NAME

trapex − An SNMP trap forwarder with filtering and translation capability

SYNOPSIS

trapex [**−h**] [**−c** <config_file>] [**−b** <bind_ip>] [**−p** <listen_port>] [**−d**] [**−v**]

DESCRIPTION

The **trapex** program an SNMP Trap proxy/forwarder. It can receive, filter, manipulate, log, and forward SNMP traps to zero or mulitple destinations. It can receive and process SNMPv1, SNMPv2c, or SNMPv3 traps.

Trapex is modeled after, and closely resembles a commercial product called *eHealth Trap Exploder*. The legacy *trapexploder* program does not support SNMP v3 traps, and may no longer be suppored. **Trapex** was written to provide a suitable replacement for *trapexploder*.

Presently, all v2c and v3 traps are converted to v1 before they are logged and/or forwarded. Support for sending other versions may be added in a future release.

ARGUMENTS

The command−line arguments supported by **trapex** are optional if the **trapex** configuration file is in it default location (*/etc/trapex.conf*), and the remaining configuration options are set there. Note that any options set via the command−line will override their corresponding values in the configuration file.

The options are as follows:

−h
Print a usage summary message and exit.

−c *<*/path/to/trapex.conf*>*
Specify/override the location of the trapex configuration file. If not set, the default is */etc/trapex.conf"

−b *<bind_ip>*
By default, **trapex** will listen on all active interfaces for incoming traps. This option allow for specifying a specific IP on witch to listen.

−p *<port>*
Specify the UDP port on which to listen for incoming traps. The default is port 162.

−d
Enable *debug* mode. This causes **trapex** to print very verbose information on the incoming traps as well as the trap log entries to STDOUT.

−v
Print the **trapex** version and exit.

TRAPEX CONFIGURATION FILE

The **trapex** configuration file (*trapex.conf*) is used to set the various runtime options as well as the filtering, forwarding, and logging directives.

Blank lines are allowed, and those that start with *"#"* are for comments.

There are 3 types of directives in the *trapex.conf* file:

Configuration directives −
These are lines in the file that have **trapex** configuration options and their values. In most cases these line will have a parameter/option name and its corresponding value. Some directives (like *debug*) do not have a value and are a boolean that is true when that entry is uncommented.

IP Set definitions −
These are lines the define name lists of IP addresses that can be used in the filter directive IP address fields (Source IP and Agent Address). Unlike the Configuration and Filter directives, IP Set definitions will span multiple lines and the list of IPs will be contained within curly−brackets ({}).

Filter directives –

These are the lines that define a filter for matching incoming traps and specifying an *action* for traps that match the filter. All of these directives start with the word "*filter*".

CONFIGURATION DIRECTIVES

General Options:

debug

Enable *debug* mode. This causes **trapex** to print very verbose information on the incoming traps as well as the trap log entries to STDOUT.

trapexHost *<hostname>*

Set/overide the hostname for this trapex instance. If not specified, **trapex** will attempt to determine the local hostname and use that. Currently, this data is only used for the CSV log data.

listenAddress *<bind_ip>*

Specify the IP address on which to bind to listen for incoming traps. When set to a specific IP, only traps coming in to the network interface that has that IP will be received and processed. If not specified here or on the command−line, the default is *0.0.0.0* (all IPs).

listenPort *<port>*

Specify the UDP port on which to listen for incoming traps. If not set here or on the command−line, the default is 162.

ignoreVersions *<SNMP Version>[,<SNMP Version>]*

Specify one or more SNMP versions to ignore. Any traps that have a version that matches any listed here will be ignored and dropped by trapex. Valid versions are: *v1*, *v2c*, and *v3* (or just *1*, *2*, *3* would suffice as well). Multiple entries are separated by a comma (no spaces).

Note: Specifying all 3 versions will cause trapex to complain and exit at startup because no traps would be processed at all in that case.

Log File Handling:

Note that these directive affect only the regular log files. They do not apply to the CSV−based logs.

logfileMaxSize *<value in MB>*

Specify the maximum size in MB the log files can grow before they are rotated.

logfileMaxBackups *<value>*

Specify how many backup (rotated) log files to keep. Older backups beyond this number will be removed. The backup log files are renamed with a date− time stamp.

compressRotatedLogs

When uncommented, this option causes backup log files to be compressed with gzip after they are rotated.

