CYREN

LocalView Rules Management Guide

(For: ctengine Version 8.0 and
ctasd Version 5.0)

ctengine/ctasd Module and Documentation Usage Restrictions

This program and the accompanied documentation is SECRET AND CONFIDENTIAL, and constitute a proprietary trade secret of CYREN Inc. (herein after referred to as "CYREN").

No person is allowed to copy, decompile, reverse engineer, use, sublicense or otherwise access this program unless the prior express, written consent is received from CYREN. The possession and use of this program shall be governed by the terms of a license agreement between CYREN and each authorized licensee. Unauthorized use of this program is strictly prohibited, and those perpetrating such unauthorized uses shall be prosecuted to the fullest extent of the law. The confidentiality and non-disclosure obligations of licensee shall be strictly maintained at all times by licensee and licensee, in receiving a copy of this program, acknowledges that it shall not be disclosed to third parties; rather, only to employees or consultants having a firm need to know, and provided that they are bound by confidentiality restrictions at least as restrictive as those adopted by licensee within the framework of its relationship with CYREN.

The failure to maintain confidentiality will likely cause severe damages and irreparable harm to CYREN and, therefore, in addition to any other remedies and rights available at law, CYREN shall be entitled to seek injunctive relief without the need for the posting of any bond or other guarantee.

Trademark and Copyright Statement

CTT20-800-111-128-R1

ctengine is a licensed SDK product featuring patented technology. CYREN's patented solution is protected by U.S. patent #6,330,590.

RPD, Zero-Hour Virus Protection, VOD and ctengine are trademarks of CYREN Inc. For more information, visit our website: http://www.cyren.com/ Microsoft is a trademark and/or registered trademark of Microsoft Corp. Linux is a trademark of Linus Torvalds. Red Hat is a trademark of Red Hat, Inc. in the United States and other countries. Debian is a registered trademark of Software in the Public Interest, Inc. FreeBSD is a registered trademark of Wind River Systems, Inc. Slackware® is a registered trademark of Slackware Linux, Inc. Sun and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All SPARC trademarks are trademarks or registered trademarks of SPARC International, Inc. in the United States and other countries. COPYRIGHT AND PERMISSION NOTICE Copyright (c) 1996 - 2014, Daniel Stenberg, <daniel@haxx.se>. All rights reserved. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All other trademarks and registered trademarks are the property of their respective owners.

Contacts

Any technical questions you or your developers have about using the ctwsd should be addressed to support@cyren.com.

About CYREN

CYREN™ provides proven Internet security technology to more than 150 security companies and service providers including 1&1, Check Point, F-Secure, Google, Microsoft, Panda Security, Rackspace, US Internet, and WatchGuard, for integration into their solutions. CYREN's GlobalView™ and patented Recurrent Pattern Detection™ (RPD™) technologies are founded on a unique cloud-based approach, and protect effectively in all languages and formats. CYREN Antivirus utilizes a multi-layered approach to provide award winning malware detection and industry-leading performance.

CYREN technology automatically analyzes billions of Internet transactions in real-time in its global data centers to identify new threats as they are initiated, enabling our partners to protect end-users from spam and malware, and ensure safe, compliant browsing. The company's expertise in building efficient, mass-scale security services mitigate Internet threats for thousands of organizations and hundreds of millions of users in 190 countries.

CYREN, formerly known as CYREN, was founded in 1991, is headquartered in the US in McLean, Virginia, with offices in Palo Alto, California, Herzliya, Israel, Berlin, Germany, and Reykjavik, Iceland.

For more information about enhancing security offerings with CYREN technology, visit our website at www.cyren.com, see our blog at http://blog.cyren.com or write to info@cyren.com.

Table of Contents

1 Introduction

CYREN is constantly looking for ways to maintain and improve its competitive edge in providing the highest detection rates to identify spam and other email-borne threats. CYREN's LocalView Engine is an additional engine supporting CYREN's ongoing efforts to provide solutions for identifying and neutralizing email-borne malware.

The LocalView engine was designed to enhance CYREN's patented Recurrent Pattern Detection (RPD) technology, implemented through its RPD detection engine, which is the base of most of CYREN's product offerings. The LocalView engine enhances the already high detection rate delivered by the RPD engine by adding rule-based detection via either CYREN's ctengine or its Anti-Spam daemon (ctasd).

