Cloudmark Cartridge

Installation and Administration Guide

Cloudmark, Inc. 128 King Street, 2nd Floor, San Francisco, CA 94107 USA

Cloudmark Europe, Ltd. Carmelite, 50 Victoria Embankment, Blackfriars, London EC4Y ODX UK

Cloudmark Cartridge version 3048
Last modified: March 11, 2008

Contents

CHAPTER 1

Introduction

Cloudmark's gateway solutions use the Cloudmark Cartridge to deliver the latest Cloudmark anti-abuse technology for your email platform. This guide explains how to install, configure, and administer the cartridge. You can find out what's new in this version of the cartridge in "What's new in Cartridge 3048" below.

The rest of this chapter introduces the technology behind the Cloudmark Cartridge:

What's new in Cartridge 3048

Cartridge 3048 includes these changes:

- The Cartridge now keeps track of its last known viable state and returns to it upon restarting after a crash.
- A new configuration key specifies an alternate port on which to download micro-updates. See "micro-update port" on page 16.
- A new configuration key controls how the sending IP address is determined. See "use envelope for ip information" on page 16.
- The default value for the "use ip information" configuration key is now "yes." See "use ip information" on page 16.
- The micro-update file set now includes .fsl and .xrl metadata files. See "Data files" on page 20.

- A new fingerprinting scheme provides faster processing.
- A new statistics field reports your unique installation ID. See "What statistics are collected" on page 31.

Cloudmark fingerprinting algorithms

The Cloudmark Cartridge includes Cloudmark's fingerprinting algorithms, designed to target the most current spamming techniques. Using these algorithms, the Cloudmark Cartridge generates a set of fingerprints for each incoming message.

The Cloudmark Cartridge maintains a cache of all fingerprints that have a known classification, such as spam, phishing, or virus fingerprints. The fingerprints of an incoming message are compared to these known fingerprints, and a message score is generated. This list of known fingerprints is regularly updated with the latest data from the Cloudmark Global Threat Network, using the micro-updates mechanism. See "Cloudmark Global Threat Network" below and "Micro-updates" on page 3.

Cloudmark Global Threat Network

Cloudmark's community of millions of end users provides constant, real-time feedback about which messages are considered spam, phishing, or email-borne viruses, and which ones are considered legitimate. The Trust Evaluation System (TES) assigns each user a trust level based on how well the user's feedback concurs with that of other trusted users. Less-trusted users have less influence over network-wide message classification, while the most trusted users have more influence.

When a sufficient number of trusted users block a certain message as junk, this message's fingerprint is flagged. Information about the fingerprint is distributed throughout the network to automatically block that message (and all its permutations) for other users.

Micro-updates provide the latest known fingerprints as determined by the Cloudmark Global Threat Network. By using micro-updates, you protect your platform against the most current email-borne threats. See "Micro-updates" below.

Micro-updates

Cloudmark stores message fingerprints generated though the Global Threat Network in near-real-time. Micro-updates are the mechanism that allows Cloudmark customers to download the latest fingerprint data at regular intervals.

Micro-updates enable Cloudmark to

- maintain the highest level of accuracy on spam, virus, and phishing messages as well as legitimate messages
- handle new varieties of email threats proactively and automatically
- reduce false positives
- eliminate manual message analysis with a fully-automated approach

To maintain the highest possible accuracy, the micro-updates feature must be correctly configured. For complete information, see Chapter 4, “Micro-Updates”.

Message scoring

When the Cloudmark Cartridge scans a message, it assigns a spam score (as a percentage) to indicate the likelihood that the message is an abusive message (such as spam, phishing, or a virus). For example, if the cartridge assigns a message a score of 99, it means that Cloudmark is 99% certain that message is bad; a score of 1 means that Cloudmark is almost certain the message is legitimate.

When used in conjunction with Cloudmark Authority Engine SDK (CMAE SDK) 2.0 or later, the cartridge may also provide information about each message’s classification (spam, phishing, virus, and so on). Consult your vendor to find out whether your implementation of the CMAE SDK supports this feature.

You can establish your own policies for handling spam, and configure your application to take action on a message based on its spam score. Such actions typically include one or more of the following:

- storing spam in a designated folder
- flagging spam messages in the Subject field
- deleting spam
- returning spam to its original sender

Message categorization

When scoring a message with the Cloudmark Authority Engine SDK's CMAE_Score() function, an application can request that the cartridge return a category and a subcategory for the message. Categories and subcategories are expressed as integers, which are mapped to categories in the .cats file. See the .cats file for the list of categories.

For example, using the Authority Engine SDK, the following call produces a message score and category:

```
CMAE_Score(CMAE_Envelope Envelope,
        const char *RFC822Content, size_t RFC822ContentLength,
        unsigned int *ScoreOut,
        unsigned int *CategoryOut, unsigned int *SubCategoryOut,
        unsigned int *RescanOut, char **AnalysisOut);
```

If CategoryOut is 7and SubCategoryOut is 0, then the cartridge has categorized the message as a virus message. The following call provides more information:

```
CMAE_DescribeCategory(unsigned int Category, unsigned int SubCategory,
        char **CategoryDescOut, char **SubCategoryDescOut);
```

With Category=7 and SubCategory=0, CategoryDescOut would contain an allocated string "virus", and SubCategoryDescOut would contain an allocated string "undefined".

For detailed information about using this feature in your application, see the *Cloudmark Authority Engine SDK Guide*.

Cartridge statistics

By default, the Cloudmark Cartridge sends cartridge configuration information and message scanning statistics back to Cloudmark. By collecting this information, Cloudmark can more effectively detect potential accuracy issues and proactively address them before there is a need for the customer to contact Cloudmark. If your organization has special privacy concerns, contact Cloudmark.

For complete information, see Chapter 6, "Cartridge Statistics Reporting".

Whitelisting

A whitelist is a list of trusted senders from whom you always accept email, or email characteristics which indicate a trusted message. This feature of the Cloudmark Cartridge minimizes the filtering of legitimate messages and allows system administrators to conveniently manage the receipt of messages from known safe senders.

For complete information, see Chapter 5, “Whitelisting”.

CHAPTER 2

Cloudmark Cartridge Installation

This chapter provides the Cartridge installation instructions:

- "The Cartridge installation package" below.
- "Installing or updating the Cartridge" on page 8

! *Be sure to refer to the release notes of each Cartridge version for special installation instructions.*

The Cartridge installation package

The Cartridge installation package is provided in either a TAR or a ZIP file, depending on your platform. Before installation, verify that the installation package contains all the required installation files.

Below is a list of the components in a standard Cartridge installation package:

- etc/micro_updates/<dpl_version_number>.dpl
- etc/micro_updates/<rpl_version_number>.rplv1
- etc/micro_updates/<awl_version_number>.awl
- etc/micro_updates/<acf_version_number>.acf
- etc/micro_updates/<csl_version_number>.csl
- etc/micro_updates/<fsl_version_number>.fsl
- etc/micro_updates/<xrl_version_number>.xrl
- etc/micro_updates/<mpl_version_number>.mpl
- etc/micro_updates/<cats_version_number>.cats
- etc/micro_updates/<mfl_version_number>.mfl

- etc/micro_updates/<impl_version_number>.implv1
- etc/micro_updates/states/srl_set.package
- etc/whitelist.cfg.sample
- etc/cartridge.cfg.sample
- lib/cartridge.so

Additional files are downloaded as micro-updates. For more information about these files, see "Data files" on page 20.

