Cisco ASR 5000 Series Content Filtering Services Administration Guide

Version 10.0

Last Updated June 30, 2010

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-22959-01

Cisco ASR 5000 Series Content Filtering Services Administration Guide

CONTENTS

About this Guide

This document pertains to features and functionality that run on and/or that are related to the Cisco® ASR 5000 Chassis, formerly the Starent Networks ST40.

Conventions Used

The following tables describe the conventions used throughout this documentation.

Icon	Notice Type	Description
	Information Note	Provides information about important features or instructions.
	Caution	Alerts you of potential damage to a program, device, or system.
	Warning	Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.
	Electro-Static Discharge (ESD)	Alerts you to take proper grounding precautions before handling a product.

Typeface Conventions	Description
Text represented as a `screen display`	This typeface represents displays that appear on your terminal screen, for example: `Login:`
Text represented as **`commands`**	This typeface represents commands that you enter, for example: **`show ip access-list`** This document always gives the full form of a command in lowercase letters. Commands are not case sensitive.
Text represented as a **`command`** *`variable`*	This typeface represents a variable that is part of a command, for example: **`show card`** *`slot_number`* slot_number is a variable representing the desired chassis slot number.
Text represented as menu or sub-menu names	This typeface represents menus and sub-menus that you access within a software application, for example: Click the File menu, then click New

Command Syntax Conventions	Description
{ **`keyword`** or *`variable`* }	Required keywords and variables are surrounded by grouped brackets. Required keywords and variables are those components that are required to be entered as part of the command syntax.

Command Syntax Conventions	Description
[**keyword** or *variable*]	Optional keywords or variables, or those that a user may or may not choose to use, are surrounded by square brackets.
|	With some commands there may be a group of variables from which the user chooses one. These are called alternative variables and are documented by separating each variable with a vertical bar (also known as a pipe filter). Pipe filters can be used in conjunction with required or optional keywords or variables. For example: **{ nonce | timestamp }** OR [**count** *number_of_packets* | **size** *number_of_bytes*]

Contacting Customer Support

Use the information in this section to contact customer support.

For New Customers: Refer to the support area of http://www.cisco.com for up-to-date product documentation or to submit a service request. A valid username and password is required to this site. Please contact your local sales or service representative for additional information.

For Existing Customers with support contracts through Starent Networks: Refer to the support area of https://support.starentnetworks.com/ for up-to-date product documentation or to submit a service request. A valid username and password is required to this site. Please contact your local sales or service representative for additional information.

IMPORTANT: For warranty and repair information, please be sure to include the Return Material Authorization (RMA) tracking number on the outside of the package.

Chapter 1
Content Filtering Support Overview

This chapter provides an overview of the Content Filtering In-line Service feature.

This chapter covers the following topics:

Introduction

Content Filtering is an in-line service available for 3GPP and 3GPP2 networks to filter HTTP and WAP requests from mobile subscribers based on the URLs in the requests. This enables operators to filter and control the content that an individual subscriber can access, so that subscribers are inadvertently not exposed to universally unacceptable content and/or content inappropriate as per the subscribers' preferences.

The Content Filtering service offers the following solutions:

- URL Blacklisting:

  In the URL Blacklisting solution, all HTTP/WAP URLs in subscriber requests are matched against a database of "blacklisted" URLs. If there is a match, the flow is discarded, redirected, or terminated as configured. If there is no match, subscribers view the content as they would normally.

  URL Blacklisting may/may not be a subscriber opt-in service, operators can enable URL Blacklisting either for all subscribers or for a subset of subscribers. Typical cases include applying a blacklisted database of child porn URLs to all subscribers so that they are inadvertently not exposed to such universally unacceptable content.

- Category-based Static Content Filtering:

  In Category-based Static Content Filtering, all HTTP/WAP URLs in subscriber requests are matched against a static URL categorization database. Action is taken based on a URL's category, and the action configured for that category in the subscriber's content filtering policy. Possible actions include permitting, blocking, redirecting, and inserting/altering content.