For those customers who have already adopted a dual-engine, anti-spam strategy, CYREN's LocalView offers an ideal solution for incorporating a rule-based engine similar to SpamAssassin. Like SpamAssassin, an open source project by the Apache Software Foundation, LocalView works using similar terminology and rule-based scoring. The LocalView Engine eases migration of existing rules and minimizes the learning curve for those developers who are familiar with SpamAssassin, while adding rule-based spam detection to CYREN's pattern-based technology and unparalleled detection rates.

This document details the types of rules that can be configured for the LocalView engine and how CYREN customers (such as OEM partners and Service Providers) can create and manage custom rules. For information on integrating LocalView with ctengine, refer to the *ctengine Developer's Guide*; for information on integrating LocalView with ctasd, refer to the *ctasd Integration Manual*.

1.1 Rule Groups

When provisioned in the license key, the LocalView engine leverages CYREN's cloud-based architecture to distribute its rules to all clients. There are two groups of rules that can be applied to incoming or outgoing messages:

- **LocalView CYREN Rules (CT Rules)**– these are rules that are defined and maintained by CYREN and updated on a regular basis.
- **LocalView Custom Rules** – These rules are defined and maintained by CYREN customers. As part of these LocalView Custom Rules, Blacklists and Whitelists can be maintained by the customer and applied to messages. The ability to define and implement these rules enables customers to customize CYREN's email security services to the their particular needs, industry, location, etc.

1.1.1 LocalView CYREN Rules

Although the final determination of which rules to run and which score to associate to each rule is determined by the customer, it is recommended that the customer adopt the default CYREN Rules. The reasoning for this is twofold:

1. CYREN invests tremendous effort to select, define, and test the most effective rules that can enhance its RPD detection engine and further improve CYREN's already very high detection rates. These rules are implemented via the LocalView engine and are made available to all customers.
2. The resources required to run the CT rules are much lower than the resources required to run the Custom rules.

By default, CYREN rules (CT rules) and their updates are distributed to all customers.

Note: *Though not recommended, if a customer does not wish to have the CYREN Rules distributed, contact CYREN's Support Team.*

1.1.2 LocalView Custom Rules

LocalView supports two sub-types of custom rules:

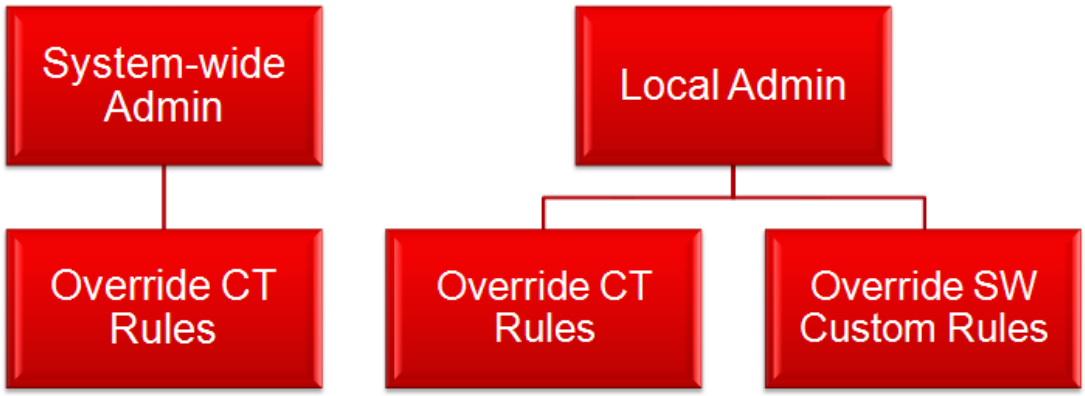
- **System-wide Custom Rules (SW Custom Rules)**: These custom rules are managed and distributed centrally per all clients (for all instances of ctengine or ctasd daemons) for a given customer's license key. In order to distribute the SW Custom Rules, these rules must first be uploaded to the CYREN Datacenter. SW Custom Rules are managed by a system-wide partner or Service Provider Administrator.
- **Local Custom Rules (LOC Custom Rules)**: These custom rules are managed locally per each client or client group. It is the responsibility of the customer to manage the distribution of these custom rules to its target clients. LOC Custom rules are managed by a Local System Administrator.