Installing or updating the Cartridge

Follow the installation instructions for the product with which you are using the Cartridge:

- "Installation for Cloudmark Authority Engine-based products" below
- "Installation for the Cloudmark Authority Plug-In for SpamAssassin" on page 9
- "Installation for Cloudmark Immunity" on page 10
- "Installation for Openwave Email Mx" on page 11
- "Installation for Openwave Edge Gx" on page 12

These instructions apply to both new installation and updates to existing Cartridge installations.

Installation for Cloudmark Authority Engine-based products

To install the Cartridge for Authority-based products

1. Stop the server/service using the Cloudmark Authority Engine.
2. If you are updating an existing Cartridge installation, remove all of the files in the etc/micro-updates/ directory, as well as the etc/micro-updates/states/ subdirectory.

   The new Cartridge will download the correct files with which to re-populate this directory.
3. Place the compressed Cartridge file in the product home directory.
4. Decompress it.
   - For Linux/Solaris, extract the Cartridge with the following command:

```
gzip -d -c < SpamDNA-3048.x.x.x-<platform>.tar.gz | tar xvf -
```

- For Windows installation, double-click the .zip file, then click Extract.

5 Create the etc/license.cfg file.

This file must contain the two-line license text that you received from Cloudmark.

6 If you are updating an existing Cartridge installation, update your cartridge.cfg to the latest defaults listed in the file etc/cartridge.cfg.sample.

7 Restart the server/service using the Cloudmark Authority Engine.

8 Check for the following log message:

```
INFO:MICROUPDATE: Successfully updated <file> from network
(new serial <serial>)
```

There should be one such message for every micro-updates file listed in "The Cartridge installation package" on page 7.

See also the *Cloudmark Authority Engine SDK Guide*.

Installation for the Cloudmark Authority Plug-In for SpamAssassin

During a new installation of the Cloudmark Authority Plug-In for SpamAssassin, the Cartridge is installed automatically. To upgrade the Cartridge for an existing installation of the plug-in, use the instructions below.

To upgrade the Cartridge for the SpamAssassin Plug-In

1 Become the superuser.

2 Switch to the SpamAssassin CMAE subdirectory:

```
cd /etc/mail/spamassassin/cmae/cloudmark
```

This path may vary in your installation. Make sure you are in the cloudmark subdirectory.

3 Remove all of the files in the etc/micro-updates directory.

The new Cartridge will download the correct files with which to re-populate this directory.

4 Extract the Cartridge:

```
gzip -d -c < SpamDNA-3048.x.x.x-<platform>.tar.gz | tar xvf -
```

5 If you are upgrading from Cartridge 3046 or earlier, create the etc/license.cfg file.

   This file must contain the two-line license text that you received from Cloudmark.

6 Update your cartridge.cfg with the latest defaults listed in the file etc/cartridge.cfg.sample.

7 Restart the CMAE server:

```
bin/cmaed restart
```

Installation for Cloudmark Immunity

To install the Cartridge when using Cloudmark Immunity, follow the steps below:

To install the Cartridge for Immunity

1 Stop Immunity.

2 If you are updating an existing Cartridge installation, remove all of the files in the etc/micro-updates directory.

   The new Cartridge will download the correct files with which to re-populate this directory.

3 Place the compressed Cartridge update file in the product home directory.

4 Decompress it.

   - For Linux/Solaris, extract the Cartridge with the following command:

```
gzip -d -c < SpamDNA-3048.x.x.x-<platform>.tar.gz | tar xvf -
```

   - For Windows, double-click the .zip file.

5 Create the etc/license.cfg file.

   This file must contain the two-line license text that you received from Cloudmark.

6 If you are updating an existing Cartridge installation, update your cartridge.cfg with the latest defaults listed in the file etc/cartridge.cfg.sample.

7 Restart Immunity.

   If you are installing Cloudmark Immunity for the first time, complete the following additional steps:

8 Manually copy the file new_cm_egm.db.temp (externally provided) to the following location before running dbsetup.pl:

   - (Immunity 2.0.2) <immunity root>/data/new_cm_egm.db.temp

- (Immunity 2.0.1) <immunity root>/setup/sql/sqlite/cm_egm.db

(<immunity root> is typically /srv/immunity)

9 The default cartridge.cfg and whitelist.cfg files will not be installed in etc/. To create the default configuration files, copy etc/cartridge.cfg.sample to etc/cartridge.cfg and etc/whitelist.cfg.sample to etc/whitelist.cfg.

10 Check for the following log message:

```
Aug 21 12:28:06 : INFO:MICROUPDATE: Successfully updated
<file> from network (new serial <serial>)
```

There should be one such message for every micro-updates file listed in "The Cartridge installation package" on page 7.

See also the *Cloudmark Immunity Installation and Administration Guide.*

Installation for Openwave Email Mx

Before proceeding with the installation instructions below, Cloudmark recommends backing up the entire $AUTH_HOME directory.

To install the Cartridge for Openwave Email Mx

1 Follow these steps:

```
cd $AUTH_HOME
```

(AUTH_HOME=value from configuration key /<host>/authority/homeDir). You should see an "etc" and a "lib" directory here.

```
cp lib/cartridge.so lib/cartridge.so.<datestamp>
$INTERMAIL/lib/imservctrl stop imextserv
gzip -d -c < SpamDNA-3048.x.x.x-<platform>.tar.gz | tar xvf -
```

2 Create the etc/license.cfg file.

This file must contain the two-line license text that you received from Cloudmark.

3 If you are updating an existing Cartridge installation, update cartridge.cfg to the latest defaults listed in the file etc/cartridge.cfg.sample.

4 Start the extensions service:

```
$INTERMAIL/lib/imservctrl start imextserv
```

5 Check for the following log message:

```
Aug 21 12:28:06 : INFO:MICROUPDATE: Successfully updated
<file> from network (new serial <serial>)
```

There should be one such message for every micro-updates file listed in "The Cartridge installation package" on page 7.

See also the *Cloudmark Authority Engine Extensions Service for Openwave Email Mx Administration Guide.*

Installation for Openwave Edge Gx

Before starting this procedure, Cloudmark recommends backing up entire $AUTH_HOME directory.

TO INSTALL THE CARTRIDGE FOR OPENWAVE EDGE GX

1. Stop the server:

   ```
   $INTERMAIL/lib/imservctrl stop imextserv
   ```

2. Switch to the $AUTH_HOME directory:

   ```
   cd $AUTH_HOME
   ```

   (AUTH_HOME=<value of key /<host>/cloudmarkeas/homeDir>). You should see an “etc” and a “lib” directory here.

