

Utimaco Safeware

www.utimaco.com

The Data Security Company

SafeGuard[®] MailGateway

SNMP Support

Gateway Version 5.70

utimaco[®]
s a f e w a r e
a member of the Sophos Group

Print

Copyright 2009

Utimaco Safeware AG
A member of the Sophos Group
Germanusstr. 4
52080 Aachen

Phone

+49 (0)241 / 1696-200

Fax

+49 (0)241 / 1696-199

Internet

www.utimaco.de

e-mail

info.sh@aachen.utimaco.de

Document Version

1.0

Date

28.01.09

Status

Final

Author

Marcel Strunk

All rights reserved

No part of this documentation may be reproduced in any form (printing, photocopy or according to any other process) without the written approval of Utimaco AG or be processed, reproduced or distributed using electronic systems.

The company Utimaco AG reserves the right to modify or amend the documentation at any time without prior notice. Utimaco AG assumes no liability for typographical errors and damages incurred due to them.

Table of Contents

1	Introduction.....	5
1.1	About this manual	5
2	What Is SNMP?	6
2.1	Functionality	7
3	Monitoring The SafeGuard MailGateway	8
4	SNMP And The SafeGuard MailGateway	9
4.1	Requesting SNMP data packets	9
4.1.1	Modifying "rc.config".....	9
4.1.2	Modifying "snmpd.conf".....	10
4.2	Sending alarm messages using SNMP.....	11
4.2.1	Creating "spm_SNMP.sh".....	11
4.2.2	Modifying "Logdrc"	12
4.2.3	Example script used to send SNMP traps.....	13
5	What Else Can I Do With SNMP?.....	17
5.1	Monitoring processes	17
5.2	Monitoring the load on SafeGuard MailGateway	18
5.3	Adding scripts/program calls in SNMP	20

1 Introduction

Thank you for purchasing our SecurE-Mail Gateway security system. We hope that you are satisfied with our product. However, please do not hesitate to contact us if you have any complaints or other comments.



You should read this SNMP Support manual carefully before you start working with SafeGuard MailGateway (also referred to below as SGMG).

1.1 About this manual

Utimaco Safeware AG retains the right to change or add to this manual without prior notice, at any time. Utimaco Safeware AG accepts no liability for print errors and any damage that might occur as a result of them.

To help you find notes in the text quickly and easily in this manual, we have used icons to highlight the most important information.



These contain important safety information that you must follow.



These are additional notes or supplementary information.



This is an example.

A separate manual, "Subject line control", has been created for e-mail users who want to use SafeGuard MailGateway to encrypt their e-mails.

The "Manual for System Administrators" is designed to help you manage SafeGuard MailGateway.

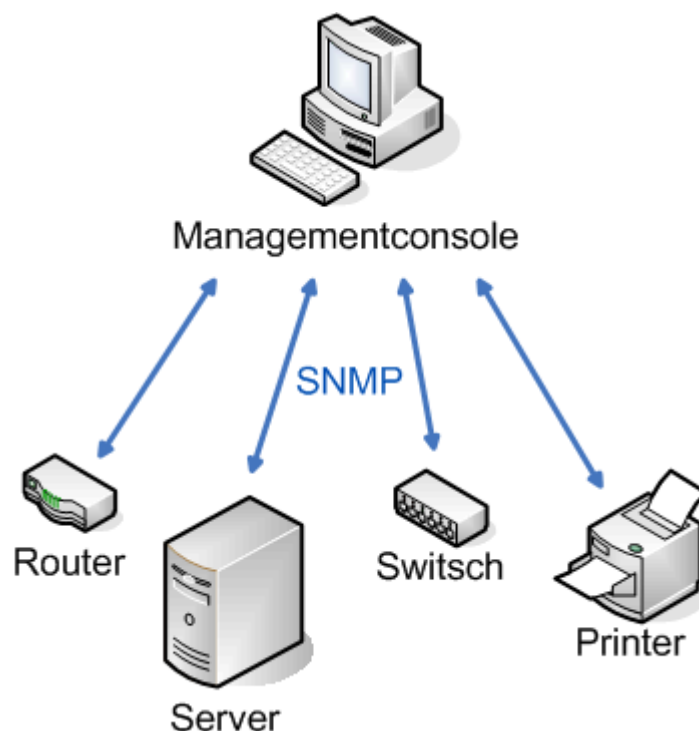
This manual is primarily intended to be used by SafeGuard MailGateway administrators who want to use the Simple Network Management Protocol (SNMP).

