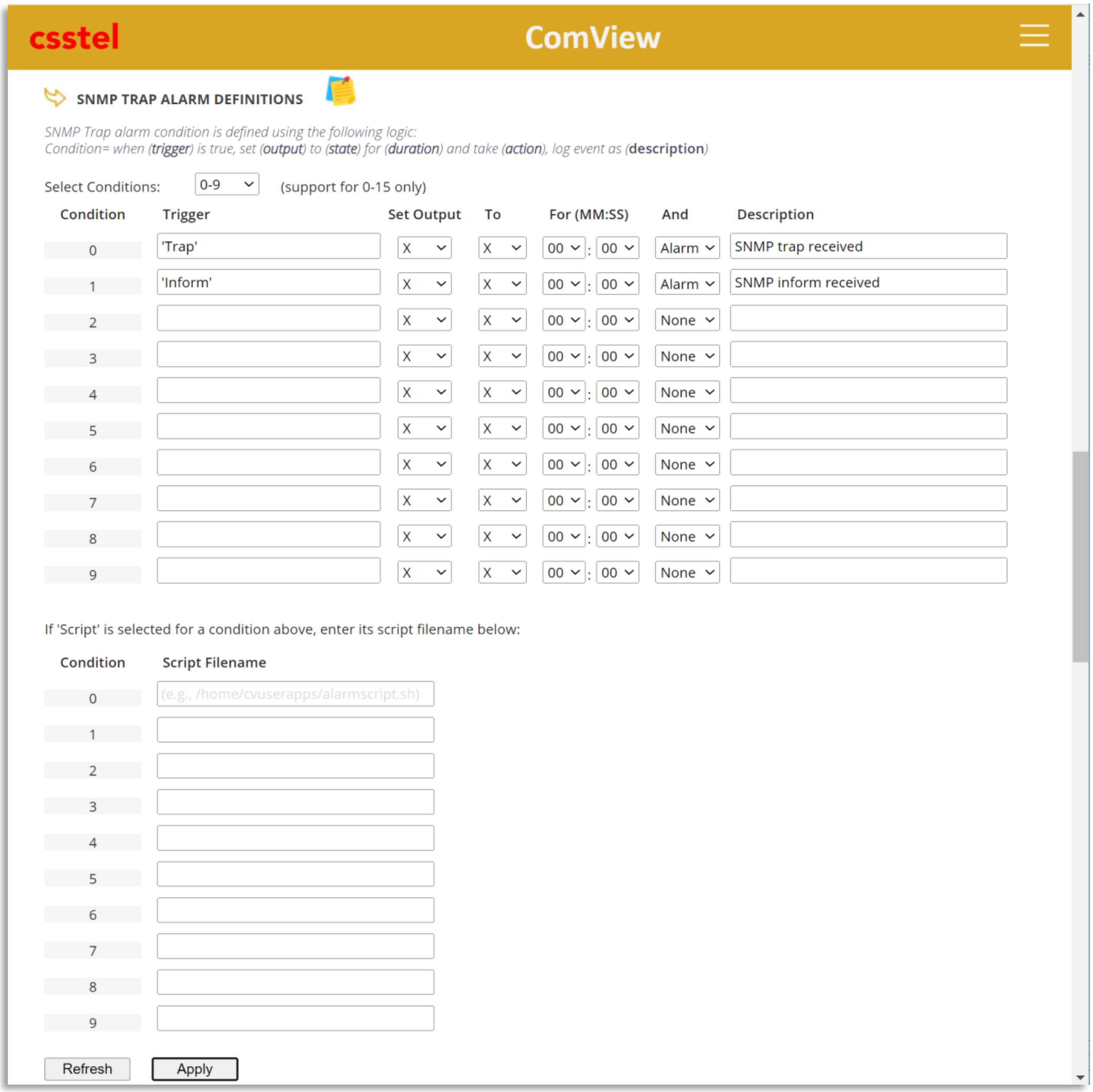
Usage: Enter values in ‘kB,%,min’ format

Receiver Alarm File Size:

Description: To define alarm file size in kilo bytes (kB), the percentage (%) to remove data in FIFO manner when file overflows its limit, and the time interval in minutes (min) to check the file size.