3. Back up the old cartridge.so file:

   ```
   cp lib/cartridge.so lib/cartridge.so.<datestamp>
   ```

4. Decompress the Cartridge installation package:

   ```
   gzip –d –c < SpamDNA-3048.x.x.x-<platform>.tar.gz | tar xvf –
   ```

5. Create the etc/license.cfg file.

   This file must contain the two-line license text that you received from Cloudmark.

6. Install the whitelist configuration file, if it does not already exist:

   ```
   cp etc/whitelist.cfg.sample etc/whitelist.cfg
   ```

7. Use imconfedit to edit the config.db and make all applicable changes. Update the cartridgeParameters key with values as specified in the cartridge.cfg.sample file.

8. Start the server:

   ```
   $INTERMAIL/lib/imservctrl start imextserv
   ```

9. Check for the following log message:

   ```
   Aug 21 12:28:06 : INFO:MICROUPDATE: Successfully updated
   <file> from network (new serial <serial>)
   ```

   There should be one such message for every micro-updates file listed in “The Cartridge installation package” on page 7.

CHAPTER 3

Cloudmark Cartridge Configuration

This chapter discusses the configuration settings in the cartridge.cfg file. Please also refer to the Cartridge release notes for updates to these settings.

You can configure Cartridge parameters in the cartridge.cfg file – located in the etc/ directory in the root directory of your Cloudmark installation – using any text editor. Table 1 below lists the configuration parameters, as specified in cartridge.cfg. All configuration parameters are case-insensitive. If cartridge.cfg is not available, the default settings are used.

Table 1 *Micro-updates configuration settings*

Parameter	Value(s)	Default	Description
consider empty messages spam	yes *or* no	no	Specifies whether messages with empty subject and empty body, or a subject or body consisting only of whitespace, are considered spam.
customer id	Company name *or* CNFS login	none	Specify your organization's name. This helps identify your reports when they reach Cloudmark. If you are a Cloudmark Network Feedback System (CNFS) customer, use your CNFS logon name for this parameter.
download micro-updates before init	yes *or* no	no	Specifies whether micro-updates are downloaded before startup.
enable micro-updates	yes *or* no	yes	Enables (or disables) the download of micro-updates over the network from the micro-update host. Setting this option to "no" will adversely affect accuracy.

Table 1 *Micro-updates configuration settings*

Parameter	Value(s)	Default	Description
exclude from stats reports	whitelist *or* proxy auth *or* whitelist, proxy auth	none	If present, the corresponding values in statistics reports will be replaced by the string "<excluded>".
favor analysis over speed	yes *or* no	no	When set to "yes", the Cartridge calculates all fingerprints before returning a result for the message. When set to "no", the Cartridge stops when it finds one known fingerprint that matches the message, or when it finds a matching whitelist entry.
gateway received hosts	hostname *or* IP address	NULL	This parameter specifies a comma-separated list of hostnames or IP addresses of the email gateway hosts in your network. The value should match that of the Received header in messages that enter your network from the outside. If no value is configured, heuristics are used instead.
http proxy	host:[port] *or* ip:[port]	NULL	Specifies the http proxy for connecting to the Cloudmark micro-updates service.
http proxy basic auth	user:password	NULL	Specifies user and password for HTTP proxy basic authentication. A colon (:) is not allowed in usernames and passwords.
http proxy ntlm auth	domain:username:password	NULL	Enable NTLM authentication with a proxy server. A colon (:) is not allowed in domains, usernames, and passwords.

Table 1 *Micro-updates configuration settings*

Parameter	Value(s)	Default	Description
image processing depth	none *or* low *or* medium *or* high	high	When analyzing images, the Cartridge can apply a variety of fingerprinting algorithms, some more resource-intensive than others. The default value of "high" applies all image-specific fingerprinting algorithms, achieving the highest possible accuracy. If you observe excessive CPU load while processing very high message volumes, you can adjust this value to use fewer resources when processing images (at the expense of accuracy).
local address for m-u cons	IP address	NULL	Forces the Cartridge to bind to the specified IP address for all micro-update connections, whether direct or proxied.
micro-update cache path	File path	<etcdir>/ micro-updates	Specifies the directory on the local server where micro-updates data files will be stored.
micro-update hostname	hostname	microupdates. cloudmark.com	Specifies the hostname to connect to when downloading micro-updates. ! *If you change this hostname, you must clear the .../etc/<micro-update cache path>/micro_updates directory at the same time. Failure to do so may result in degraded accuracy.*
micro-update interval	Positive integer *or* 'auto'	auto	Specifies how frequently (in hours) to check for the latest micro-update; if set to auto, the software checks for deltas on a periodic basis.

Table 1 *Micro-updates configuration settings*

Parameter	Value(s)	Default	Description
micro-update port	Positive integer	80	Connect to an alternate port on the host specified by micro-update hostname. Note that Cloudmark's micro-update servers only accept connections on port 80.
micro-update timeout	Positive integer	60	Specifies the timeout period (in seconds) for HTTP requests used when checking for micro-updates.
report statistics	yes *or* no	yes	Enables (or disables) the communication of message scanning statistics to Cloudmark. ! *As of Cartridge 3047, this setting is overridden by Cloudmark's back-end servers. Statistics reporting is now mandatory.*
use envelope for ip information	yes *or* no	no	When this option and "use ip information" are *both* set to "yes", then the sending IP address is derived from the message envelope instead of the Received headers. This is a useful option when scoring a message that does not yet contain the Received header that reflects the hop into the MTA.
use ip information	yes *or* no	yes	When set to "yes", the Cartridge uses the sender IP address of the message as one of its fingerprints.

CHAPTER 4

Micro-Updates

Micro-updates is the mechanism that allows the Cloudmark Cartridge to regularly download the latest fingerprint data used to identify abusive messages. The fingerprint data is provided by highly trusted reporters and analysis by the Trust Evaluation System in the Cloudmark Global Threat Network in near real-time. Therefore, proper use of the micro-update mechanism is critical in maintaining Cartridge accuracy as new types of spam, phishing, and virus messages are reported.

! *Your installation must have a valid license file in order to download and process micro-updates. If you do not have a license file, please request one from Cloudmark.*

This chapter includes the following topics:

Micro-update frequency

The administrator can set the frequency at which the software downloads the micro-updates in one of two ways:

- using automatic updates

  This is the default setting. See "Automatic micro-updates" on page 18.

- setting a specific update interval

  This setting is not recommended, as it may compromise the accuracy of filtering. See "Updates at user-specified intervals" on page 18.

These options are configured using the 'micro-update interval' configuration setting in the cartridge.cfg file. See also Chapter 3, "Cloudmark Cartridge Configuration".

Automatic micro-updates

By setting the "micro-update interval" configuration setting to 'auto', the Cloudmark Cartridge will automatically download the latest micro-update information at the interval defined by Cloudmark in the .acf file. When the time intervals are updated, the Cartridge will automatically download a new version of the .acf file.

By default, the Cartridge checks for the availability of changes to the micro-update data every minute. The entire micro-update is downloaded once every three hours and saved to disk so that the data can be used offline if necessary.

! *The default time intervals as stated above may change as Cloudmark releases new .acf files.*

Cloudmark strongly recommends that customers use the 'auto' setting for micro-updates.

