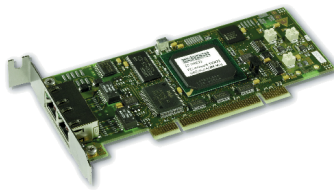


Application Notes

Local AntiVirus Pattern Update Server



mGuard smart



mGuard PCI



mGuard blade



mGuard industrial



mGuard delta

Innominate Security Technologies AG

Albert-Einstein-Straße 14

12489 Berlin

Germany

Phone: +49 (0)30-6392 3300

Fax: +49 (0)30-6392 3307

contact@innominate.com

<http://www.innominate.com/>

© Innominate Security Technologies AG

May 2006

“Innominate” and “mGuard” are registered trademarks of Innominate Security Technologies AG. All other brand names or product names are trade names, service marks, trademarks, or registered trademarks of their respective owners.

mGuard technology is protected by the German patents #10138865 and #10305413. Further national and international patent applications are pending.

No part of this documentation may be reproduced or transmitted in any form or by any means without prior written permission of the publisher.

All information contained in this documentation is subject to change without previous notice.

Innominate offers no warranty for these documents. This also applies without limitation for the implicit assurance of scalability and suitability for specific purposes.

In addition, Innominate is neither liable for errors in this documentation nor for damage, accidental or otherwise, caused in connection with delivery, output or use of these documents.

This documentation may not be photocopied, duplicated or translated into another language, either in part or in whole, without the previous written permission of Innominate Security Technologies AG.

“Windows” is a registered trademark of Microsoft Corporation in the United States and other countries.

“UNIX” is a registered trademark of The Open Group.

Table of Contents

1 Introduction.....	4
2 Technical Background.....	4
2.1 User View.....	5
2.2 How the Update Works.....	7
2.3 Quantification of Network Traffic.....	8
3 Local Update Server.....	8
3.1 Setting Up the Web Server.....	8
3.2 Populating the Document Tree.....	8

1 Introduction

Innominate Security Technologies AG (referred to hereafter as “Innominate”) distributes mGuard devices capable of scanning electronic mail and web downloads for viruses. The feature needs to be licensed separately. Once licensed, the devices' firmware runs a scan engine for this purpose.

Beginning with the firmware version 4.0.0, the Clam Antivirus (ClamAV for short) is used as a scan engine. This engine needs to be provided with updated patterns which tell it how to detect viruses. ClamAV must have these patterns as files at its disposal, stored locally on the mGuard device's flash RAM. Within the context of the mGuard devices, the files are called AntiVirus Patterns. The firmware of the mGuard devices has a mechanism for updating the AntiVirus Patterns automatically and on a regular basis.

Innominate provides a web server with recent AntiVirus Patterns for ClamAV to the customers who have licensed this feature. It offers recent AntiVirus Patterns for ClamAV. This web server is already preconfigured for firmware 4.0.0 but can be reconfigured.

This document describes how to set up a web server mirroring the AntiVirus Patterns from Innominate. Customers might need to set up such a mirror if their mGuard devices are not connected to the Internet or do not have access to Innominate's web server due to other restrictions (network policies, for example).

This is a technical document describing the background and technical options. *The existence and content of this document do not construct license nor permission to set up a web server mirroring the AntiVirus Patterns provided with Innominate's web server. For contractual arrangements with Innominate regarding this, please contact Innominate directly.*

2 Technical Background

Beginning with firmware version 4.0.0, the firmware for the variants of the different models of mGuard devices which support virus scanning run ClamAV as a scan engine. As described above, ClamAV needs to be provided with recent AntiVirus Patterns in order to be able to work efficiently. The firmware contains a mechanism capable of providing ClamAV with new AntiVirus Patterns on a regular basis. The default configuration for this mechanism is that no automatic updates of the AntiVirus Patterns are performed at all. Once you have activated the scan engine of your mGuard device, you should also make sure to activate a regular update of the AntiVirus Patterns. Please refer to your device's manual on how to do this. These guidelines merely repeat the most important steps in order to provide proper context.