Usage: Enter values in ‘kB,%,min’ format

The second section of SNMP Trap Receiver configuration lets you define alarm conditions by setting values in the fields accordingly, as shown in the screenshot below:



SNMP TRAP ALARM DEFINITIONS

SNMP Trap alarm condition is defined using the following logic:
*Condition= when (**trigger**) is true, set (**output**) to (**state**) for (**duration**) and take (**action**), log event as (**description**)*

Select Conditions: 0-9 (support for 0-15 only)

Condition	Trigger	Set Output	To	For (MM:SS)	And	Description
0	'Trap'	X	X	00 : 00	Alarm	SNMP trap received
1	'Inform'	X	X	00 : 00	Alarm	SNMP inform received
2		X	X	00 : 00	None	
3		X	X	00 : 00	None	
4		X	X	00 : 00	None	
5		X	X	00 : 00	None	
6		X	X	00 : 00	None	
7		X	X	00 : 00	None	
8		X	X	00 : 00	None	
9		X	X	00 : 00	None	

If 'Script' is selected for a condition above, enter its script filename below:

Condition	Script Filename
0	(e.g., /home/cvuserapps/alarmscript.sh)
1	
2	
3	
4	
5	
6	
7	
8	
9	

SNMP Trap Receiver implements the following expression to help users define an alarm condition more clearly and easily:

> Condition= when (**trigger**) is true, set (**output**) to (**state**) for (**duration**) and take (**action**), log event as (**description**)

The expression above is then mapped to columnar format for user entries as shown in the screenshot. These columns have the following syntax and usage:

Trigger: regular expression in single quotes (we entered 'Trap' and 'Inform' strings for condition 0 and 1, respectively)

Set Output: Dropdown list [X,0..5] to select output number to set when an alarm is detected:

- X: for 'not used' (i.e., no output selection)
- 0..5: to use Output[0..5]

To: Dropdown list [X, On, Off] to select the state to set the selected output port (i.e., output relay) on detected alarm

- X (uppercase): for no change for current state
- On: to set output to On; i.e., output relay is energized
- Off: to set output to Off; i.e., output relay is deenergized

For (MM:SS): in MM:SS format for Min:Sec, the time duration to set output state:

- Valid time range is [00:00 - 59:59]
- Value 00:00 sets the output state without resetting it

And: Dropdown list [None, Alarm, Script] to select action to take on alarm condition:

- None: no action to take
- Alarm: send alarm notifications via methods enabled
- Script: full pathname to executable user-specific bash script file (e.g., '/home/cvuserapps/myscript.sh') to execute immediately on alarm condition

Description: user description of alarm condition (we entered 'SNMP trap received' and 'SNMP inform received'):

- Up to 30 characters (including space)
- Comma ',' not allowed (since comma is used as data field separator)

In the screenshot above, we completed setting up 2 alarm conditions with 'Trap' and 'Inform' as alarm signatures to monitor SNMP traps/informs for alarms. We also set action to 'Alarm' for alarm notifications. Note that for actual usage, change the alarm signatures used in the above to match with those of actual network devices.

Setup Verification

To verify our setup, we need to confirm that SNMP traps/informs sent to ComView are received and logged in its raw data file, alarms are logged in alarm files, and notifications are received on alarm.

To generate SNMP traps/informs, engage your network devices so that they generate your required traps/informs according to your alarm signatures. However, in this application note, we emulate that by using another ComView device to interactively send traps/informs using command lines.

snmptrap and snmpinform

We use snmptrap command to send traps and snmpinform to send informs. These commands are commonly available on any host that supports SNMP. In this example, we make up a simple trap/inform message with 3 OIDs (TestSite, trap/inform version, and Heartbeat) and use another ComView device to send these commands in shell environment to SNMP Trap Receiver at IP address 192.168.0.100.

--- SNMPv1 trap command

```
sudo snmptrap -v1 -c public 192.168.0.100 .1.3.6.1.4.1.29760.10 "" 6 19 "" \
.1.3.6.1.4.1.29760.10.19.1 s "TestSite" \
.1.3.6.1.4.1.29760.10.19.12 s "SNMPv1 Trap" \
.1.3.6.1.4.1.29760.10.19.13 s "System heartbeat"
```

--- SNMPv2c trap command

```
sudo snmptrap -v2c -c public 192.168.0.100 "" .1.3.6.1.4.1.29760.10.19 \
.1.3.6.1.4.1.29760.10.19.1 s "TestSite" \
.1.3.6.1.4.1.29760.10.19.12 s "SNMPv2c Trap" \
.1.3.6.1.4.1.29760.10.19.13 s "System heartbeat"
```

--- SNMPv2c inform command

```
sudo snmpinform -v2c -t 20 -c public 192.168.0.100 "" .1.3.6.1.4.1.29760.10.19 \
.1.3.6.1.4.1.29760.10.19.1 s "TestSite" \
.1.3.6.1.4.1.29760.10.19.12 s "SNMPv2c Inform" \
.1.3.6.1.4.1.29760.10.19.13 s "System heartbeat"
```