Updates at user-specified intervals

You can specify an interval (in hours) for the "micro-update interval" setting. If an update is available at that interval, the full micro-update will be downloaded and saved to the etc/micro_updates directory. The two latest versions of the micro-updates data will be kept on disk. Older versions will be deleted automatically.

! *This configuration is not recommended, as it may compromise the accuracy of filtering.*

Network interaction

Micro-updates are downloaded using standard HTTP requests. If an HTTP proxy is not enabled, then at the specified interval, a download will be attempted over port 80 by default, or the port configured by "micro-update port" on page 16. If the filtering server is unable to connect to the Cloudmark micro-updates service, the most recent offline version of the data file in the etc/micro_updates directory will be used until the next micro-updates interval before attempting to download again.

! *Micro-updates cannot be downloaded if a valid license file is not present.*

Using HTTP proxies

To use an HTTP proxy, set the "http proxy" configuration parameter in cartridge.cfg to your proxy server, in one of the following forms:

```
<host>:<port>
<address>:<port>
```

If HTTP proxying is enabled, then the port specified will be used.

If the proxy uses HTTP basic authentication, then set the "http proxy basic auth" parameter, in the following form:

```
<user>:<password>
```

If the proxy uses NTLM authentication, then set the "http proxy ntlm auth" parameter, in the following form:

```
<domain>:<username>:<password>
```

See also Chapter 3, "Cloudmark Cartridge Configuration".

Connection timeout logic

The "micro-update timeout" parameter sets the timeout that will be used for all network calls. Therefore, this setting is not related to the actual download time—just the network call.

Data files

The Cloudmark service currently generates new versions of the full micro-updates data files at intervals designed to balance bandwidth usage with the required amount of updates. Delta updates are generated at more frequent intervals to complement the micro-updates files.

! *New files will only be generated if new data is available.*

Micro-updates consist of the file types listed below:

metadata files The metadata files have extensions of .mfl, .mpl, .rplv1, .dpl, .awl, .ipl, .implv1, .csl, .fsl, .xrl, .cfg, or .cksum.txt.

state files The etc/micro-updates/state directory contains files that track the state of the micro-update file set: srl_set.safe, srl_set.unsafe, and srl_set.package.

signature files The signature files have the extension .hs2.z.aes. Delta micro-updates data are not updated in the hs2.z.aes file. Therefore, the serial number of the hs2.z.aes file is not updated for each successful delta micro-updates download. The numerical portion of the zdata.txt file can be higher than the serial number of the hs2.z.aes file during normal operation.

checksum files There is one checksum file per signature file. These guarantee the integrity of the data. Each checksum file also contains the decryption key for its corresponding signature file.

category file This file defines the categories and sub-categories returned by the Cartridge. See “Message categorization” on page 4.

! *This file may be overwritten by micro-updates; do not modify it.*

Offline files

At least one micro-update must be downloaded in order for the Cartridge to operate offline. If the Cartridge is unable to download the next micro-update, the previous micro-update file set is used.

When a valid set of data files is downloaded from the micro-updates service, they are saved to the <micro-update cache path>/micro_updates/ directory with version numbers in the filenames. New files are loaded from disk at startup.

! *Delta micro-updates files are read into memory and are not kept on disk.*

Data file integrity and security

The micro-update files containing data are compressed and encrypted. Data that is not encrypted with the correct key will be ignored. As a result, DNS or IP spoofing is not a concern. The contents of the data files are read into memory at startup, as well as each time a new file is downloaded.

Advanced micro-update configurations

Cloudmark provides flexible options for complex customer deployments. If your organization has constraints that restrict you from enabling automatic micro-updates downloads from the Cloudmark micro-updates service, the following advanced configurations may allow you to use micro-updates within the constraints of your organization.

Using an HTTP proxy

If your organization has constraints that prohibit the Cloudmark software from accessing the Internet directly, you may enable an HTTP proxy to direct all traffic through the proxy server.

For configuration information, see “Using HTTP proxies” on page 19.

CHAPTER 5

Whitelisting

The Cloudmark product provides system-level whitelisting support, allowing you to pass messages automatically based on domains, IP ranges, envelope, header or body features. Whitelisting configuration settings are stored in the file whitelist.cfg located in the etc/ directory.

This chapter introduces the four types of whitelisting, plus a sample configuration:

! *For products based on versions CMAE SDK prior to 2.0, changes to the whitelist take effect upon a restart of the product. CMAE SDK 2.0 has the ability to reload whitelists without a restart.*

Host whitelisting

Host whitelisting checks the IP address or the domain of the mail server from which a message is being received. While host whitelisting is a highly effective anti-spam measure, it requires that the Cloudmark product have the address of the connecting MTA. This usually means that the connecting MTA must connect directly to the MTA that hosts the Cloudmark product.

Following are example configurations that allow mail from hosts with the specified IP addresses to bypass spam-filtering:

```
type=host; address=[1.2.3.4]
type=host; address=[1.2.3]
type=host; address = [192.168.32.0/24];
```

In the first example, only mail from the host at the exact IP address bypasses spam filtering. The second example allows mail to bypass spam-filtering if only the first three octets of the IP address match. The third example shows an entry in CIDR format.

! *A CIDR mask of /0 is invalid.*

In the case of hostname whitelisting, a domain suffix match will be performed to determine if an email should bypass spam filtering. Here is an example:

```
type=host; address=[bar.com]
```

In this case, if the server's IP address resolves to bar.com or anything.bar.com, the email will bypass the spam filter. As long as there is an exact match on the right side of the hostname, then the email will be whitelisted.

! *It is important to understand that host whitelisting does not make use of regular expressions. The matches must be exact prefix matches in the case of IP addresses and exact suffix matches in the case of hostnames.*

Header whitelisting

Header whitelisting checks the header fields of incoming mail. The header whitelist value is a regular expression that is applied to the whole header.

Header whitelisting is not foolproof, because spammers sometimes forge (or "spoof") header information. Additionally, a header may be added to a spam message when it is relayed by a trusted host and could then be passed to the recipient without being scanned for spam.

Following are example configurations which use header whitelisting to bypass spam filtering for all addresses from the .gov and doj.org domains, respectively:

```
type=envelope; command=[mail from]; value=[@.*\.gov\b];
type=envelope; command=[rcpt to]; value=[@.*\.doj\b];
```

To match an explicit email address:

```
type=header; header=[From]; value=[\buser@domain\.com\b];
```

Body whitelisting

Body whitelisting checks the body of the email to see if it matches the given regular expression. This type of whitelisting should only be used in cases where you want to ensure delivery of a message that is considered spam by the Cloudmark community.

Suppose that you want to whitelist automatic email messages from a house-hunting site that contains an HTML header:

```
type=body; regex=[<h1>HomeHunters</h1>];
```

It is important to understand that complex regular expressions, when applied to the entire email body, can adversely affect performance. For this reason, it is best to stick to very explicit match patterns that do not use too many '*' or '+' patterns.

Envelope whitelisting

Envelope whitelisting checks the SMTP envelope of an incoming message. This provides a flexible whitelisting mechanism that will match any substring or regular expression in the specified command field of the envelope. Like header whitelisting, spammers can forge (or "spoof") the From field to create the illusion that the message comes from a trusted source.