SNMP v3 Setting

These are options for receiving SNMP v3 traps. Note that **trapex** currently only supports the SNMP v3 *User−based Security Model* (USM).

v3msgFlags *<AuthPriv|AuthNoPriv|NoAuthNoPriv>*

This specifies the *SNMP v3 Message Flags*. Currently, **Trapex** supports only the *Auth* (Authentication) and *Priv* (Privacy) flags. These are set via a single string as follows:

- **AuthPriv –** Authentication and privacy
- **AuthNoPriv –** Authentication and no privacy

- **NoAuthNoPriv** – No authentication, and no privacy

v3User *<username>*
Set the SNMP v3 username. This is required for v3.

v3authProtocol *<MD5|SHA>*
Set the SNMP v3 *authentication protocol*. Valid values are *MD5* or *SHA* (default). Note that this parameter is required if the Auth *Msg Flag* is set (v3msgFlags = *AuthNoPriv* or *AuthPriv*).

v3authPassword *<password>*
Set the SNMP v3 authentication password. This is required if Auth mode is set.

v3privProtocol *<AES|DES>*
Set the SNMP v3 *authentication protocol*. Valid values are *AES* (default) or *DES*. Note that this parameter is required if Priv mode *Msg Flag* is set (v3msgFlags = *AuthPriv*).

v3authPassword *<password>*
Set the SNMP v3 privacy password. This is required if Priv mode is set.

IP SETS

An IP Set is a named list of IP addresses that can be referenced in the filter entries for the Source IP or Agent IP fields. The format is:

```
ipset <ipset_name> {
   10.1.3.4
   10.1.3.5
   100.3.66.4
}
```

You can also put multiple (whitespace−separated) IPs on a single line

```
ipset <ipset_name2> {
   10.1.3.4 10.1.3.5 100.3.66.4
   192.168.3.4 192.168.3.5 200.4.99.1 200.4.99.26
   10.222.121.7
}
```

In the filter lines, you can then use "*ipset:<ipset_name>*" in either or both the *Source IP* or *Agent Address* fields.

FILTER DIRECTIVES

The **trapex** configuration *filter* directives are used for specifying which traps are processed and what action is taken for traps that match the filter.

Each *filter* line starts with the word "*filter*" followed by the *filter expressions*, the *action* for that filter, and for some actions, an option argument for that action.

Filter Expressions

The *filter expression* is a space separated set of 6 filter criteria for trap data fields in the following order:

SNMP Version
The SNMP version. Only incoming traps that match this version are processed by this filter. Valid values are *v1*, *v2c*, or *v3*.

Source IP
The source IP of the incoming trap packet. This can be a string match for a single IP address, a subnet in CIDR notation, or a regular expression.

Agent Address

The SNMNP v1 AgentAddr IP address. This can be a string match for a single IP address, a subnet in CIDR notation, or a regular expression.

Generic Type
The trap *Generic Type* (integer: 0−6).

Specific Type
The trap *Specific Type* (integer: 0−n).

Enterprise OID
The trap *Enterprise OID* value. This uses a regular expression for matching.

An asterisk (*****) can be used as a wildcard to indicate that any value for that field matches. For instance, a filter that would match all traps and forward them to 192.168.1.1 port 162 would look like this:

filter * * * * * * forward 192.168.1.1:162

If multiple fields are set to a non−wildcard value, then all of them have to match (logical AND) in order for the trap to match and trigger the action.

Filter Actions

The *actions* that are currenly supported by **trapex** are:

forward <ip_address:port> [break]
Forward the trap to the specified IP address and port. **WARNING:** Do not specify the trapex host and port as a destination or you will create a trap forwarding loop! Note that this action also supports an optional second argument: *break*. This tells trapex to stop processing this trap after the forward operation.

nat <ip_address|$SRC_IP>
Set the trap *AgentAddress* value to the specified IP address or use "*$SRC_IP*" to set it to the source IP of the trap packet.

log </path/to/log/file> [break]
Save the trap data to the specified log file. Any files created by log actions are subject to the log file handling configuration directives. Note that this action also supports an optional second argument: *break*. This tells trapex to stop processing this trap after the log operation.

csv </path/to/csv/file> [break]
Save the trap data to the specified file in a CSV format that is meant specifically for feeding directly to a Clickhouse database. This feature is specific to the SungardAS snmp_trap table in Sungard's internal Clickhouse implementation.

break
The *break* action means ignore this trap from this point forward − do not forward it or take any other actions − halt further filter processing and drop it.

Filter Processing

The order of the filter directives in the configuration file is important.

The filters are processed in the order they appear in the configuration file. When a trap is received, it is checked against each filter in order. If it matches a filter, the trap data is processed by the *action* for that filter, and that trap is checked against the next filter, and so on (unless the action is *break* − where the trap is dropped and ignored from that point on).

AUTHORS

Trapex was written by Damien Stuart <damien.stuart@sungardas.com>.

ACKNOWLEDGEMENTS

Trapex is written in Go and uses open−source *Go packages* for some of its core functionality. Special thanks goes to the the following for their work on these packages:

- Sonia Hamilton <sonia@snowfrog.net> for the *gosnmp* package which provides the SNMP packet receiving, parsing, and trap sending functionality.
- Nate Finch <nate.finch@gmail.com> for the *lumberjack* go package which is used for the log file handling.

BUGS

Bound to be some...

Send questions or bug reports to damien.stuart@sungardas.com Suggestions and/or comments are always welcome as well.