IMPORTANT: Category-based Static-and-Dynamic Content Filtering feature is not supported in this release.

Typically Category-based Content Filtering is an opt-in service, subscribers self-choose a content-filtering policy or plan, such as Teen, Child, Adult, etc., and are subjected to content filtering as per their chosen plan. Also, the content-filtering policies of different subscribers may be different, enabling differential access of content to them. This solution provides maximum flexibility, and is also referred to as the Policy-based Content Filtering.

Both URL Blacklisting and Category-based Content Filtering support can be concurrently enabled on a system.

Content Filtering uses Deep Packet Inspection (DPI) feature of Enhanced Charging Service (ECS) / Active Charging Service (ACS) to discern HTTP and WAP requests.

Supported Platforms and Products

Content Filtering is an in-line service supported on ASR5000 running 3GPP, 3GPP2, and LTE core network services.

Licenses

URL Blacklisting

URL Blacklisting is a licensed feature requiring the following license:

[600-00-7801] *Blacklisting Integrated Service*

Category-based Content Filtering

Category-based Content Filtering is a licensed feature requiring the following license:

[600-00-7586] *Integrated Content Filtering Service, 1k Sessions*

For information on license requirements for any customer-specific features, please contact your local sales/service representative.

IMPORTANT: External Content Filtering Server support through Internet Content Adaptation Protocol (ICAP) interface is a licensed feature, requiring a separate license. For more information, see the *ICAP Interface Support* chapter of the *System Enhanced Feature Configuration Guide*.

IMPORTANT: For information on obtaining and installing licenses, refer to *Managing License Keys* in the *System Administration and Configuration Guide*.

URL Blacklisting Support

In the URL Blacklisting solution, a blacklist is a list of known URLs/URIs, which for some reason are being denied recognition. The blacklist can be obtained from a known source such as the National Center for Missing & Exploited Children (NCMEC, http://www. missing kids.com), or any other IP source. The blacklist is a clear text file, the file must be named cumulative.csv, and must use the same format as the blacklist file from NCMEC. For more information on the blacklist file, please contact your local service representative.

Unlike the Category-based Content Filtering solution, which categorizes URLs as per a static database and takes different actions based on the different policies associated with subscribers, URL Blacklisting is applicable to all subscribers associated with a blacklisting-enabled rulebase. The same blacklist database is used for all subscribers, and for a specific URL, the same action is taken for all subscribers.

The blacklist file is downloaded and converted into a non human-readable optimized format (OPTBLDB) and then made available in the system. Once in place, all HTTP and WAP requests from subscribers are inspected in order to determine the requested destination URL/URI. If the URL/URI is not present in the blacklist then the request is passed on as usual. If the URL/URI is present in the blacklist, the request is dropped, or the flow is redirected or terminated as configured. There is no indication/messaging sent to the requesting subscribers that the requested HTTP/WAP URL/URI was rejected due to a blacklist match.