1.2 Rule Hierarchy

Within the implementation of rules, there is a built-in hierarchy that allows administrators to set which rules can override others in cases where one rule conflicts with another or defines different scoring values.

This hierarchy includes:

- A Local System Administrator can override both CT Rules and SW Custom Rule definitions.
- A System-Wide Administrator can override the CT Rules definitions.



1.3 Black List and White List Rules

CYREN uses its ability to offer custom rules as a way of integrating a customer's Black and White lists. Whitelisted messages are matched to CYREN's NonSpam classification, while Blacklisted messages receive a classification of ConfirmedSpam. System administrators can access and update the white and black lists by adding, removing or modifying black and white entries as needed.

Note: *A local IP Ignore list should be populated with all IP addresses of the organization's trusted networks. Where an entry from the IP Ignore list is found in either the Whitelist or the Blacklist, the Whitelist/Blacklist entry is ignored.*

1.4 Scoring Values

CYREN's LocalView supports rule scores similar to the rules managed in SpamAssassin such that negative scores indicate a message with "good" email characteristics, while a positive score indicates that it contains "bad" email characteristics. Rules that define the score value as zero (0) will be considered disabled.

In general, it is advised that you assign low scores to your rules so to avoid creating false positives – messages that are incorrectly classified as spam when, in fact, they are not.

1.5 Thresholds

A threshold defines the message score at which a classification should be given. Typically, in CYREN's case, this defines the levels at which a message will be classified as either Bulk or Spam. The system is defined with the following default thresholds:

- LocalView_BulkThreshold=5
- LocalView_ConfirmedThreshold=10

1.6 LocalView Short-Circuit Option

The LocalView engine provides an option for shortening the classification process by "short-circuiting" when a threshold is met. If enabled (in the connection string for ctengine or the ctasd.conf for ctasd), the message scanning process will be stopped as soon as the threshold is reached. By default, this option is disabled (set to false).

1.7 Viewing Active Loaded Rules

Loaded rules are those that are currently being used by ctengine or ctasd. In the case of CT Rules and SW Custom Rules, they are the rules contained in files currently loaded to the CYREN Datacenter and distributed to all clients. For LOC Custom Rules, they are the current rules stored locally and accessed by ctengine or ctasd. Whenever an update is performed locally on any of the custom rule files located in CustomRulesFilePath, the new rule files are immediately loaded to ctengine's cache.

Note: *The System-Wide LocalView Custom Rules must be defined in a single file with the name SWCustomRules.txt.*

In addition, whenever an update for a CT Rule or SW Custom Rule file is performed, the client is refreshed and updated with the rule updates.

In order to view what is currently loaded to the ctengine's cache the following APIs can be used:

- **GetRulesList:** will display the rule tags of all loaded rules (including CT rules, SW Custom rules, LOC Custom rules, Blacklist and Whitelist rules).
- **GetCTRuleDefinition**: will display the definition of the specified CYREN rule

- **GetCTRulesDefinitionList:** will display for loaded CT rules full rule definitions.

Note: *Refer to the ctengine Developers guide or ctasd Integration Guide for a detailed function call/API description.*

2 Implementing Custom Rules

CYREN customers can define additional rules for any purpose. For example, a customer may choose to define a rule that addresses local attacks or add additional criteria to enhance the CYREN Rules for their specific environment. These rules can be defined to support Fixed, Regex, and Compound Rules.

In order to support Custom Rules, the CustomRulesFilePath parameter must be defined in the ctengine connection string or the ctasd configuration file. Only custom rule files should be contained in this folder. You cannot use '.' path or ./filename.txt.

Note: *The System-Wide LocalView Custom Rules must be defined in a single file with the name SWCustomRules.txt.*

2.1 Custom Rule Components

LocalView Custom rules are defined and managed by CYREN customers. For each rule, the following information may be required (see below for an explanation of which attributes are mandatory):

Attribute	Status	Description
Rule tag	Mandatory	The name/tag of the rule (for example: FORGED_MUA_OUTLOOK). The rule tag is used as a unique identifier. If you already have a SpamAssassin-based rule defined, you can use the same name, but you may need to modify some of the syntax. See *Message Classification,* Scores and for more information. ***Note:*** *The Rule tag name is case-sensitive*
Rule Type	Mandatory	Defines the part of the mail that is scanned for the rule. For a list of rule types, See *Custom Rule Types*. ***Note:*** *If you wish to add an optional description, this can be done by adding a # character followed by the description text.*
Rule Score	Mandatory	The score of the rule. If the score is defined as 0, the rule is disabled.