--- SNMPv3 trap noAuthNoPriv command

```
sudo snmptrap -v3 -e 0x8000744003dca632be3acf -l noAuthNoPriv -u cvSNMPv3 \
192.168.0.100 "" .1.3.6.1.4.1.29760.10.19 \
.1.3.6.1.4.1.29760.10.19.1 s "TestSite" \
.1.3.6.1.4.1.29760.10.19.12 s "SNMPv3 Trap NoAuthNoPriv" \
.1.3.6.1.4.1.29760.10.19.13 s "System heartbeat"
```

--- SNMPv3 trap authNoPriv command

```
sudo snmptrap -v3 -e 0x8000744003dca632be3acf -l authNoPriv -u cvSNMPv3 \
-a SHA -A cvSNMPv3Auth \
192.168.0.100 "" .1.3.6.1.4.1.29760.10.19 \
.1.3.6.1.4.1.29760.10.19.1 s "TestSite" \
.1.3.6.1.4.1.29760.10.19.12 s "SNMPv3 Trap AuthNoPriv" \
.1.3.6.1.4.1.29760.10.19.13 s "System heartbeat"
```

--- SNMPv3 trap authPriv command

```
sudo snmptrap -v3 -e 0x8000744003dca632be3acf -l authPriv -u cvSNMPv3 \
-a SHA -A cvSNMPv3Auth -x AES -X cvSNMPv3Key \
192.168.0.100 "" .1.3.6.1.4.1.29760.10.19 \
.1.3.6.1.4.1.29760.10.19.1 s "TestSite" \
.1.3.6.1.4.1.29760.10.19.12 s "SNMPv3 Trap AuthPriv" \
.1.3.6.1.4.1.29760.10.19.13 s "System heartbeat"
```

--- SNMPv3 Inform noAuthNoPriv

```
sudo snmpinform -v3 -t 20 -l noAuthNoPriv -u cvSNMPv3 \
192.168.0.100 "" .1.3.6.1.4.1.29760.10.19 \
.1.3.6.1.4.1.29760.10.19.1 s "TestSite" \
.1.3.6.1.4.1.29760.10.19.12 s "SNMPv3 Inform NoAuthNoPriv" \
.1.3.6.1.4.1.29760.10.19.13 s "System heartbeat"
```

--- SNMPv3 Inform AuthNoPriv

```
sudo snmpinform -v3 -t 20 -l authNoPriv -u cvSNMPv3 \
-a SHA -A cvSNMPv3Auth \
192.168.0.100 "" .1.3.6.1.4.1.29760.10.19 \
.1.3.6.1.4.1.29760.10.19.1 s "TestSite" \
.1.3.6.1.4.1.29760.10.19.12 s "SNMPv3 Inform AuthNoPriv" \
.1.3.6.1.4.1.29760.10.19.13 s "System heartbeat"
```

--- SNMPv3 Inform: authPriv

```
sudo snmpinform -v3 -t 20 -l authPriv -u cvSNMPv3 \
-a SHA -A cvSNMPv3Auth -x AES -X cvSNMPv3Key \
192.168.0.100 "" .1.3.6.1.4.1.29760.10.19 \
.1.3.6.1.4.1.29760.10.19.1 s "TestSite" \
.1.3.6.1.4.1.29760.10.19.12 s "SNMPv3 Inform AuthPriv" \
.1.3.6.1.4.1.29760.10.19.13 s "System heartbeat"
```

The above commands follow the syntax below:

```
snmptrap -v1 -c <community> <destination_host> <OID|MIB> "" <trap-type> <trap-id> "" \
[<oid> <type> <value>...]

snmptrap -v2c -c <community> <destination_host> <uptime> <OID|MIB> \
[<oid> <type> <value>...]

snmpinform -v2c -c <community> <destination_host> <uptime> <OID|MIB> \
[<oid> <type> <value>...]

snmptrap -v3 -e <engine_id> -l <noAuthNoPriv|authNoPriv|authPriv> -u <username> \
[-a <MD5|SHA>][-A <authphrase>][-x <DES|AES>][-X <privaphrase>] \
<ipaddress>[:<dest_port>] <uptime> <OID|MIB> \
[<oid> <type> <value>...]

snmpinform -v3 -l <noAuthNoPriv|authNoPriv|authPriv> -u <username> \
[-a <MD5|SHA>][-A <authphrase>][-x <DES|AES>][-X <privaphrase>] \
<ipaddress>[:<dest_port>] <uptime> <OID|MIB> \
[<oid> <type> <value>...]
```