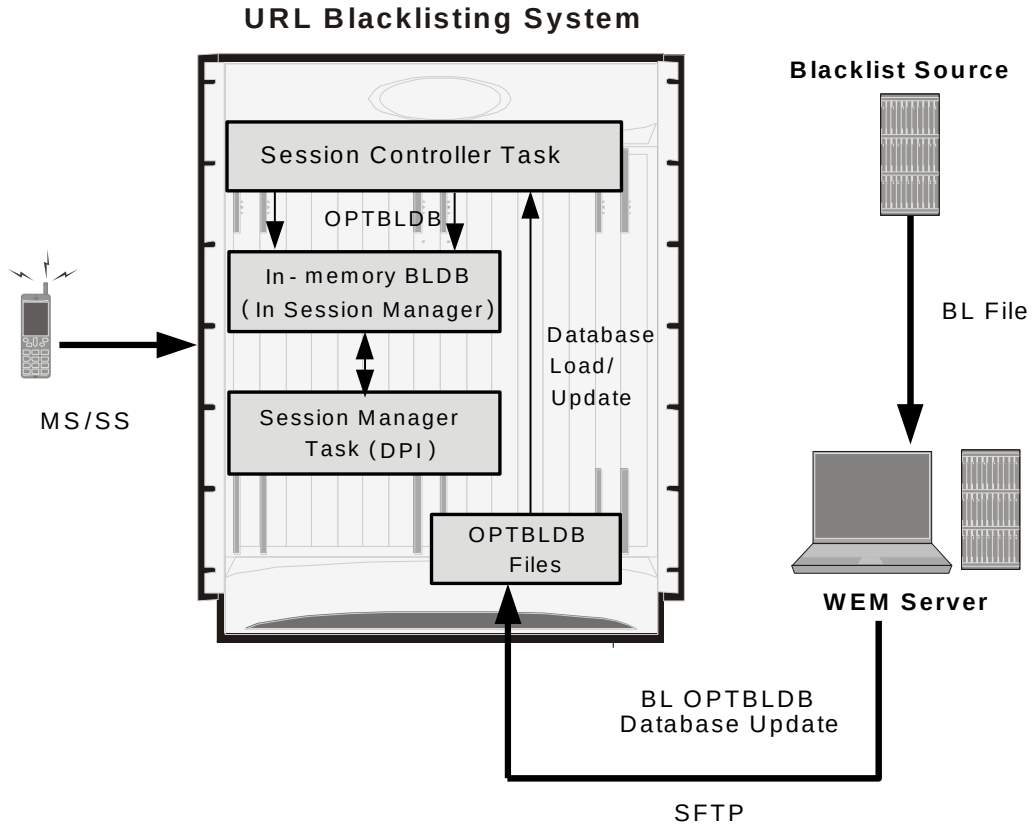
The URL Blacklisting match-method can be configured to either be generic or to look for any URL/URI in its exact, literal form.

The system generates usage/event data that can be utilized as the basis for blacklist reporting. The offline reports consist of, at a minimum, a running total of the number of times a match was made against the blacklist without any information regarding the specifics of the request.

The default/configured number of versions of the Blacklist database are maintained on the chassis (both the SPCs). This enables reverting to a particular version if required.

The following figure shows the high-level URL Blacklisting architecture with ECS, and other components in a deployment scenario.

Figure 1. High-Level Architecture of URL Blacklisting with ECS



URL Blacklisting Solution Components

The URL Blacklisting solution uses the deep-packet inspection capabilities of ECS for URL/URI extraction.

ECS functionality is managed by the following components:

- **Session Controller (SessCtrl)**: The SessCtrl runs on the primary SPC/SMC and is responsible for managing ECS and URL Blacklisting services.
- **Session Manager (SessMgr)**: A single SessMgr treats ECS charging and URL Blacklisting that is applicable to common subscriber sessions.

Apart from ECS, the URL Blacklisting solution uses the following components:

- Content Filtering Subsystem in ECS
- Web Element Manager (WEM)

Web Element Manager (WEM)

The WEM is a server-based application enabling complete element management of the system. The UNIX-based server application works with the network elements within the system using the Common Object Request Broker Architecture (CORBA) standard.

IMPORTANT: For information on WEM administration, refer to the *Web Element Manager Installation and Administration Guide*.

The WEM server must be set up with access to the following networks:

- Internet—to communicate with the source of the blacklist file (NCMEC/other)

The WEM application includes the following features:

- Single point of management for a large operator deployment
  - Service configuration and monitoring
  - Alarm/trap management for the WEM server
- URL Blacklisting database management functions:
  - Downloads the URL Blacklist database *(cumulative.csv)* from the specified source at configured schedule
  - Converts the URL Blacklist database *(cumulative.csv)* file to Starent Format Master Database (SFMDB) file
  - Computes OPTBLDB suitable for updating the system
- Distributes OPTBLDB/OPTBLDB-INC files to the chassis automatically at configured interval

How URL Blacklisting Works

This section describes how URL Blacklisting works.