2 What Is SNMP?

The Simple Network Management Protocol (abbreviated to SNMP) is a network protocol developed by IETF for monitoring and controlling network devices (e.g. routers, servers, switches, printers, computers etc.) from one central station. To enable this, the protocol regulates communications between the devices and the monitoring station. SNMP describes the structure of the data packets that are sent and the communications processes. SNMP has been structured in such a way as to allow any device with network capability to be monitored. The network management tasks SNMP can perform include:

- monitoring network components.
- remote control and configuration of network components.
- error recognition and resolution.

Due to its very simplicity, SNMP has become the standard supported by the majority of management programs.



2.1 Functionality

Agents are used to perform the monitoring tasks. These agents are programs that run directly on the devices being monitored. These programs are not only able to monitor the status of these devices but also make settings and trigger actions on their own. SNMP enables the central management station to communicate with these agents over a network. To allow this, it provides six different data packets that can be sent:

- GET, to request a management data record
- GETNEXT, to call the next data record (to run through tables)
- GETBULK, to call several data records at the same time, for example, several rows in a table (available from SNMPv2)
- SET, to change a data record for a network element.
- RESPONSE, to reply to one of the previous packets.
- TRAP, an unrequested message sent by an agent to the manager to say that an event has occurred.

The three GET packets (Get, GetNext, GetBulk) can be sent by the manager to a particular agent to request data about a specific station. The agent then replies by sending a response packet which contains either the requested data or an error message.

The manager can then use the Set packet to change the values for the agent. This means it can then make settings or trigger actions. The agent then also sends a response packet to confirm it has received the values.

If the agent identifies an error when monitoring the system, it can send a Trap packet to report it to the Management station without being specifically requested to do so. The manager does not confirm these packets. As a result, the agent cannot be sure that the Trap actually reached the manager.

To ensure that the network load remains as low as possible, the connection-free UDP protocol is used to send messages. In this situation, the agent receives the queries (requests) on port 161, whereas port 162 is defined as the port on which the manager receives trap messages.

3 Monitoring The SafeGuard MailGateway

The log contains normal log messages and messages that the Administrator should respond to immediately because an error has occurred on the SGMG, which must be resolved as quickly as possible. The system classifies this type of message as an alarm message. A corresponding value is set in the flags.

Alarm messages are written to the log in the same way as normal log messages so that you have a summary of all the messages in the same place. In addition, a program is run for each alarm message to give the Administrator the most current information. This is usually the script

```
/usr/bin/Gateway/spm_mail.sh
```

This script creates an e-mail from the upper data fields and sends this to the e-mail address you entered in Web management.

In the following example, you see an alarm message that was generated by an SGMG and sent as an e-mail:

```
From: root@SGMGtest.utimaco.de
```

```
To: SGMGadmin@utimaco.de
```

```
Subject: [Alarm from SGMGtest.utimaco.de]
```

```
Alarm Message from gateway "SGMGtest.utimaco.de"
```

```
Flags: 0003
```

```
Gateway: 00000000
```

```
Date: 05.07.2004
```

```
Time: 09:53:01.65
```

```
Application: System
```

```
Counter: 0
```

```
Event ID: EF03
```

```
Priority: Warning
```

```
Group: Syslog
```

```
Process ID: 545
```

```
Text: init.d/gateway: Shutting down gateway operation!
```

The e-mail includes all the most important fields from the original log message. Priority and Group are classifications that are calculated from the Event ID.

4 SNMP And The SafeGuard MailGateway

From version 5.10.1 onwards, SGMG supports both the querying of data packets by a management station (Get, GetNext and GetBulk) and also the sending of status messages (Trap). However, the default setting is for these services to be switched off.

Before you can query data packets you must first configure and activate the SNMP service (snmpd). Section 4.1 describes how you do this.

To send alarm messages, you can use the sample script that is documented in section 4.2. Section 4.2.3 describes which files you need to change, on the SGMG, so that you can use SNMP TRAP to send status messages.