Following are sample configurations for envelope whitelisting that bypasses spam-filtering for messages containing ".ite" in the "mail from" and "recipient to" attributes, respectively.

```
type=envelope; command=[mail from]; value=[@.*ite\b];
type=envelope; command=[rcpt to]; value=[@.*ite\b];
```

Note that email address strings contain a leading less than sign (<) and trailing greater than sign (>). Either of the following examples can be used when searching for an explicit email address (for both header and envelope whitelisting):

```
type=envelope; command=[rcpt to]; value=[<user@domain\.com>$];
type=envelope; command=[rcpt to]; value=[\buser@domain\.com$];
```

Suppose that you want to configure a more complex rule to allow mail from the following domain:

```
user@domain.com
user@primary.domain.com
user@secondary.domain.com
```

but explicitly exclude:

```
user@machine.primary.domain.com.
```

A suitable rule could look like:

```
type=envelope; command=[rcpt to]; value=[@([^.]*\.domain|domain)\.com\b];
```

Sample whitelist configuration file

```
## Whitelist Configuration File

#
# This configuration file defines whitelisted
# domains, IPs and headers. When an item is
# matched, the message is guaranteed to be let
# through, unmodified.

# Empty lines or lines where the first non-whitespace
character is a ``#'' are ignored.

# Type: Host
# This type of whitelist entry applies to any kind
# of ip or domain name. If a DNS name is provided,
# then whitelist effectiveness is contingent on DNS
# being properly enabled and set up on the system
# this product is installed on. IP subclasses and
# DNS subdomains are supported.

#type = host; address = [1.2.3.4];
#type = host; address = [192.168.];
#type = host; address = [mx1.somecompany.com];
#type = host; address = [.gov];

# Type: Header
# This type of whitelist entry effectively matches any
substring or regular expression against the specified header
field.

#type = header; header = [From]; value = [@.*gov>];
#type = header; header = [From]; value = [@cloudmark.com];

# Type: Envelope
# This type of whitelist entry applies to commands
# on the SMTP envelope. The value will match
# against any substring or regular expression in
# the specified command field of the envelope.

#type = envelope; command = [helo]; value = [.*ite];
#type = envelope; command = [mail from]; value = [@.*ite];
#type = envelope; command = [rcpt to]; value = [@.*ite];

# Type: body
# This type of whitelist only applies to the body
# of the message. The value of those entries can be
# any kind of regex. Be careful as to what is
# entered here since the most complex regexes will
```

```
# affect performances.
# type = body; regex = [CLOUD.ARK];
```

CHAPTER 6

Cartridge Statistics Reporting

The Cloudmark Cartridge collects statistics about message classification and reports these statistics to Cloudmark. These statistics are used in conjunction with feedback data collected from the Cloudmark Network Feedback System (CNFS) to provide customers and Cloudmark with visibility into filtering accuracy at customer sites.

! *As of Cartridge 3047, statistics reporting is mandatory. If your organization has special privacy concerns, contact Cloudmark.*

This chapter includes the following topics:

How statistics are reported to Cloudmark

Statistics are always collected at each Cartridge installation. Collected statistics are not written to disk and they are not accessible at an installation. By default, statistics are reported to Cloudmark at the following URL:

```
http://lvc.cloudmark.com/cmstats
```

The default hostname is microupdates.cloudmark.com. If the Cartridge is configured to use an HTTP proxy for micro-update downloads, statistics will be sent using the same HTTP proxy. See "Using HTTP proxies" on page 19.

The POST body will be of content type text/plain and contain a collection of key-value pairs. Below is a list of the key-value pairs in the POST body:

Table 2 *POST body key-value pairs*

Key-value pair	Description
report = spamdna stats	This identifies that this report consists of Cartridge statistics. It may be different for future possible communications from the Cartridge to Cloudmark.
version = <val>	This identifies the version of the reported statistics. When Cloudmark adds, removes, or changes the statistics reported by the Cartridge, Cloudmark will increment this version number.
encoding = [e|c,e]	This identifies the encoding of the statistics block. • "e" implies that the block has been encrypted and then base64-encoded. (This was the default for Cartridge 3044 and 3045.) • "e,c" implies that the block has first been compressed using the compress() function before being encrypted. (This was the default as of Cartridge 3046.)
stats = <compressed, encrypted, encoded blob>	All remaining statistics are compressed and encrypted using a symmetric key embedded into the Cartridge. The blob is compressed using the zlib compress() function, encrypted using AES, and base64 encoded. The raw (uncompressed, unencrypted) data consists of key = value pairs, separated by newlines.

By default, the Cartridge communicates statistics back to Cloudmark every hour. The statistic reporting interval is managed by Cloudmark; Cloudmark may adjust the frequency of statistic reporting interval when assisting customers with accuracy issues.

What statistics are collected

Information regarding the specific Cartridge instance is reported to Cloudmark upon each Cartridge installation. The following is a list of statistics that are collected by the Cartridge if reporting is enabled.

Table 3 *Statistics key-value pairs*

Key-value pair	Description
customer id = <string>	The exact value of "customer id" in cartridge.cfg. It is possible that this value may be blank for customers who enable statistics but do not specify a customer id.
host id = <string>	The fully-qualified domain name of the local machine (or a sha-1 of the domain, depending on the value of "anonymize statistics").
report serial = <positive integer>	Identifies a specific report from a given Cartridge instance. This is incremented for every report, whether or not it is sent successfully.
spamdna version = <string>	Identifies the Cartridge version of the instance making the report.
cmae version = <string>	This identifies the Cloudmark Authority Engine version using this Cartridge. It may be "unknown", meaning that the Cartridge is being used by a non-CMAE application or CMAE < 2.0.

Table 3 *Statistics key-value pairs*

Key-value pair	Description
application name = <string>	This identifies the application using this Cartridge. It may be "unknown", meaning that the Cartridge is being used by a non-CMAE application or CMAE < 2.0. It may also be "unspecified", meaning that a CMAE 2.0 application chose to not provide an application name.
log messages = <base64 encoded block>	If present, this contains any log messages with severity >= INFO in the last reporting period. These log messages are separated by newlines and base64-encoded.
message sizes = <size1> <size2> ... <sizeN>	This shows the message sizes, in kilobytes, used by any statistics with names "... sizes". These sizes are used to break down statistics by size into buckets [<size1>:<size2>), [<size2>:<size3>), ... [<sizeN>:infinity). This statistic may not be present for initial statistics reports (that is, when the report serial number is 1).
scored 0 = <count> scored 1 = <count> ... scored 100 = <count>	Counts of the number of messages in the last data period with scores from 0 to 100. To save space, non-zero counts may not be included in a stat report.
scored 0 sizes = <count> <count> ... scored 1 sizes = <count> <count> ... ... scored 100 sizes = <count> <count> ...	These count the sizes of messages scored with scores 0 through 100 in the last data period. If all counts are zero, the entry will not be included in a statistics report. However, if any size count for a given score is non-zero, all sizes will be included.