Blacklist Updates

The following steps describe how the blacklist is updated in the system:

Step 1 The WEM downloads the blacklist file from the specified source (NCMEC/other). The clear text file is converted into a non-human readable optimized format (OPTBLDB) and then pushed to the chassis.

Step 2 The WEM pushes the optblk.bin file to the chassis (to the *flash/pcmcia* device) at pre-determined intervals. The optblk.bin file contains the full blacklist. If this file is verified to be correct it replaces the optblk.bin file on the chassis, and the last optblk.bin is rolled over.

Step 3 The blacklist file is auto-detected by the Session Controller (SessCtrl), which verifies the integrity of the Blacklist database using checksums, and then loads it.

The new blacklist is loaded only if it has been received properly. If the full Blacklist database is not found, corrupted, or if the loading fails, traps are generated. Correspondingly clear traps are also generated on a valid Blacklist database being available, and after a successful load.

Step 4 The SessMgrs read the file and load the blacklisted URLs in a local in-memory database.

IMPORTANT: The URL Blacklisting feature is enabled only if the url-blacklisting action is set in any of the rulebases. Thus, the automatic detection of the Blacklist database, storing it in memory, and loading onto the SessMgrs will happen only if the url-blacklisting action is set in any of the rulebases.

Step 5 The Blacklist database is loaded on each SessMgr as and when they come up (if URL Blacklisting is set in any rulebase) or when URL Blacklisting gets set in any of the rulebases.

When the SessMgrs start for the first time or after recovery, if URL Blacklisting is set in any of the rulebases, the stored Blacklist database at SessCtrl is loaded onto the SessMgrs. This holds true for standby managers as well i.e., when standby managers come up the Blacklist database is loaded onto them.

Whenever a SessMgr is killed, standby manager which already has the Blacklist database loaded takes its place, and a new standby manager is created which loads the Blacklist database as part of SessMgr getting started for the first time.

If SessCtrl is killed, while recovering it checks if URL Blacklisting is set in any of the rulebases, if set it will store the Blacklist database onto itself and load all the SessMgrs as well.

Step 6 When a new Blacklist database is loaded on to the SessMgrs, the new database (and any stored versions that have rolled over) are synced to the other SPC so that after switchover, the proper Blacklist database can be accessed.

URL Blacklisting Action

The following steps describe how the URL Blacklisting feature works:

Step 1 When an initial HTTP/WAP request comes for ECS processing and is processed by the ECS subsystem, a check is made to see if the URL Blacklisting support is enabled.

Step 2 If enabled, the URL is extracted from the incoming request and is matched with the local in-memory Blacklist database.

If a match is found for the URL in the Blacklist database, the packets are treated as per the blacklisting action configured—Discard, Redirect, or Terminate flow.

In case of multiple HTTP requests in the same TCP packet, if any of the URLs match the packet is treated as per the blacklisting action configured.

If a match is not found, the request is allowed to pass through.

Category-based Content Filtering Support

The Category-based Content Filtering application is a fully integrated, subscriber-aware in-line service provisioned on chassis running HA services. This application is transparently integrated within the ECS, and utilizes a distributed software architecture that scales with the number of active HA sessions on the system.

Content Filtering policy enforcement is the process of deciding if a subscriber should be able to receive some content. Typical options are to allow, block, or replace/redirect the content based on the rating of the content and the policy defined for that content. For the list of content categories, refer to the *Category List* appendix in the *Content Filtering Services Administration Guide*.

Benefits of Category-based Content Filtering

The Category-based Content Filtering solution enables operators to ensure a simplified end-to-end traffic flow with a simple network topology. In-line deployment of Content Filtering provides a more attractive solution in contrast to out-of-line solutions where the filtering and policy enforcement is provided at some offload point that is decoupled from the bearer-processing layer.