In this chapter, you will learn how to configure regular updates for the AntiVirus Patterns, how the device's update mechanism checks for updated AntiVirus Patterns and downloads them, if required, and what needs to be provided by the web server for this mechanism to work correctly.

The update mechanisms are the same for all the mGuard models

- mGuard smart
- mGuard PCI
- mGuard bladePack
- mGuard industrial and
- mGuard delta

as well as for the variants of these models for which AntiVirus licenses are available (for example, the *mGuard smart enterprise XL*).

2.1 User View

In order to enable the mechanism which automatically updates the AntiVirus Patterns for his/her mGuard, the user must follow the steps as described within the corresponding user manual:

- a) Access the mGuard device's web based GUI via a browser.
- b) Open the "Management" menu and select the "Update" entry.
- c) Select the "AntiVirus Pattern" tab appearing on the right side. Now you should see a page similar to that in Illustration 1.
- d) Change the Update Schedule from "Never" to a value appropriate for your security needs.
- e) Optionally, you may change the update location by entering a different hostname.

- f) You may also optionally enter the connection details for the HTTP proxy server to be used exclusively for downloading the AntiVirus Pattern. This depends on your network environment.

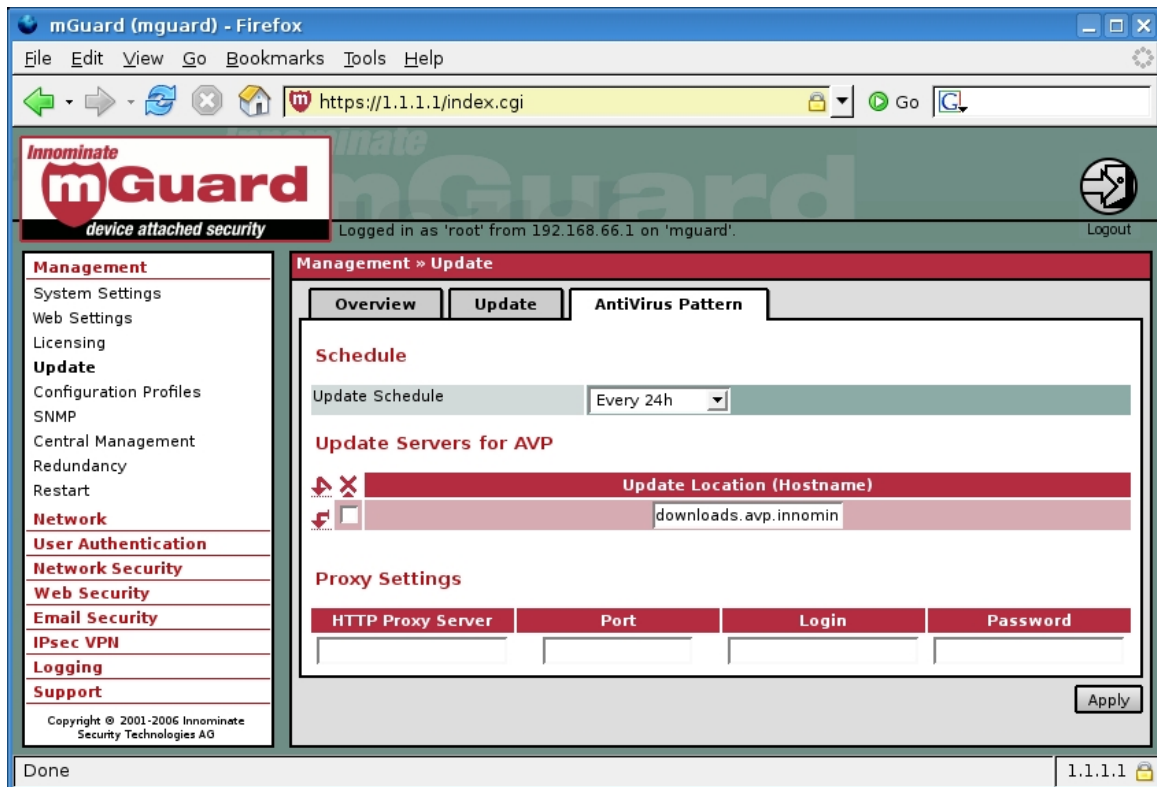


Illustration 1: Update Configuration and Installation Screen

- g) Enable at least one of the features provided in the menus “Web Security” or “Email Security” in order to start the scan engine and its update mechanism.
- h) By selecting the “Overview” tab, you may check which version of the AntiVirus Pattern has been downloaded recently for installation by the update mechanism. The information is displayed under the headline “AntiVirus Information” within the field labelled “Last Virus Update”. See Illustration 2 for an example.