4.1 Requesting SNMP data packets

To request data packets from the SGMG you must first start the SNMPD service. This service starts automatically if you entered `SNMPD_LISTEN_SOCKET` in the "`rc.config`" file. You must also modify the "`snmpd.conf`" file. This file is used to define the SNMPD configuration.

4.1.1 Modifying "rc.config"

Although the "`rc.config`" file is present in the "`/etc`" file, you cannot modify it directly because it is overwritten every time the SGMG is configured. For this reason you should modify the "`rc.config`" file in the "`/gateway/chroot/webmgnt/conf/curr`" directory and, for `SNMPD_LISTEN_SOCKET`, input 161 as the value (port 161 is the default port used by SNMP to query the data packets). All other entries in this file remain unchanged.

```
#
# SNMP support
#
# Listen socket: e.g. "161" for UDP port 161 on all interfaces
# see 'man snmpd', LISTENING ADDRESSES section, for more
information
# don't start snmpd if empty
SNMPD_LISTEN_SOCKET="161"
```

4.1.2 Modifying "snmpd.conf"

The "snmpd.conf" file is stored in the "/etc/snmp" directory and can also be processed here. The SNMPD service has been preconfigured in such a way that it only has read access to SNMP data. Write access is only possible via the "localhost" (127.0.0.1) which is only present locally on the SGMG.

In this file you must modify at least the following entries:

- IP address area that is permitted to query the data packets;
- The "Community" name.

In addition, we recommend you also modify the "syslocation" and "syscontact" entries so that you can easily see who is responsible for the SGMG and where the SGMG is actually located (physically).

In the example below, every computer in the address range 192.168.0.0 through 192.168.0.255 can query data packets using the SGMG:

```
#      sec.name  source          community
com2sec local    localhost       public
com2sec mynetwork 192.168.0.0/24 public
```

... other entries ...

```
"syslocation" Utimaco IT department in Oberursel
"syscontact"  admin <admin@utimaco.de>
```

After you have modified these two files, click "Configure" in SGMG Web management. This ensures that the modified "rc.config" file is copied to the "/etc" directory.

Then use this command to start the SNMPD service on the SGMG:

```
"/etc/init.d/utimaco-snmp start".
```

Use the "'snmpwalk -v 1 localhost -c public system" command to check that the SNMPD service has been configured correctly. When you call this command, you must change the "community" name to match the settings.

4.2 Sending alarm messages using SNMP

As described in Chapter 3, the SGMG uses the "spm_Mail.sh" script in the "/usr/bin/Gateway" directory to send alarm messages. These alarm messages are usually sent by e-mail.

You can modify this script so that alarm messages will be sent by SNMP in future. However, we recommend you create a new script which takes on the functions of the original script. This has the advantage that the script changes are not overwritten when SGMG is updated.

4.2.1 Creating "spm_SNMP.sh"

The "spm_mail.sh" script is used as the basis of the "spm_snmp.sh". After this, the commands used in the script to send alarm messages by e-mail are replaced by an `SNMPTRAP` command. Section 4.2.3 contains an example of the changed script. You can, of course, change the script so that alarm messages can be sent both by SMNP and by e-mail.

Switch SGMG to "Blocked" mode. To do this, enter the command "init 2" in the command line.

You can now modify the existing script ("spm_mail.sh") or create a new one.

Switch the SGMG back to "Gateway" mode. To do this, enter the command "init 3" in the command line.

If an existing script has been modified, alarm messages will now be sent by SNMP. If you created a new script, you must then still modify the "Logdrc" file. You will find more information about this on the next page.

4.2.2 Modifying "Logdrc"

The "Logdrc" file is stored in the `"/etc/Gateway"` directory. However, the same restrictions apply here as they did for the `"rc.config"` file. It is always overwritten when the SGMG is reconfigured. For this reason, modify the file in the `"/gateway/chroot/webmgnt/conf/curr"` directory and, in `spm_exec_path`, enter the name of the script that was created to send alarm messages. All other entries in this file remain unchanged.

```
delay_reread          60
bastion_id            0

logevents             d601 d602 d609 d60a f608 e201 e202 e204 \
                    f201 f202 f203 fe02 d000 e000 f000 f010 \
                    dd01 dd02 ed03 fd04 fd05 df02

spmevents             fd06 fd07 ef03 ef08 ff04 ff05 ff06 ff07

ignoreevents         cd00 cf00 df01 c000 c001 c002 c003 c004 \
                    c005 c006 c007 c008 c009 c00a c00b c00c \
                    c00d c00e c00f

ignoreevents

spm_mode              exec
spm_exec_path         /usr/bin/Gateway/spm_snmp.sh
```

After you have modified the file, click "Configure" in SGMG Web Management. This ensures that the changed "Logdrc" is copied to the `"/etc/Gateway"` directory.