Attribute	Status	Description
Rule Expression	Mandatory	The defined expression of the rule. The rule expression can be a fixed string or any Perl Compatible Regular Expression (PCRE).

2.2 Custom Rule Structure

LocalView's custom rule structure is similar to rules composed for SpamAssassin. There are several types of rules, explained in the following sections. The rule structure, as appears below, uses individual lines to define each rule, assign a score. An optional description line can be used to explain the purpose of the rule or store other relevant comments.

Although each line contains the rule tag (name of the rule), the line containing type of rule and rule expression must physically appear before the line defining the rule's score.

The following describes the structure of a custom rule:

Line Content	Example
Rule Type | Rule Tag | Rule Expression	header Scan_header_viagra viagra
score | Rule Tag | Score of the rule	score Scan_header_viagra 0.6

In the above example, a header rule "Scan_header_viagra" is used to scan the header for the word "viagra". If it is found, a score of 0.6 will be applied to the message. When combined with other tests, the message may be confirmed as spam.

2.3 Custom Rule Types

The following section details the types of custom rules that are supported by CYREN's LocalView engine, and examples of when you would use each rule type.

2.3.1 Header

Header rules are used to check messages that contain a specified string or regular expression in the header section of the message. For example, if you know that a spam attack contains a specific string of letters or words, you can create a rule for that string and assign a score that will contribute to the message being identified as spam.

When this rule is run, the header section of the message is tested to determine if there is a threat. Specific header definitions are supported. If you do not define the header value, all message headers will be supported. If a message header value is found to match the header rule, the specified score is assigned to that message.

2.3.2 Body

A LocalView body rule searches the body of a message only to see if it contains a specified expression and if it matches, the specified score is assigned.

Note: *The subject is not considered part of the body content. This behavior differs from SpamAssassin who considers the Subject as the first line of the body content.*

2.3.3 Meta

A Meta custom rule enables you to combine scores (Boolean or arithmetic combinations) of various rules to determine the likelihood that it is spam based on more than one factor. For example, you could combine both a header and a body rule to create a combined rule that may be better at identifying spam.

Example:

```
meta Meta_Rule_Example  (Rule_100 && __Rule_200)
score  Meta_Rule_Example 0.3
```

2.3.4 Uri

URI rules are used to check both the plain text and the HTML sections of a message for specific URIs. For example, if you have information that confirms a URL is one that you want to protect your users from (spam, phishing, etc.), you can specify a rule with the URL of that site, and the Uri rule will search both the plain text as well as text contained within HTML tags.

Example:

```
uri URI_Example   www\.spamsite\.com
score URI_Example 0.1
#
```

The URI scan mode is relevant for body only. The URI string must start with the protocol name, for example http, ftp, etc.

2.3.5 Raw

The LocalView Raw rule enables you to search for specified text without having to first remove HTML tags, line breaks, etc. It searches both the body and the headers for a match and applies the specified score value when found.

2.4 Regular Expression Syntax for Custom Rules

CYREN requires a Perl Compatible Regular Expression (PCRE), while SpamAssassin rules are written in Perl regular expressions.

If you wish to migrate SpamAssassin rules and load then to LocalView, the following syntax formatting differences should be taken into account:

Issue	SpamAssassin Format	CYREN Custom Rule Format
Comparison operator =~	HeaderName =~ Rule expression	When defining a Header based rule, CYREN custom rules do not require a =~ between the header name and the header rule expression.
Regular expression syntax	Regular expression between forward slashes	While in SpamAssassin the regular expression is placed between forward slashes (/), CYREN PCRE-based custom rules do not require this. If the SpamAssassin rule is between forward slashes, these forward slashes should be deleted.
Match definitions	/i, /m, /s, /x	(?i), (?m), (?s), (?x) respectively

2.5 Custom Rules Location

The CustomRulesFilePath parameter in the ctengine connection string or the ctasd.conf file defines the directory where the Custom Rule files should be stored. It is a mandatory parameter, if the LocalView engine has been enabled. It is recommended that you create a separate folder in which to store the custom rule files. Only custom rule files should be contained in this folder. You cannot use '.' path or ./filename.txt.