2.2 How the Update Works

mGuard's update mechanism differentiates between a large file containing the *main database* updated at low frequency and a small file containing the *daily database* updated much more frequently. Those two files carry the name `main.cvd`, respectively `daily.cvd`. Despite its name, the file `daily.cvd` may be renewed with lower frequency, depending on the occurrence of new viruses.

The screenshot shows the mGuard web interface in a Firefox browser window. The address bar displays `https://1.1.1.1/index.cgi`. The page header includes the mGuard logo and the text "device attached security". Below the header, a navigation menu is visible on the left, and the main content area displays "Management > Update". The "Update" tab is selected, showing "System Information" and "AntiVirus Information".

System Information

Version	4.0.0.default
Base	4.0.0.default
Updates	[none]

AntiVirus Information

AntiVirus Engine Status	up
Last AntiVirus Update	main(38):21 Apr 2006 21:13 +0200 daily(1489):27 May 2006 15:47 +0200
AntiVirus Update Status	ok

Package Versions

Package	Number	Version	Flavour
bootloader	0	1.3.2	default
bridge-utils	0	0.9.5	default
busybox	0	1.1.6	default
bzip2	0	0.0.2	default
clamav	0	0.88.50	default
djbdns	0	1.5.1	default
eatables	0	0.3.0	default
...

The interface also includes a sidebar with various management options like System Settings, Web Settings, Licensing, Update, Configuration Profiles, SNMP, Central Management, Redundancy, Restart, Network, User Authentication, Network Security, Web Security, Email Security, IPsec VPN, Logging, and Support. The footer shows "Copyright © 2001-2006 Innominate Security Technologies AG" and "Done" status.

Illustration 2: Version Information about the AntiVirus Pattern

The files are digitally signed, so that the update mechanism of the mGuard device can verify their integrity and authenticity. The mGuard device's update mechanism expects to be able to retrieve these files from the given Update Location's hostname via HTTP. More precisely, it expects those files to be available from the named web server's document root. The web server must not impose any access restrictions upon the files, and in particular, it must not require any authentication before download.

Starting from the point in time in which the user has configured an Update Schedule for the AntiVirus Patterns other than "Never", the update mechanism of the mGuard will check for more recent versions of the file `main.cvd`, resp. `daily.cvd`, available from the update location. Whenever it finds more recent versions, these are downloaded and verified

for integrity and authenticity. Once verified, they are installed for use by the scan engine. In other words, there is no need for user interaction. Updates of the AntiVirus Patterns will occur automatically. The verification as to whether any of the files `main.cvd`, resp. `daily.cvd`, is more recent causes just a few kilobytes of traffic per device if the files installed on the device are up-to-date.

Under the mGuard devices' default configuration, the specific URLs for downloading the files `main.cvd`, resp. `daily.cvd`, are

<http://downloads.avp.innominat.com/main.cvd> and

<http://downloads.avp.innominat.com/daily.cvd>.

The hostname can be customised via the mGuard device's configuration.

2.3 Quantification of Network Traffic

Whenever a single mGuard device's AntiVirus Pattern update mechanism checks the update server for new files, the mGuard device retrieves the file's header plus its first 512 bytes. If the update mechanism considers the file on the server to be more recent, the complete file is retrieved with a subsequent HTTP request. In each instance, both files, `main.cvd` and `daily.cvd`, are checked.

3 Local Update Server

This chapter provides technical details on how to install and operate an AntiVirus Pattern update server.

3.1 Setting Up the Web Server

As explained above, the mGuard device simply needs to retrieve two files from the AntiVirus update server. This is carried out with the help of simple GET commands sent via HTTP to the update server's web server. Therefore, the update server is merely a plain web server.