Table 3 *Statistics key-value pairs*

Key-value pair	Description
identified e4 = <count> identified e7 = <count> identified e8 = <count> identified e9 = <count> identified e10 = <count> identified e14 = <count> identified e15 = <count> identified e16 = <count> identified e17 = <count> identified e18 = <count> identified as empty = <count>	Count of messages identified by specific fingerprint algorithms in the last data period.
identified e4 sizes = <count> <count> ... identified e7 sizes = <count> <count> ... identified e8 sizes = <count> <count> ... identified e9 sizes = <count> <count> ... identified e10 sizes = <count> <count> ... identified e14 sizes = <count> <count> ... identified e15 sizes = <count> <count> ... identified e16 sizes = <count> <count> ... identified e17 sizes = <count> <count> ... identified e18 sizes = <count> <count> ...	These count the sizes of messages identified by various engines in the last data period. If all counts are zero, the entry will not be included in a report. However, if any size count for a given whitelist type is non-zero, all sizes will be included.
local header whitelist = <count> local envelope whitelist = <count> local body whitelist = <count> local ip whitelist = <count> awl header whitelist = <count> awl envelope whitelist = <count> awl body whitelist = <count> awl ip whitelist = <count>	Count of messages whitelisted by local whitelist.cfg and awl whitelist files.
local header whitelist sizes = <count> <count> ... local envelope whitelist sizes = <count> <count> ... local body whitelist sizes = <count> <count> ... local ip whitelist sizes = <count> <count> ... awl header whitelist sizes = <count> <count> ... awl envelope whitelist sizes = <count> <count> ... awl body whitelist sizes = <count> <count> ... awl ip whitelist sizes = <count> <count> ...	These count the sizes of messages whitelisted by various methods in the last data period. If all counts are zero, the entry will not be included in a statistics report. However, if any size count for a given whitelist type is non-zero, all sizes will be included.
(local|awl) (header|host|env|body) entry 1 = <count> local|awl) (header|host|env|body) entry 2 = <count> ...	These count how many times each whitelist.cfg/awl entry whitelisted a message. Only entries with counts > 0 will be included.

Table 3 *Statistics key-value pairs*

Key-value pair	Description
category <cat_number> = <count> subcategory <cat_number>.<subcat_number> = <count>	<cat_number> is the category number, <subcat_number> is the subcategory number. Key-value pairs are only sent when <count> > 0. Since subcategories are not always used, there may be more messages classified as a particular category than messages classified in that category's subcategories.
whitelist = <base64 encoded blob>	This holds the exact contents of whitelist.cfg as read by the Cartridge. It is base64-encoded as the file may contain multiple lines, blank lines, malformed content, and so on.
successful complete micro-update downloads = <count> successful incremental micro-update downloads = <count> failed complete micro-update downloads = <count> failed incremental micro-update downloads = <count> successful complete micro-update downloads from disk = <count> failed complete micro-update downloads from disk = <count> failed micro-update serial number downloads from network = <count>	Count of the number of successful and failed micro-update downloads. Note that a micro-update download includes rpl, dpl, awl, and other files in addition to fingerprints.
mu <source> serial = <complete serial>.<incremental serial>	Reports the latest serial number for all micro-update data files. (i.e. fingerprints, rpl, dpl, and so on.)
sigs seen = <sig>:<engine>:<count> <sig>:<engine>:<count> ...	If present, this key lists th signatures that have been seen by the Cartridge in the last reporting period. • <sig>:<engine> names the signature. • <count> names the number of times the signature was seen. This key will only be present for installations that have enabled last seen checking via a hidden cartridge.cfg option.

Table 3 *Statistics key-value pairs*

Key-value pair	Description
license id = <integer >0>	The license number as found in license.cfg. If the license is not found, the value of this key is zero.
clean shutdown = [01]	This key indicates whether the cartridge shut down properly during its previous run; 1 if it did, 0 if it did not.
installation id = <string>	This is a base64-encoded string that uniquely identifies a cartridge installation.

Cartridge reporting configuration

The following configuration items configure the Cartridge's reporting mechanism:

- report statistics

  By default, this is set to "yes" to enable reporting. If your organization has special policies that prohibit the type of reporting described in this chapter, set this parameter to "no". However, this is not recommended as it may complicate Cloudmark's technical support process in the event of a problem.

- exclude from stats reports

  Use this configuration key to exclude whitelist and proxy authentication statistics from the report.

- customer id

  This sets the name of your organization, or your organization's CNFS login, if any.

See also Chapter 3, "Cloudmark Cartridge Configuration".

APPENDIX A

Logging

Log messages are passed programmatically from the cartridge to your application. See your application's documentation for information about how these messages are exposed.

This appendix explains the logging variables and the log messages in which they are found:

Common variables

The following common variables are used in the INFO log messages:

Table 4 *Log variables*

Variable	Sample Values	Description
<cachedir>	N/A	The directory specified by "micro-update cache dir", possibly prepended with <etcdir>
<count>	N/A	A message/entry/statistic count
<customer id>	N/A	The customer id specified in cartridge.cfg
<decode_error>	"Could not decrypt data", "Could not decompress data", "Truncated content"	An error string related to decoding content

Table 4 *Log variables*

Variable	Sample Values	Description
<disk_error>	"Permission Denied", "Not a Directory", "No Such File or Directory"	Any of the standard error strings returned on Unix or Windows systems related to disk I/O
<encrypt_errnum>	N/A	An error code related to encryption or decryption
<etcdir>	N/A	Directory containing authority.cfg, immunity.cfg or the ConfigDirectory passed to CMAE_Init()
<host id>	N/A	The local machine's hostname, as returned by gethostbyname()
<http_request_str>	N/A	The contents of a HTTP request to the configured micro-updates host or proxy, with unprintable characters converted to '.'
<http_response_str>	N/A	The contents of a HTTP response from the configured micro-updates
<mu_file>	N/A	A file in the micro_updates directory
<mu_source>	N/A	Contains one of the following values: "signatures", "signature meta-data-rpl", "signature meta-data-dpl", or "awl"
<net_error>	"Connection Refused"	
"Timed Out"	Any of the standard error strings returned on Unix or Windows systems related to TCP/IP networking	
<serial>	N/A	The serial number of a micro-update file in the form of <complete>.<incremental>
<sig>	N/A	An individual message fingerprint
<whitelist_module>	N/A	Indicating the specific whitelisting module

INFO log messages

Will load source %s from safe set (%s is not safe) A suspect file was found; an older file will be loaded instead.

Copied package microupdate files to "safe" set Displayed during first cartridge initialization after installation when the package files are considered safe.

Committed key "report statistics" from "License Cfg" value=yes A configuration parameter was set from the LDAC.

Licensing OK (session #<session number>)" A new session was retrieved from Cloudmark's licensing host.

(<whitelist_module>) Backend whitelist enabled with <count> (entry/entries) This message is logged when a .awl micro-update file is downloaded and loaded that contains any entries of the specified type.

(<whitelist_module>) enabled with <count> (entry/entries>) This message is logged when whitelist.cfg is loaded and contains any entries of the specified type.