The out-of-line model forces a session to make multiple hops through a redundant array of equipment which has a negative impact on traffic latency and limits subscriber and network visibility. In addition, the out-of-line model requires all subscriber sessions to be steered to the adjunct Content Filtering platform for policy enforcement regardless of whether this additional processing is needed. This leads to increased bandwidth provisioning requirements on gateway routers.

To facilitate network simplicity, it makes sense to leverage the benefits of deep packet inspection at a single policy enforcement point that is tied to the bearer processing layer. The advantages of this approach implemented in include the following benefits:

- **Reduced processing latency**: In-line service processing eliminates unnecessary hand-offs and forwarding to external network elements.
- **Simplified policy provisioning**: Enables all policies like Content Filtering, ECS and QoS to be retrieved from same AAA/Policy Manager signaling interface thus reducing total volume of control transactions and associated delay.
- **Simplified provisioning and complete service integration**: Provisioning of separate resources like packet processing cards for processing subscriber data sessions and discrete services are eliminated. The same CPU can contain active Session Manager tasks for running Content Filtering and ECS charging.
- **Integration with Content Service Steering (CSS) architecture**: Enables applicable sessions to be forwarded to the in-line content filtering subsystem while delay and time sensitive voice/multimedia services immediately forwarded to Internet.
- **Service control**: Precise control over the interaction and service order handling of bearer flows with required applications like Content Filtering, ECS, Subscriber-aware Stateful Firewall, integrated Policy Charging and Rules Function (PCRF) for Service Based Bearer Control.

Apart from the advantages described previously, Category-based Content Filtering service reduces the requirement of over-provisioning of capacity at neighboring gateway routers. It also eliminates requirements of external Server Load Balancers and enhances the accuracy in subscriber charging records.

The Category-based Content Filtering solution has the following logical functions:

- Deep Packet Inspection (DPI) for Content Rating (event detection and content extraction)
- Content Rating Policy Enforcement; for example, permit, discard, deny, redirect
- Content-ware accounting CF-EDR generation for events of interest

ECS and Content Filtering Application

The Category-based Content Filtering subsystem is integrated within the Enhanced Charging Service (ECS) subsystem. Although it is not necessary to provision content-based charging in conjunction with content filtering, it is highly desirable as it enables a single point of deep-packet inspection for both services. It also enables a single policy decision and enforcement point for both services thereby streamlining the required number of signaling interactions with external AAA/Policy Manager servers. Utilizing both services also increases billing accuracy of charging records by insuring that mobile subscribers are only charged for visited sites content.

The Category-based Content Filtering solution uses Content Filtering Policy to analyze the content requested by subscribers. Content Filtering Policy provides a decision point for analyzed content on the basis of its category and priority.

The Category-based Content Filtering solution also utilizes ECS rulebases in order to determine the correct policy decision and enforcement action such as accept, block, redirect, or replace. Rulebase names are retrieved during initial authentication from the AAA/Policy Manager. Some possible examples of rulebase names include Consumer, Enterprise, Child, Teen, Adult, and Sport. Rulebase names are used by the ECS subsystem to instantiate the particular rule definition that applies for a particular session. Rulebase work in conjunction with a content filtering policy and only one content filtering policy can be associated with a rulebase.

IMPORTANT: For more information on rulebases and rule definitions, refer to the *Enhanced Charging Services Administration Guide*.

The ECS subsystem includes L3–L7 deep packet inspection capabilities. It correlates all L3 packets with higher layer criteria such as URL detection within an HTTP header, it also provides stateful packet inspection for complex protocols like FTP, RTSP, and SIP that open ports for the data path.

The Content Filtering subsystem uses the deep-packet inspection capabilities of ECS for URL/URI extraction.