NOTE: Inform commands use ‘-t’ option for timeout of 20 seconds between attempts

Raw Data File '/tmp/cvdata/cvSNMPTrapAppRaw.txt'

SNMP Trap Receiver logs incoming traps/informs in its raw data file. Each trap/inform is encapsulated by '---Begin---'/'---End---' header/trailer pair for legibility and ease in parsing, as defined in snmptrapd.conf configuration file. The following shows the file content with traps/informs received from the commands above:

```
---Begin---
Agent_Address = 192.168.0.130
Date = 2022-10-31 10:28:54
Security_Info = TRAP, SNMP v1, community public
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv1 Trap"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"
---End---

---Begin---
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:29:16
Security_Info = TRAP2, SNMP v2c, community public
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.2.1.1.3.0 = Timeticks: (30542175) 3 days, 12:50:21.75
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv2c Trap"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"
---End---

---Begin---
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:29:35
Security_Info = INFORM, SNMP v2c, community public
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.2.1.1.3.0 = Timeticks: (30544065) 3 days, 12:50:40.65
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv2c Inform"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"
---End---

---Begin---
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:30:02
Security_Info = TRAP2, SNMP v3, user cvSNMPv3, context
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.2.1.1.3.0 = Timeticks: (30546723) 3 days, 12:51:07.23
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv3 Trap NoAuthNoPriv"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"
---End---

---Begin---
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:30:17
Security_Info = TRAP2, SNMP v3, user cvSNMPv3, context
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.2.1.1.3.0 = Timeticks: (30548280) 3 days, 12:51:22.80
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv3 Trap AuthNoPriv"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"
```

```
---End---

---Begin---
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:30:37
Security_Info = TRAP2, SNMP v3, user cvSNMPv3, context
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.2.1.1.3.0 = Timeticks: (30550270) 3 days, 12:51:42.70
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv3 Trap AuthPriv"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"
---End---

---Begin---
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:31:02
Security_Info = INFORM, SNMP v3, user cvSNMPv3, context
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.2.1.1.3.0 = Timeticks: (30552737) 3 days, 12:52:07.37
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv3 Inform NoAuthNoPriv"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"
---End---

---Begin---
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:31:13
Security_Info = INFORM, SNMP v3, user cvSNMPv3, context
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.2.1.1.3.0 = Timeticks: (30553883) 3 days, 12:52:18.83
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv3 Inform AuthNoPriv"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"
---End---

---Begin---
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:31:29
Security_Info = INFORM, SNMP v3, user cvSNMPv3, context
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.2.1.1.3.0 = Timeticks: (30555458) 3 days, 12:52:34.58
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv3 Inform AuthPriv"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"
---End---
```

Alarm File '/tmp/cvdata/cvSNMPTrapAppAlarm.txt'

SNMP Trap Receiver logs alarms as records in its alarm file. Each record consists of timestamp, user-defined alarm description, and formatted SNMP trap/inform source, as follows:

```
20221031,102854,ethernet,SNMPTRAPD,'Trap',192.168.0.130,SNMP trap received
Agent_Address = 192.168.0.130
Date = 2022-10-31 10:28:54
Security_Info = TRAP, SNMP v1, community public
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv1 Trap"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"

20221031,102916,ethernet,SNMPTRAPD,'Trap',0.0.0.0,SNMP trap received
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:29:16
Security_Info = TRAP2, SNMP v2c, community public
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.2.1.1.3.0 = Timeticks: (30542175) 3 days, 12:50:21.75
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv2c Trap"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"

20221031,102935,ethernet,SNMPTRAPD,'Inform',0.0.0.0,SNMP inform received
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:29:35
Security_Info = INFORM, SNMP v2c, community public
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.2.1.1.3.0 = Timeticks: (30544065) 3 days, 12:50:40.65
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv2c Inform"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"

20221031,103002,ethernet,SNMPTRAPD,'Trap',0.0.0.0,SNMP trap received
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:30:02
Security_Info = TRAP2, SNMP v3, user cvSNMPv3, context
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.2.1.1.3.0 = Timeticks: (30546723) 3 days, 12:51:07.23
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv3 Trap NoAuthNoPriv"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"

20221031,103017,ethernet,SNMPTRAPD,'Trap',0.0.0.0,SNMP trap received
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:30:17
Security_Info = TRAP2, SNMP v3, user cvSNMPv3, context
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.2.1.1.3.0 = Timeticks: (30548280) 3 days, 12:51:22.80
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv3 Trap AuthNoPriv"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"

20221031,103037,ethernet,SNMPTRAPD,'Trap',0.0.0.0,SNMP trap received
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:30:37
Security_Info = TRAP2, SNMP v3, user cvSNMPv3, context
PDU_Attribute_Value_Pair_Array:
```

```
iso.3.6.1.2.1.1.3.0 = Timeticks: (30550270) 3 days, 12:51:42.70
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv3 Trap AuthPriv"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"

20221031,103102,ethernet,SNMPTRAPD,'Inform',0.0.0.0,SNMP inform received
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:31:02
Security_Info = INFORM, SNMP v3, user cvSNMPv3, context
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.2.1.1.3.0 = Timeticks: (30552737) 3 days, 12:52:07.37
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv3 Inform NoAuthNoPriv"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"

20221031,103114,ethernet,SNMPTRAPD,'Inform',0.0.0.0,SNMP inform received
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:31:13
Security_Info = INFORM, SNMP v3, user cvSNMPv3, context
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.2.1.1.3.0 = Timeticks: (30553883) 3 days, 12:52:18.83
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv3 Inform AuthNoPriv"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"

20221031,103129,ethernet,SNMPTRAPD,'Inform',0.0.0.0,SNMP inform received
Agent_Address = 0.0.0.0
Date = 2022-10-31 10:31:29
Security_Info = INFORM, SNMP v3, user cvSNMPv3, context
PDU_Attribute_Value_Pair_Array:
iso.3.6.1.2.1.1.3.0 = Timeticks: (30555458) 3 days, 12:52:34.58
iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.29760.10.19
iso.3.6.1.4.1.29760.10.19.1 = STRING: "TestSite"
iso.3.6.1.4.1.29760.10.19.12 = STRING: "SNMPv3 Inform AuthPriv"
iso.3.6.1.4.1.29760.10.19.13 = STRING: "System heartbeat"
```