2.5.1 System-Wide Custom Rules

System-wide Custom Rules must be uploaded to the CYREN Datacenter. Once uploaded, these Custom Rules are automatically distributed to all clients (both those connecting via instances of ctengine as well as those using ctasd daemons) for the specified license key.

Note: *The System-Wide LocalView Custom Rules must be defined in a single file with the name SWCustomRules.txt.*

Contact CYREN Support to receive information on SW Custom Rules file upload URL and credentials.

2.5.2 Local Custom Rules

Local Custom Rules are defined in one or more files. LOC Custom Rules can be locally stored in the location defined in the CustomRulesFilePath parameter (in the connection string for ctengine or the configuration file for ctasd). Only custom rule files should be contained in this folder. You cannot use '.' path or ./filename.txt.

Alternatively, if a local static content server is used to distribute the LOC Custom rules, the URL of this server must be defined as the value of the LocalCustomRulesDistributionURL parameter. The LOC Custom rules are then downloaded and stored locally in the CustomRulesPath location.

Note: *The LocalCustomRulesDistributionURL parameter value must start with http://.*

2.6 Custom Rule Score Updates

A local system admin may decide to update a rule definition or score of a SW custom rule. To override the expression of a SW Custom rule, define in a Custom Rule file a new expression to a tag of a SW custom rule; to override the score of a SW Custom rule, define an updated score to a tag of a SW custom rule.

2.7 Validating Custom Rules

Use the ValidateCustomRules parameter to validate and report errors on LocalView Custom Rules files. The ValidateCustomRules parameter does not load the custom rules, rather it checks to confirm that the defined rule is valid and can be applied by the LocalView engine.

During the validation process, an error message will be generated if any rules are found to be invalid, and a list of which rule expressions could not be validated is included in the error message.

The ValidateCustomRules parameter is used to check all files within a specified directory to validate the syntax using. This can include both System-wide custom rules, Local custom rules.

Note: *This parameter does not check the logic of the content of the files, rather it confirms that the syntax of the file construction is properly presented.*

3 Customizing CYREN CT Rules

CYREN rules are defined and managed by CYREN. While customers cannot modify the rules themselves, customers can customize the associated score for these rules. When modified by the customer, the customer's modified score overrides the CYREN CT Rules score. The CT rule definitions can be viewed in using the GetCTRulesDefinitionList API.

3.1 CT Rule Components

Each CT Rule contains some or all of the following attributes:

Attribute	Status	Description
Tag	Mandatory	A unique name for the rule.
Description	Optional	A description describing the logic of the rule.
Type	Mandatory	Defines the part of the mail that is scanned for the rule.
Score	Mandatory	The score of the rule. If the score is defined as 0, the rule is disabled.
Expression	Optional	The defined expression of the rule. ***Note:*** *Some CT rules may not have an associated rule expression, but rather a written description will be provided.*

If a rule was imported from the SpamAssassin community, then it will maintain the SpamAssassin rule tag name.

3.2 Overriding CT Rules Definitions and Scores

3.2.1 CT Rule Score Updates

Though CYREN recommends that customers adopt the CYREN rules and their scores, LocalView enables customers to disable or alter the scores of CT rules in cases that some local optimization is required.

3.2.1.1 Changing a CT Rule Score

To change the score of CYREN CT Rule, override the score of the rule as a Custom Rule. For example, to override the score of CT_Rule1 tag, enter the following line in a custom rules file: score CT_Rule1 0.5

While scanning, the score from the Custom Rule file (either SW Custom Rule or LOC Custom Rule) will override the score in the CT Rules file.

Note: *CT Rule expressions cannot be updated.*

3.2.1.2 Disabling a CT Rule

To disable a rule in the CYREN CT Rule list, set the score to 0.

4 Blacklist and Whitelist Rules

CYREN's LocalView engine enables customers to maintain and apply Whitelists and Blacklists. Blacklist and Whitelist rules are considered as a special type of custom rules. Whitelisted messages are matched to CYREN's NonSpam classification, while Blacklisted messages receive a classification of ConfirmedSpam.

When a value is placed in both the Whitelist and the Blacklist, the Whitelist will override the Blacklist entry. Messages that match an entry in either list are short-circuited, meaning, no further scanning is required to determine whether it is spam or non-spam.