Will (not) consider empty messages as spam This message is logged at initialization and reflects the option "consider empty messages spam" in cartridge.cfg.

Micro-updates enabled, updating all micro-update files from network and/or disk before continuing... This message is logged during initialization when micro-updates are enabled. The cartridge will attempt to load the latest micro-update files from the network and from disk before initialization completes.

Micro-update configuration... This message is logged when the micro-update module is initialized, and gives a summary of micro-updates related configuration

Attempting to create micro-update directory <cachedir>. This message is logged at cartridge initialization when the configured cache directory does not exist. The cartridge will attempt to create the directory

Successfully updated <mu_source> from disk (new serial <serial>).

This message is logged when the cartridge loads a new micro-update file from disk; it does this at startup or when it finds a file in the micro_updates directory whose serial number is greater than the serial number of the in-memory version of the source

Removed partial <mu_source> file <file> This message is logged when the micro-update module finds and removes partial micro-update files with the

extension .part in the micro-updates directory. These files may exist if the cartridge is shut down while a micro-update download is in progress

Successful <mu_source> download from network (new serial <serial>). This message is logged when the cartridge successfully downloads and decodes a micro-update file from the network

Removed old <mu_source> file <filename>. This message is logged when the cartridge successfully removes an old file (i.e. the one with the lowest serial number) below the micro_updates directory. The cartridge will keep at most 2 micro-update files for a source at any one time

The option "micro-update-path" has been deprecated, ignoring

This message occurs when a deprecated configuration option is encountered.

Successful <mu_source> incremental download from network (new serial <serial>) This message is logged when the cartridge successfully downloads and decodes a micro-update file from the network

Shutting down, stopping micro-update downloads. This message is logged at shutdown to indicate that micro-updates will no longer be downloaded and saved

The option "micro-update-filename" has been deprecated, ignoring "These messages are logged at startup when either of these deprecated options are present in cartridge.cfg

Could not connect to host <host>:80 (addr <addr>)--<net_error>.

This message is logged when the cartridge attempts to connect to the micro-updates host on port 80 but fails. If an HTTP proxy is configured, the cartridge only makes a single connection attempt to the configured HTTP proxy.

Network feedback whitelist hit for <email> This message is logged when Cloudmark's network feedback system has whitelisted a message for a specific email address. These messages should only occur if your installation is submitting feedback through Cloudmark's Network Feedback System and this message is similar to false positive feedback received for that specific address.

Current statistics for period <report_id>... These messages comprise a statistics report, and are logged at the end of every reporting period

Shutting down, stopping background stats reporting. This message is logged at shutdown

Statistics initialized. Current settings... This message is logged when the statistics module is initialized, and gives a summary of statistics related configuration

Connecting to licensing host without authentication Cloudmark's licensing host has served a checksum file unencrypted.

WARN log messages

Can not create directory <cachedir> (<disk_error>). Expect errors saving micro-update files. This warning is logged when the pathname specified by "micro-update cache path" does not exist; the cartridge attempts to create the directory and fails

Can not create partial micro-update file <mu_file> (<disk_error>)

This warning is logged when the micro-update module cannot open a file to hold downloaded micro-update content. Micro-update content will still be loaded into memory but will not be saved on disk.

Can not write to micro-update file <mu_file> (<disk_error>) This warning is logged with the micro-update module cannot write downloaded micro-update content to disk. Content will still be loaded into memory.

Can not rename partial micro-update file <mu_file>.part to <mu_file> (<disk_error>) The micro-update module initially creates micro-update files with the extension .part and renames them once all content has been downloaded and validated. This warning occurs if the module cannot rename a complete file with the extension part. Downloaded content is still loaded into memory.

Can not remove partial micro-update file <mu_file>.part (<disk_error>) If a micro-update download is terminated before it is complete or if there is a problem validating the downloaded content, the micro-update module attempts to remove the partial file <mu_file> part. This warning is logged if the partial file cannot be removed.

Have not updated <mu_source> yet--accuracy will be affected until content is successfully downloaded from the network. This message is logged at cartridge initialization when no suitable files are found below the micro_updates directory. As these files are all extremely important in achieving

high accuracy, it is important to wait until files are successfully downloaded from the network before passing messages through the cartridge.

Can not read information about micro-update directory <cachedir> (<disk_error>). This warning is logged when the cartridge is unable to read any information about the configured micro-updates directory. This message will not be logged if the directory does not exist. In that case, it will attempt to create the directory instead.

Micro-update directory <cachedir> must be a writable directory.

This warning is logged when the configured micro-update cache directory exists, but is not a writable directory.

Could not load <mu_source> files from disk (<disk_error|decode_error>). This message is logged when the cartridge attempts to read cached micro-update files from disk but fails. This message will be logged only if the cartridge is configured to perform micro-update downloads, as it can fall back to network-based downloads. If the cartridge is not configured to perform micro-update downloads, it will log an error message instead of a warning message.

Could not load <mu_source> files from disk (<disk_error|decode_error>). This message is logged when the cartridge attempts to read cached micro-update files from disk but fails. This message will be logged only if the cartridge is configured to perform micro-update downloads, as it can fall back to network-based downloads. If the cartridge is not configured to perform micro-update downloads, it will log one of the below errors instead.

Can not remove old <mu_source> files (<disk_error>). This message is logged when the cartridge attempts to remove an old file (i.e. the one with the lowest serial number) below the micro_updates directory but fails. The cartridge will keep at most 2 micro-updates files for a source at any one time.

Can not resolve [micro-update, http proxy] hostname <host> (<net_error>), expect micro-update downloads to fail. This message is logged at startup when the cartridge cannot resolve the configured micro-update host to an IP address. This typically indicates a problem with DNS resolution at the installation. Cartridge accuracy will be significantly reduced without current micro-update files, so the problem should be identified and solved as quickly as possible.

Could not remove signature <sig> from local set This message is logged when a downloaded micro-update file contains a signature that is already present in the in-memory set of signatures. This warning can occur in rare

instances, but a large number of warnings indicate a problem with the in-memory set of signatures.

Could not update signature <sig> from local set ([add/remove] portion) Incremental micro-update files may contain updates to meta-data associated with a signature, and this log message indicates a problem updating signature meta-data.

Unable to parse mime parts of message, skipping This message is logged when the cartridge is unable to parse the mime parts present in a message. In this case the network classifier will not compute fingerprints for the message and the message will be scored with a score of zero.

Statistics reporting is enabled, but "customer id" is set to "none". Set it to an address of the form "<companyname>@feedback.cloudmark.com" This message is logged when statistics reporting is enabled but "customer id" is still set to its default value of "none".

Invalid configuration directory This message is logged when the cartridge is configured with an invalid directory containing whitelist.cfg. In this case no local whitelist entries will be used.

Could not [find | open] config file <file> (<disk_error>), local whitelist will be disabled This message is logged when the cartridge cannot open or read the configured whitelist.cfg file. In this case no local whitelist entries will be used.

Malformed line '<line>', <parse_error>, skipping This message is logged when the whitelist module could not parse a line in whitelist.cfg. Such lines will be ignored. Examples include "unknown type", "missing command or regex", "malformed regex", "missing address", etc.