ECS functionality is managed by the following components:

- **Session Controller (SessCtrl)**: The SessCtrl runs on the primary SPC/SMC and is responsible for managing ECS and Content Filtering services.
- **Session Manager (SessMgr)**: A single SessMgr treats ECS charging and Content Filtering that is applicable to common subscriber sessions.

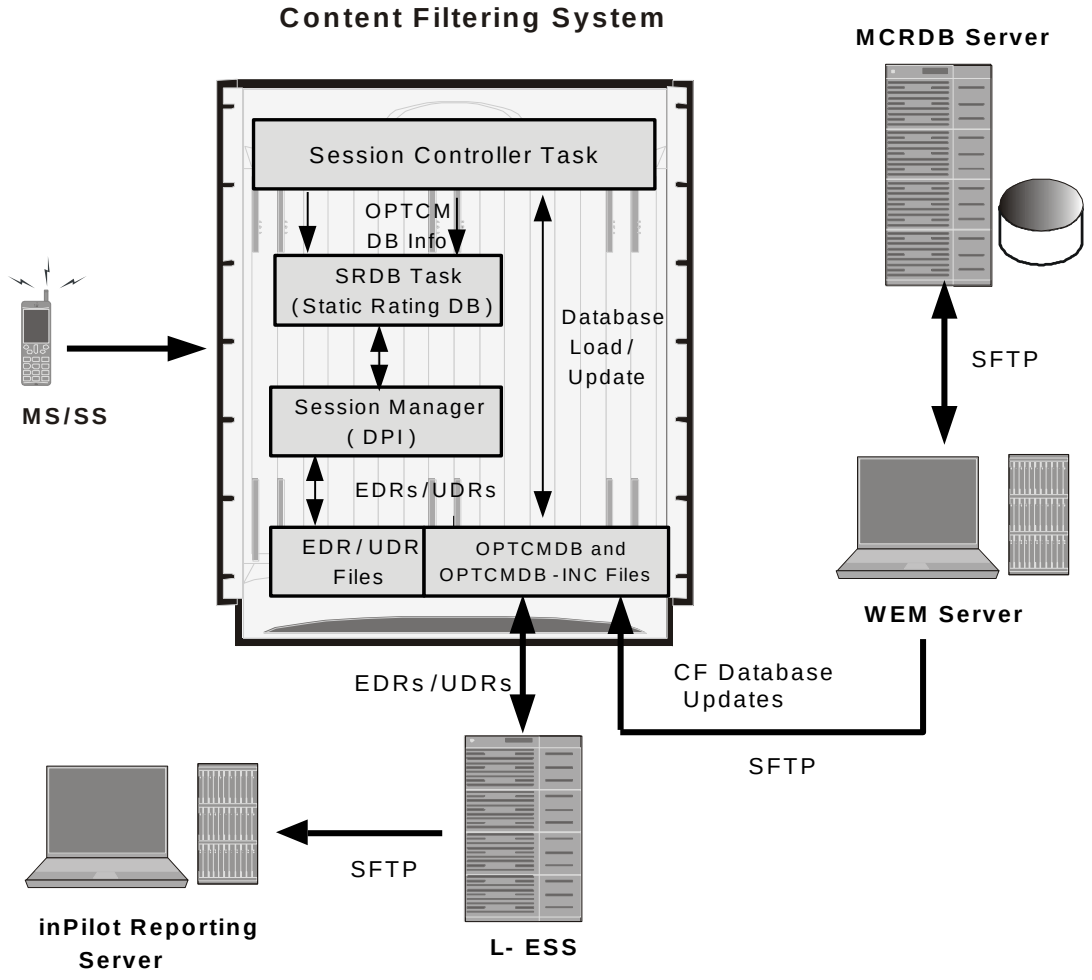
Components of Category-based Content Filtering Solution

The Category-based Content Filtering solution uses the following components:

- Content Filtering Subsystem in ECS
- Content Rating Rules Update Server
- Master Content Rating Database Server (MCRDBS)
- ECS Storage System (ESS)
- RADIUS Server/Policy Manager
- Web Element Manager (WEM)
- inPilot

The following figure shows a high-level view of the Category-based Content Filtering architecture with ECS, and other components in a deployment scenario.