Note that since we defined alarm conditions with ‘Trap’ and ‘Inform’ as triggers and each of the trap/informs sent contains either ‘Trap’ or ‘Inform’ string in the .12 OID, therefore all are detected as alarms.

System Alarm File '/tmp/cvdata/cvAlarms.txt'

ComView logs alarms from all alarm sources in one consolidated alarm file. For each alarm record it logs in this file, ComView also sends alarm notifications via delivery methods that users defined.

The following shows the alarm records from SNMP Trap Receiver logged in the consolidated alarm file:

```
20221031,102854,ethernet,SNMPTRAPD,'Trap',192.168.0.130,SNMP trap received
20221031,102916,ethernet,SNMPTRAPD,'Trap',0.0.0.0,SNMP trap received
20221031,102935,ethernet,SNMPTRAPD,'Inform',0.0.0.0,SNMP inform received
20221031,103002,ethernet,SNMPTRAPD,'Trap',0.0.0.0,SNMP trap received
20221031,103017,ethernet,SNMPTRAPD,'Trap',0.0.0.0,SNMP trap received
20221031,103037,ethernet,SNMPTRAPD,'Trap',0.0.0.0,SNMP trap received
20221031,103102,ethernet,SNMPTRAPD,'Inform',0.0.0.0,SNMP inform received
20221031,103114,ethernet,SNMPTRAPD,'Inform',0.0.0.0,SNMP inform received
20221031,103129,ethernet,SNMPTRAPD,'Inform',0.0.0.0,SNMP inform received
```

ComView device notifies users of alarm conditions as listed in the consolidated alarm file, while the SNMP Trap Receiver alarm file provides more details on the notified alarms.

Summary

This application note illustrates how ComView SNMP Trap Receiver app can be configured to detect alarm conditions in SNMP traps/informs sent by network devices. SNMP Trap Receiver app can also format incoming traps/informs from these network devices in an easy-to-understand message and log them in a file for network diagnostics and troubleshooting.

About CSSTEL

CSSTEL is a privately held developer and manufacturer of ComView hardware and software solutions for secure, remote infrastructure site management since 1997 with installations in over 30 countries around the world.

We offer ComView solutions that are scalable and customizable to monitor and manage virtually the entire spectrum of remote site infrastructure and site conditions.

We help telecom service providers, carriers, financial institutions, healthcare providers, government agencies, utilities, and other public and private sector organizations maintain constant visibility and control over their remote site infrastructure.

⚠ IMPORTANT:

- CSSTEL Inc. assumes no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that accompany it. In no event shall CSSTEL Inc. be liable for any loss of profit, or any other commercial damage caused or alleged to have been caused directly or indirectly.
- No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of CSSTEL Inc.
- Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. CSSTEL Inc. makes no claim to these trademarks.
- All rights reserved.

Revision History

Revision	Date	Description
1.00	2023-01-08	Initial release

*** End of document ***