4.2.3 Example script used to send SNMP traps

```
#!/bin/bash
#####
#####

###
###

### File:
###

### /usr/bin/Gateway/spm_snmp.sh
###

###
###

### Copyright (c) 2006 Utimaco Safeware AG, Aachen, Germany
###

### All rights reserved
###
###
###

### Author:
##

### Marcel Strunk (MST)
###

###
###

### Description:
###

### This is a modified version of the spm_mail.sh bash script.
It is used ###

### for sending Spontaneous Messages as snmptraps.
###
###
###

### Comment:
###

### This script will be started by the Log Daemon for each SPM,
if the ###

### option spm_exec_path is set to the full path of this script.
Some ###

### information is read from /etc/rc.config.
###

###
###
```

```
### Synopsis:
###
###   spm_snmp.sh <flags> <gateway id> <date DDMMYYYY> <time
HHMMSShh>      ###
###
###   <application> <counter> <event id> <process id>
<text>      ###
###
###
### Changelog:
###
### 18.12.2006  MST  initial version
###
###
###
#####
#####

SNMPTRAP="192.168.0.119"      # IP-Address of SNMP Management
Center
. /etc/rc.config
case "$5" in
C000)  APPL="Log Daemon";;
C001)  APPL="Authentication Daemon";;
C002)  APPL="Interconnection Daemon";;
C003)  APPL="Configuration Daemon";;
C004)  APPL="Security Management Daemon";;
C005)  APPL="Control Daemon";;
C006)  APPL="Management Daemon";;
C007)  APPL="Alarm Daemon";;
C008)  APPL="SSL Proxy";;
C009)  APPL="LED configuration";;
C00A)  APPL="Java Authentication";;
C00B)  APPL="Certificate Daemon";;
C00C)  APPL="OCSP Daemon";;

C100)  APPL="Telnet Proxy";;
C101)  APPL="FTP Proxy";;
C102)  APPL="HTTP Proxy";;
C103)  APPL="SMTP Proxy";;
C104)  APPL="SMTP Daemon";;
```

```
C105)  APPL="NNTP Proxy";;
C106)  APPL="Permit Proxy";;
C107)  APPL="S9750 Proxy";;
C108)  APPL="ESMTP Proxy";;
C109)  APPL="POP3 Proxy";;
C10A)  APPL="SOCKS Wrapper";;
C10B)  APPL="RTSP Proxy";;
C10C)  APPL="Net8 Proxy";;
C10D)  APPL="Ping Proxy";;

C200)  APPL="UDP Relay";;
C201)  APPL="TCP Relay";;
C202)  APPL="Permit Relay";;

D000)  APPL="System";;
D100)  APPL="Kernel";;

*)     APPL="Application with ID 0x$5";;
esac

case $((0x$7 & 0x3000)) in

0)     PRIO="Debug";;           #      0 = 0x0000
4096)  PRIO="Information";;    # 4096 = 0x1000
8192)  PRIO="Warning";;       # 8192 = 0x2000
12288) PRIO="Error";;         # 12288 = 0x3000
*)     PRIO="Unknown !?";;
esac

case $((0x$7 & 0x0F00)) in

0)     GROUP="Application";;  #      0 = 0x0000
512)   GROUP="System";;      #     512 = 0x0200
1024)  GROUP="Security";;    #    1024 = 0x0400
1536)  GROUP="Status";;     #    1536 = 0x0600
2048)  GROUP="Accounting";;  #    2048 = 0x0800
2560)  GROUP="Resource";;   #    2560 = 0x0A00
```

```
3328)  GROUP="Kernel";;          #      3328 = 0x0D00
3584)  GROUP="Configuration";;  #      3584 = 0x0E00
3840)  GROUP="Syslog";;        #      3840 = 0x0F00
*)     GROUP="Unknown !?";;
```

```
esac
```

```
GW_NO_ALARM=1
```

```
export GW_NO_ALARM
```

```
sleep 1
```

```
/usr/sbin/snmptrap -v 1 -c public $SNMPTRAP .1.3.6.1.4.1.2789.2005
$FQHOSTNAME 6 2476317 ' ' .1.3.6.1.4.1.2789.2005.1 s "Flag: $1;
Gateway: $2; Date/Time:
${3:0:2}.${3:2:2}.${3:4}/${4:0:2}:${4:2:2}:${4:4:2}.${4:6:2};
Application: $APPL; Priority: $PRIO; Group: $GROUP ; Process ID :
$8 ; Text: $9."
```

```
#end
```