Figure 2. High-Level Architecture of Category-based Content Filtering



Category-based Content Filtering Subsystem

The Content Filtering solution comprises the following content rating and category databases:

- Static Rating Categorization Database

Static Rating Categorization Database (SRDB)

This is an internal categorization database (periodically synchronized with an external server) that provides ratings for publicly accessible traditional and mobile Web sites. When the SessMgr passes a URL/URI to internal list server, the list server returns a list of matching category ratings.

The list server is used to determine whether a Web site has already been classified. When the list server passes back a category rating to the filtering application, the rating is compared against the Category Policy ID applied for the subscriber to determine the appropriate action like accept, block, redirect, or replace. If the list server returns a clean rating, there is no need to perform a real-time analysis of any content delivered by the site.

When a blocked or rejected content rating is returned, the SessMgr can insert data such as a redirect server address into the bearer data stream. If no rating is returned this means the site is capable of returning either clean or unacceptable content.

Each SRDB contains a replication object consisting of hash tables that map known Web sites and their subdirectories to their respective category ratings. The SessCtrl reads the index of SRDB tables with a data structure that associates keys with URL rating values and loads it onto the SRDB managers.

To boost performance and provide high availability, SRDB Manager provides functionality to load the Optimized Content Rating Master Database (OPTCMDB) volumes from its peer SRDB task. If the peer SRDB task is not in loading state then the OPTCMDB loading is done through SessCtrl to the recovered SRDB task.

Rater Package Model Files

The real-time analyzer requires a model file that defines the features which are necessary to classify a Web page as belonging to a specific category and language. A model file per category is created by analyzing the traits of thousands of pages of that category and thousands of pages that does not belong to that category. For some categories, a feature counter file is used to decide whether or not to evaluate the Web page against the respective model file.

When URL Blacklisting solution is the only content filtering enabled on a system, no SRDB tasks are spawned at startup. Only when either Category-based Content Filtering is enabled in isolation, or with URL Blacklisting, the SRDB tasks are spawned.

Content Rating Rules Update Server

This is a third-party content rating solution for exporting content filtering rules database information to the Category-based Content Filtering system. In addition, while exporting database updates, it collects reports of URLs processed by ECS and Content Filtering services that are reported as unknown in the deployed static rating database. This server analyzes these URLs and provides the rating in future updates for static rating database.

This server provides the following support to Master Content Rating Database Server (MCRDBS) for the content rating function:

- Provides full Vendor Format Master Database files (VFMDB) to MCRDB server on request from MCRDBS.
- Provides incremental Vendor Format Master Static URL Database file (VFMDB-INC) to MCRDBS when any incremented VFMDB is available and requested from MCRDBS.
- Receives the Unknown URLs file (Vendor Format Unknown Database File (VFUNKDB)) from MCRDBS.

Master Content Rating Database Server (MCRDBS)

The Category-based Content Filtering solution provides a Master Content Rating Database Server to convert the VFMDB to SFMDB. It handles both full and incremental updates and processes them on a configured schedule.

This server is also responsible for distribution of SFMDB data files to WEM servers in the customer support infrastructure on a configured interval.

The server is responsible for following functionality as the MCRDBS solution:

- Database fetching: Pulls VFMDB files from third-party Content Rating Server to MCRDBS.
- Database conversion: Converts VFMDB files to SFMDB files. It also handles the incremented and unknown database files.
- Database poller: Provides the converted SFMDB database files for WEM in a preconfigured path.
- E-mail notification: Provides alerts and notification to the administrator for alarms.

ECS Storage System

The local external storage server is a part of ECS Storage System in the ECS solution architecture.

