

OMP Remote Control

Content

- [Activate remote control](#)
- [Functionality of remote control](#)
- [Simple Examples](#)
- [Example: Execute Local Security Checks in passive mode](#)

Introduction

The Greenbone Security Manager offers remote control via the SSL secured protocol "OMP". With OMP you have all the functionality that the web interface offers. In case you would like to implement your own OMP client for the Greenbone Security Manager, please refer to the comprehensive [OMP 3.0 specification](#). Please note that the examples below use OMP 2.0, older versions may differ slightly. Do not hesitate to contact our support team for additional help.

But you don't need to implement the protocol yourself. Greenbone Networks offers to prepare the OMP command line tool for all customers, for their specific operating system. The command line tool is available for a variety of GNU/Linux distributions. The [Greenbone Desktop Suite](#) for Microsoft Windows also includes the command line tool. If you are unable to find the command line tool for your operating system, please contact our technical support (see [contact](#)). Send your customer ID, the name and version of the operating system you want to run the command line tool on, and a description of the tasks you want to solve.

Activate remote control

As per default the remote control of GSM is not active.

Log into the CLI Admin interface as administrator and activate the OMP interface as described below. See also the manual "GSM Command Line Interface: Administrator Guide". **Please note**, that a reboot of the GSM will be necessary to activate the change.

```
gsm> set public_omp enabled
gsm *> commit
gsm> reboot
```

Analogously, you can of course deactivate the remote control:

```
gsm> set public_omp disabled
gsm *> commit
gsm> reboot
```

Functionality of remote control

The protocol OMP is XML based. Every request and every response is an XML entity.

The command line tool "omp" as delivered by Greenbone Networks on the one hand offers to directly send XML requests and receive XML responses. This is especially helpful for batch processing ("scripting"). On the other hand the most important OMP commands are implemented as command line parameters. Using these parameters provides a more human readable output of the OMP responses. This is intended for spontaneous requests, tests and the for working out future automated batch processes.

Basically there are two ways to use the command line tool "omp". With the switch "--xml" OMP commands are sent in XML format. The responses will be in XML format as well. Some of these commands are also available as command line parameters. For example, "--xml=<get_tasks/>" is equivalent to "--get-tasks". With the latter switch the output is not in XML format but in the form of a simple text table.

Simple Examples

These examples use a classic Unix-like shell. Thus the commands should easily integrate in similar environments.

For convenience the connection data is stored in the file "omp.config" in the home directory of the user. Under unix-like systems this is "\$(HOME)/omp.config" and under newer Windows-Systems "%USERPROFILE%\omp.config". Create this file with the following content (of course host, username and password need to be adjusted and remind that all contents is case-sensitive):

```
[Connection]
host=gsm
port=9390
username=demouser
password=demouser
```

Please take care that this file is readable and writable only by you before you enter the real connection data. (on Unix-like systems e.g. via "touch omp.config && chmod 600 omp.config").

Overview of existing tasks of the user in human readable formatted XML:

```
omp --pretty-print --xml="<get_tasks/>"
```

Overview of command line parameters of "omp":

```
omp --help
```

Overview of available OMP commands of the connected GSM:

```
omp --xml="<help/>"
```

Example: Execute Local Security Checks in passive mode

This example illustrates well how powerful the remote control feature can be: local security checks are executed for a target host without having GSM connect to the host, nor having to know its IP or even be able

to reach it via the network.

Preparation of the scan configuration

This step does not need to be done for each scan, but only for each desired scan configuration. This example Offline-LSC-Scan is about Unix-like systems (GNU/Linux, Solaris, HP-UX).

Import the scan configuration [Offline Unixoid LSC](#):



Execute scan

This example uses a classic Unix-like shell. Thus the commands should easily integrate in similar environments.

1. Initialize variables:

```
SCANCONFIG_NAME="Offline Unixoid LSC"
TASK_NAME="Offline LSC Scan RHEL5"
AUDIT_FILE="rhel5_example.audit"
```

SCANCONFIG_NAME: Must be identical to an existing scan configuration.

TASK_NAME: Can be chosen freely.

AUDIT_FILE: An audit file that was created on the target host, e.g. via [gb-lsc-agent](#). For a quick test you can download the above example audit file.

2. Create task:

```
TASK_UUID=`omp -c "$SCANCONFIG_NAME" -C --name "$TASK_NAME" -t Localhost`
iconv -f latin1 -t UTF-8 $AUDIT_FILE | sed 's/\\n;/g' | omp \
--modify-task $TASK_UUID --file --name /tmp/results.lsc
```

3. Start task:

```
REPORT_UUID=`omp --start-task $TASK_UUID`
```

4. Wait for the scan to finish:

With this command you get the status of the task:

```
omp --get-tasks $TASK_UUID
```

The scan is finished once the status reaches "Done".

```
96ebc7e4-d63d-405b-a21b-af20cf787bcd Done Offline LSC Scan RHEL5
```

5. Download report in the desired format:

```
omp --get-report $REPORT_UUID --format 1a60a67e-97d0-4cbf-bc77-f71b08e7043d > report.pdf
```

The corresponding REPORT_UUID was handed over when starting the scan.

The string "1a60a67e-97d0-4cbf-bc77-f71b08e7043d" describes the format which should be used, in this case PDF. You can use the following command to request a list of available format and their identifiers:

```
omp --get-report-formats
```

6. Clean up:

```
omp --delete-task $TASK_UUID
```

The corresponding TASK_UUID was handed over when starting the scan.