5 What Else Can I Do With SNMP?

You can use SNMP to query many more kinds of data. You can also have it react automatically to particular events. However, listing all these options here would go far beyond the scope of this document. Despite this, the examples in the sections that follow are designed to give you a brief insight into these options. To test the examples, add the commands associated with each example to the "snmpd.conf" file in the "/etc/snmp" directory. You will find more information about this on the <http://www.net-snmp.org/> website.

5.1 Monitoring processes

Use the "proc" command to monitor processes that are running on SGMG. For example, you can monitor the smtpd process. If too many smtpd processes are active at the same time, this may cause a problem when you want to send an e-mail to a specific recipient. In this example, an alarm attribute is set if more than 30 smtpd processes are running simultaneously:

```
#####
#####
# Process checks.
#
# proc NAME [MAX=0] [MIN=0]
#
# NAME: the name of the process to check for. It must match
#       exactly (i.e., http will not find httpd processes).
# MAX: the maximum number allowed to be running. Defaults to
#       0.
# MIN: the minimum number to be running. Defaults to 0.
#
# Verify that maximum 30 smtpd processes are running.
proc smtpd 30 0
# -----
-----
```

This extension would then return the following data:

```
snmpwalk -v 1 localhost -c public .1.3.6.1.4.1.2021.2
enterprises.ucdavis.procTable.prEntry.prIndex.1 = 1
enterprises.ucdavis.procTable.prEntry.prNames.1 = "smtpd"
enterprises.ucdavis.procTable.prEntry.prMin.1 = 0
enterprises.ucdavis.procTable.prEntry.prMax.1 = 30
enterprises.ucdavis.procTable.prEntry.prCount.1 = 1
enterprises.ucdavis.procTable.prEntry.prErrorFlag.1 = 0
enterprises.ucdavis.procTable.prEntry.prErrMsg.1 = ""
enterprises.ucdavis.procTable.prEntry.prErrFix.1 = 0
```

You can then configure the Management station so that an alarm is triggered when the error attribute is set.

5.2 Monitoring the load on SafeGuard MailGateway

You use the "Load" command to monitor system load. These values are calculated for the following time intervals: 1 minute, 5 minutes and 15 minutes. In this example an alarm attribute is set if an overload value exceeds its defined value. It therefore does not cause a problem if the SGMG is overloaded by 50% for a brief period.

```
#####
#####
# load average checks
#
# load [1MAX=12.0] [5MAX=12.0] [15MAX=12.0]
#
# 1MAX:   If the 1 minute load average is above this limit at
query
#         time, the errorFlag will be set.
# 5MAX:   Similar, but for 5 min average.
# 15MAX:  Similar, but for 15 min average.
#
# Check for loads:
load 50 15 15
# -----
-----
```