3.2 Populating the Document Tree

In order to populate the update server's document tree, you simply need to set up a mechanism looking for new files that are available from `downloads.avp.innominat.com`. The program `wget` is in the Open Source (see <http://www.gnu.org/software/wget/>) and is well suited for this task. While `wget` was developed for *UNIX* operating systems, a port exists for the *Windows* systems family as

well. For *UNIX* operating systems, you just need to have *wget* called from a crontab at the frequency you desire. Under the *Windows* systems family, you can use the task scheduler to carry this out.

For both kinds of systems, Innominate recommends wrapping *wget* with a small shell script or batch (.bat) / command (.cmd) file. This chapter depicts an example for *UNIX*-like operating systems only. An example script for downloading the files `daily.cvd` and `main.cvd` can be found below.

```
#!/bin/sh

URL="http://downloads.avp.innominate.com"
TMPDNLD="/var/www-tmp"
DESTDIR="/var/www-http"
LOCKFILE="${TMPDNLD}/LOCK"

die() { test -n "$1" && echo "$0: $1" >&2; exit 1; }
removeLock() { rm "${LOCKFILE}" || echo "Can't remove ${LOCKFILE}" >&2; }

umask 022
test -e "${TMPDNLD}" || die "The directory ${TMPDNLD} is missing."
test -e "${DESTDIR}" || die "The directory ${DESTDIR} is missing."
test -e "${LOCKFILE}" && die "Download for ${FILE} is already locked."
touch "${LOCKFILE}" || die "Can't create lockfile."
trap removeLock EXIT

for FILE in main.cvd daily.cvd
do
    TMPFILE="${TMPDNLD}/${FILE}"
    DESTFILE="${DESTDIR}/${FILE}"
    rm -f "${TMPFILE}"
    if [ -e "${DESTFILE}" ]; then
        cp -p "${DESTFILE}" "${TMPFILE}" \
            || die "Can't copy current file ${DESTFILE} to ${TMPFILE}."
    fi
    wget -P "${TMPDNLD}" -q -N -nd "${URL}/${FILE}" \
        || die "Error while downloading ${URL}/${FILE}"
    mv "${TMPFILE}" "${DESTFILE}" \
        || die "Can't move ${TMPFILE} to ${DESTFILE} to make it public."
done
```

Example of a Download Script

For proper operation of the script, the temporary download area described by the variable `TMPDNLD` needs to be a directory which exists within the same file system as the directory denoted by the variable `DESTDIR`. The directory named by `DESTDIR` must be the one which contains the document root of your web server. Please adjust these variables, if required.

The example assumes that the program *wget* and others can be found via the current `PATH` environment. Please adjust your `PATH` environment or set the command specifications in such a way that they have an absolute path. Please make sure that the script is invoked with appropriate access rights, i.e. with the user rights dedicated to maintaining your update server's content only.

The example script thoroughly takes care of

– minimizing the network traffic it causes and

– atomically providing the recently downloaded file in whole or leaving an older file intact.

The network traffic is minimised with the help of `wget`'s time-stamping option (`-N`). This option compares the modification time and size of a locally present file with the one offered by the remote HTTP server. The remote HTTP server provides the modification time of the file it offers with the help of the `Last-Modified` header. Then `wget` will use this time for the downloaded file's modification time as well.

The script also ensures that only complete files are downloaded without error. In addition to the mGuard device checking each AntiVirus Pattern's integrity and authenticity, a robust update script running on your local update server will help you to update newly deployed mGuard devices reliably. The robustness of the script is achieved by a three step process:

1. If present, a formerly downloaded file is copied from your local server's document root to a temporary download area. `wget` will use that file's modification time to compare it with the one offered by the remote server. This is why the modification time of the copy needs to have the same modification time as the one present in the document root.
2. If the remote file differs in size or is newer, `wget` will download the file and thereby modify the locally present file directly. This is why `wget` must operate on a copy and not the public version.
3. Only if the download has been completed without error will the downloaded file be moved (not copied!) to the document root of your web server. This move will preserve its modification date. If a former file existed, it will be replaced atomically by the move operation.