Both sender email accounts and IP addresses can be blacklisted or whitelisted. Both IPv4 and IPv6 IPs are supported. The local IP Ignore list should include all IP addresses of the organization's trusted networks.

Note: *The current version of LocalView does not support IPv6 address formats in the IP Ignore List.*

Where an entry from the IP Ignore list is found in either the Whitelist or the Blacklist, the Whitelist/Blacklist entry is ignored.

The Blacklist and Whitelist rules are defined in standard custom rule files, in the locations defined above. System wide and local black and white rules are supported in an identical manner as regular custom rules. You can validate your rules by using the ValidateCustomRules parameter. For more information, see *Validating Custom Rules*.

4.1.1 Blacklist and WhiteList Header Definitions

White and black listing lookups are performed on From-related headers. The specific list of headers is defined in the parameter WBLHeaderListFrom.

The default value of WBLHeaderListFrom is "Envelope-Sender, Resent-Sender,X-Envelope-From,From,list-unsubscribe,Sender,Mail-From".

Note: *The WBLHeaderListFrom parameter value can be defined in the ctengine connection string or in the ctasd.conf file.*

4.1.2 Black and White Rule Syntax

Dedicated custom rule functions are defined for Black and White custom rules:

- white_from rule function to white list message based on sender email, domain
- black_from rule function to black list message based on sender email, domain
- white_from_rcvd function to white list message based on IP address found in Received header
- black_from_rcvd function to black list message based on IP address found in Received header

A white/black listed sender address can be defined either as an exact match string or as a substring definition. Regex expressions are not supported.

Multiple white/black entries can be placed in a single white_from/black_from line.

Exact Match Examples:

Function	Example	Explanation
black_from or white_from	sender1@xyz.com	This rule will scan WBLHeaderListFrom headers for the sender1@xyz.com email account.
black_from or white_from	@xyz.com	This rule will scan WBLHeaderListFrom headers for the string @xyz.com.
black_from_rcvd or white_from_rcvd	2.3.4.5	This will scan the Received header for an exact IPv4 address.
black_from_rcvd or white_from_rcvd	1.2.3.0:255.255.255.0	This will scan the Received header for an IPv4 mask.
black_from_rcvd or white_from_rcvd	2001:0db8:0000:0000:1111:2222:3333:4444	This rule will scan the Received header for an exact IPv6 address.

Note: *The current version of LocalView does not support IPv6 mask definitions.*

5 CYREN IP Reputation Rules

LocalView leverages CYREN's IP reputation data to add/remove message scores based on the Sender IP of the message. The SenderIP may receive one of the following IP reputation classifications rule tags:

- IP-Black
- IP-Dark-Grey
- IP-Grey
- IP-White
- IP-Very-White

The default score for each one of the IP classifications is:

- IP-Black=5
- IP-Dark-Grey=4
- IP-Grey=3.5
- IP-White= -0.3
- IP-Very-White= -0.6

A customer can change the default IP classification scores as detailed above, by updating the score values in the parameter CTIPRepRBL_Tags.

6 Message Classification, Scores and Caught Rules

Caught rules are those that are found to be relevant to the message being scanned. Each message scanned by LocalView is scanned for the following rules:

- CT Rules
- Custom Rules (both system-wide and local custom rules)
- Black and white list rules
- IP reputation rules

6.1 Final Score

LocalView aggregates together all the scores of the "caught" rules to calculate the final score of a message. The final message score can be accessed using ctengine's function call GetScore, or if ctasd is being used, by accessing the X-CTCH-Score classify message response header.

6.2 "Caught" Rules

LocalView registers all "caught" rules participating in a message's final score. To view the "caught" rules:

- **ctengine:** Use GetRules function
- **ctasd:** Access X-CTCH-Rules classify message response header

6.3 Classification By Factoring Only Custom Rules

If a customer wishes to review the classification resulting from custom rules only:

- **ctengine:** Use GetClassCustom function
- **ctasd:** Access the X-CTCH-SpamCust classify message response header

6.4 Score Factoring Only Custom Rules

If a customer wishes to review the score resulting from only custom rules only:

- **ctengine:** Use GetScoreCustom function
- **ctasd:** Access the X-CTCH-ScoreCust classify message response header