Could not save engine information '<string>' --> '<string>'. When used with CMAE, the cartridge is initialized with engine information such as the CMAE version string, application name, and so on. This information is saved within the cartridge and included in statistics reports. This message is logged when the cartridge could not properly save that information.

ERROR log messages

Failed to read current serials from disk The etc/micro_updates/states/srl_set.current file is unexpectedly missing.

Could not find source %s in current serial state file The srl_set.current file is corrupted.

Failed to write the current state file (%s) The disk may be full or there may be a permission error.

Failed to remove %s There was a permission error in the etc/micro_updates/states/ directory.

Failed to rename %s to %s There was a permission error in the etc/micro_updates/states/ directory.

Failed to open %s for writing The disk may be full or there may be a permission error.

Could not read from %s The disk may be full or there may be a permission error.

Malformed line in %s The state file is malformed.

Could not update signatures from network (malformed license id in etc/license.cfg). The license ID included non-numeric characters.

Could not update signatures from network (missing license file [etc/license.cfg]). The license file is missing.

Could not update signatures from network (invalid license [HTTP error: 401]). The license credentials are invalid; the ID and password do not match or the ID does not exist.

Could not update signatures from network (unexpected download failure - http_code:500). A micro-update file is missing.

Could not update signatures from network (license verification failed tid:38695 (License not enabled or expired)). The license is disabled in the LDAC.

Could not update signatures from network (license verification failed tid:38697 (License not enabled or expired)). The license has expired.

Could not update signatures from network (authorization denied).

The license is not authorized to download micro-updates.

Could not update signatures from network (license verification failed tid:38977 (No sessions available)). The number of sessions authorized for this license has been reached.

Could not update signatures from network (microupdate file XXXX failed the integrity check). The computed checksum is incorrect.

Failed to load checksum file from cache (<filename>) The Cartridge failed to load a checksum file from disk.

Unable to load symbol <sym> This message is logged when there is a problem loading the cartridge adapter. It is likely that there will be problems with subsequent message classification.

Could not create message score. Expect errors related to reading message scores This message is logged when the cartridge cannot initialize a new message score. If this message is logged the cartridge will return a score of zero for this message. This problem should be identified and fixed immediately to achieve high accuracy.

Could not free per-thread storage for message scores This message is logged at shutdown when the cartridge cannot free per-thread storage used for message scores. This will not affect accuracy but implies a resource leak.

Could not update <mu_source> from [network or] disk (<disk_error|net_error|decode_error>) This message is logged when the cartridge attempts to update a source from the network or disk but both methods fail. Cartridge accuracy will be significantly reduced without current micro-update files, so the problem should be identified and solved as quickly as possible.

Invalid micro-update interval <interval>, using default interval of 3 hours. This message is logged at startup when the value for "micro-update interval" in cartridge.cfg is not "auto" or a positive integer.

Invalid cache path <path>, using default value ".". This message is logged at startup when the value for "micro-update cache path" in cartridge.cfg is empty. If the path is specified but is not a directory or cannot be created, a different message will be logged.

Value <val> for "enable micro-updates" must be yes/no/1/0--keeping downloads enabled. This message is logged at startup when the value for "enable micro-updates" in cartridge.cfg is not a yes/no-style value.

Can not start networking subsystem (<net_error>), expect network operations to fail This error is logged on Windows platforms when the cartridge could not initialize networking via WSAStartup().

Invalid http proxy <proxy> (<error>), not using a http proxy.

Examples include "must be of the form host:port", "port must be in the range 1-65536", etc. This message is logged at startup when the value for "http proxy" in cartridge.cfg is invalid.

Invalid http basic auth <auth> (<error>), not using authentication for http proxy A string describing why the authentication specification is invalid.

Invalid micro-update timeout <timeout>, using default timeout of 60 seconds. This message is logged at startup when the value for "micro-update timeout" in cartridge.cfg is not a positive integer.

Could not shut down networking subsystem (<net_error>) This message is logged on Windows platforms when the cartridge could not shut down networking via WSACleanup().

Can not start thread for background loader. Will only use local micro-update files. This message is logged at startup if the background task that periodically updates micro-update files from the network and disk could not be started. Cartridge accuracy will be significantly reduced without current micro-update files, so the problem should be identified and solved as quickly as possible.

Could not encrypt statistics report (error <encrypt_errnum>), will not send report This message is logged when there is a problem encrypting a statistics report. If this message occurs, the statistics report will not be sent to Cloudmark.

Could not encode encrypted statistics report, will not send report

This message is logged when there is a problem encoding the encrypted contents of a statistics report. If this message occurs, the statistics report will not be sent to Cloudmark.

Could not construct HTTP POST for statistics report, will not send report This message is logged when there is a problem constructing the HTTP POST command used to submit a statistics report. If this message occurs, the statistics report will not be sent to Cloudmark.

Could not connect to micro-updates host (<net_error>), will not send report This message is logged when there is a problem connecting to the configured micro-updates host (potentially via a proxy if a proxy is configured). If this message occurs the statistics report will not be sent to Cloudmark.

Could not send HTTP request <http_request_str> (<net_error>), will not send report This message is logged when there is a problem sending a HTTP POST containing a statistics report to the configured micro-updates host or proxy. If this message occurs the statistics report will not be sent to Cloudmark.

Could not close socket for statistics report (<net_error>), report may not have been sent successfully This message is logged when there is a problem closing a network socket after sending a statistics report. If this message occurs, the report may not have been successfully sent to Cloudmark.

Could not receive HTTP response for statistics report (<net_error), report may not have been sent successfully This message is logged when there is a problem reading the HTTP response after submitting a statistics report. If this message occurs, the report may not have been successfully sent to Cloudmark.

Could not determine hostname (<net_error>), using a host identified of "unknown "This message is logged when the cartridge could not determine the local machine's hostname. The string "unknown" will be used in statistics reports.

Unexpected HTTP response for HTTP request <http_request_str>: <http_response_str> This message is logged when the HTTP response code from a statistics report is something other than 200 (i.e. OK). If this message occurs, the report was not accepted by Cloudmark.

Could not start background statistics thread, will not log or report statistics This message is logged when there is a problem starting the thread that periodically logs statistics and sends statistics reports. If this thread does not start, collected statistics will not be logged or sent to Cloudmark.

Could not initialize encryption, expect problems reporting statistics This message is logged when there is a problem initializing the module used to encrypt statistics reports. You can expect subsequent failures with sending statistics reports.

CRITICAL log messages

(<module>) Could not initialize micro-update module This message is logged when there is a problem initializing the module used to interact with micro-updates servers. Without this module, micro-update downloads and statistics reporting will be non-functional, both of which are extremely important in achieving high accuracy.

Could not initialize per-thread storage for message scores. Will not be able to return meaningful scores for messages Message scores are maintained in the cartridge in per-thread storage, and this message is logged if the cartridge could not initialize that storage. If this message is logged it means that the cartridge will return a score of zero for all subsequent messages. This problem should be identified and fixed immediately to achieve high accuracy.

Index

T

U

W