This extension would then return the following data:

```
# % snmpwalk -v 1 localhost -c public .1.3.6.1.4.1.2021.10
# enterprises.ucdavis.loadTable.laEntry.loadaveIndex.1 = 1
# enterprises.ucdavis.loadTable.laEntry.loadaveIndex.2 = 2
# enterprises.ucdavis.loadTable.laEntry.loadaveIndex.3 = 3
# enterprises.ucdavis.loadTable.laEntry.loadaveNames.1 = "Load-1"
# enterprises.ucdavis.loadTable.laEntry.loadaveNames.2 = "Load-5"
# enterprises.ucdavis.loadTable.laEntry.loadaveNames.3 = "Load-15"
# enterprises.ucdavis.loadTable.laEntry.loadaveLoad.1 = "0.49"
Hex: 30 2E 34 39
# enterprises.ucdavis.loadTable.laEntry.loadaveLoad.2 = "0.31"
Hex: 30 2E 33 31
# enterprises.ucdavis.loadTable.laEntry.loadaveLoad.3 = "0.26"
Hex: 30 2E 32 36
# enterprises.ucdavis.loadTable.laEntry.loadaveConfig.1 = "50.00"
# enterprises.ucdavis.loadTable.laEntry.loadaveConfig.2 = "15.00"
# enterprises.ucdavis.loadTable.laEntry.loadaveConfig.3 = "15.00"
# enterprises.ucdavis.loadTable.laEntry.loadaveErrorFlag.1 = 0
# enterprises.ucdavis.loadTable.laEntry.loadaveErrorFlag.2 = 0
# enterprises.ucdavis.loadTable.laEntry.loadaveErrorFlag.3 = 0
# enterprises.ucdavis.loadTable.laEntry.loadaveErrMsg.1 = ""
# enterprises.ucdavis.loadTable.laEntry.loadaveErrMsg.2 = ""
# enterprises.ucdavis.loadTable.laEntry.loadaveErrMsg.3 = ""
```

5.3 Adding scripts/program calls in SNMP

As you can extend the SNMP agent by using scripts or program calls, this provides you with an almost unlimited number of options for monitoring the SGMG. Scripts or programs are called with the "exec" command.

The example below shows how data about partitions and the mail queue can be provided using SNMP:

```
#####  
#####  
# Extensible sections.  
#  
  
exec .1.3.6.1.4.1.3159.50 df /usr/bin/df  
exec .1.3.6.1.4.1.3159.51 mailq /usr/bin/mailq  
  
# -----  
-----
```

You can also call scripts, like the "/bin/SGMGStatus" used in the next example:

```
#####  
#####  
# Extensible sections.  
#  
  
exec .1.3.6.1.4.1.3159.52 SGMGStatus /bin/sh /bin/SGMGStatus  
  
# -----  
-----
```

This script (see next page) determines the SGMG's current runlevel and identifies how many e-mails are currently being processed on the SGMG.

```
#!/bin/sh
echo Runlevel
runlevel
echo e-mails waiting for processing
ls /gateway/chroot/esmtp/processing/kw* 2> /dev/null | wc -l
echo e-mails being processed
ls /gateway/chroot/esmtp/processing/dir_* 2> /dev/null | wc -l
echo e-mails in bad
ls /gateway/chroot/esmtp/bad/* 2> /dev/null | wc -l
```

The data returned in this case is listed below.

```
# % snmpwalk -v 1 localhost -c public .1.3.6.1.4.1.3159.52
SNMPv2-SMI::enterprises.3159.52.1.1 = INTEGER: 1
SNMPv2-SMI::enterprises.3159.52.2.1 = STRING: "SGMGStatus"
SNMPv2-SMI::enterprises.3159.52.3.1 = STRING: "/bin/sh
/bin/SGMGStatus"
SNMPv2-SMI::enterprises.3159.52.100.1 = INTEGER: 0
SNMPv2-SMI::enterprises.3159.52.101.1 = STRING: "Runlevel"
SNMPv2-SMI::enterprises.3159.52.101.2 = STRING: "N 3"
SNMPv2-SMI::enterprises.3159.52.101.3 = STRING: "e-mails waiting
for processing"
SNMPv2-SMI::enterprises.3159.52.101.4 = STRING: "      0"
SNMPv2-SMI::enterprises.3159.52.101.5 = STRING: "e-mails being
processed"
SNMPv2-SMI::enterprises.3159.52.101.6 = STRING: "      0"
SNMPv2-SMI::enterprises.3159.52.101.9 = STRING: "e-mails in bad"
SNMPv2-SMI::enterprises.3159.52.101.10 = STRING: "      1"
SNMPv2-SMI::enterprises.3159.52.102.1 = INTEGER: 0
SNMPv2-SMI::enterprises.3159.52.103.1 = ""
```