The L-ESS is a storage application running on redundant highly available servers that collect and process EDRs and UDRs from which billing events and reports are generated. Either the system pushes the EDR/UDR files to the L-ESS, or the L-ESS fetches them from the system and processes them into formats suitable for billing mediation servers and inPilot. The L-ESS consolidates the processed EDR/UDR files into a database for report generation through inPilot. The database generated on an ESS by processing EDR/UDR records is a superset of the database required by inPilot.

IMPORTANT: For more information on External Storage Systems, refer to the *ESS Installation and Administration Guide*.

RADIUS Server and Policy Manager

The function of the RADIUS Server/Policy Manager in the Content Filtering solution is to provide per-subscriber Content Filtering provisioning information when a subscriber's session is established. It can also issue a Change-of-Authorization (CoA) to update an in-progress session to modify the Content Filtering policy for a subscriber.

The following are the basic functions provided by a RADIUS Server/Policy Manager in the Content Filtering solution:

- Support for the in/out ACL attributes to direct traffic through ECS for processing of subscriber traffic
- Support for ECS rulebase VSA to select the ECS rulebase to be applied to filtered traffic

- Support for Content Filtering Policy identifier VSA to select the content filtering policy within the selected rulebase for a subscriber
- Support exporting a subscriber provisioning record based on MSID to the customer service interface (Customer Care Interface) so that operator's customer care executive can see the provisioned content filtering policy for a subscriber

Web Element Manager (WEM)

The WEM is a server-based application providing complete element management of the system. The UNIX-based server application works with the network elements within the system using the Common Object Request Broker Architecture (CORBA) standard.

IMPORTANT: For information on WEM administration, refer to the *Web Element Manager Installation and Administration Guide*.

WEM server must be set up with access to the following networks:

- Internet: To communicate with the Master Content Rating Database Server (MCRDBS) which provides update files.

For Category-based Content Filtering, the WEM application includes the following features:

- Single point of management for a large Content Filtering Service operator deployment:
  - Content Filtering service configuration and monitoring
  - Alarm/trap management
- Configures and manages the operator-defined White/Black static rating database (WBLIST) for the network (WBLIST is maintained in SFMDB format)
- Content filtering database management functions:
  - Performs database processing in the background
  - Imports full and incremental SFMDB and SFMDB-INC files from the MCRDBS on a configured schedule
  - Processes incremental SFMDB-INC updates from MCRDBS maintaining an updated SFMDB file
  - Merge the operator's WBLIST database with the most recent SFMDB creating a SFCMDB
  - Computes an incremental update to the OPTCMDB-INC suitable for updating the Content Filtering subsystem that contains a previous version OPTCMDB
- Distributes OPTCMDB/OPTCMDB-INC files to the chassis automatically at configured interval

inPilot

The inPilot application is a Web-based application providing a unified reporting interface for diverse data from the in-line service and storage applications. The inPilot application provides comprehensive and consistent set of statistics and customized reports, statistical trending, report scheduling and distribution from chassis / in-line service product. For example, a subscriber's Quality of Experience, top 10 sites visited, top 10 users, and so on. The inPilot application facilitates and enhances the operators' ability to simply and easily determine the health and usage of the network.

The inPilot application supports the generation of various reports including CF-EDR reports in PDF and XML formats. The CF-EDR reports provide the summary of traffic over CF categories, CF actions, and CF ratings. It also provides the list of top N subscribers and URLs based on their unique subscriber's hit count and total usage.

- Summary Reports:
  - Category summary (volume/hits)
  - Action summary (volume/hits)
  - Rating summary (volume/hits)
- Top N Reports:
  - Top N Subscribers by volume/hits
  - Top N URLs by volume/hits

The CF-EDR files are pushed from L-ESS to inPilot at a configured time interval and stored in a specified data directory on the inPilot server. It can also create the files from CF-EDRs for unrated URLs which can be pulled by WEM.

IMPORTANT: For more information on the reports, refer to the *inPilot Online Help* documentation.

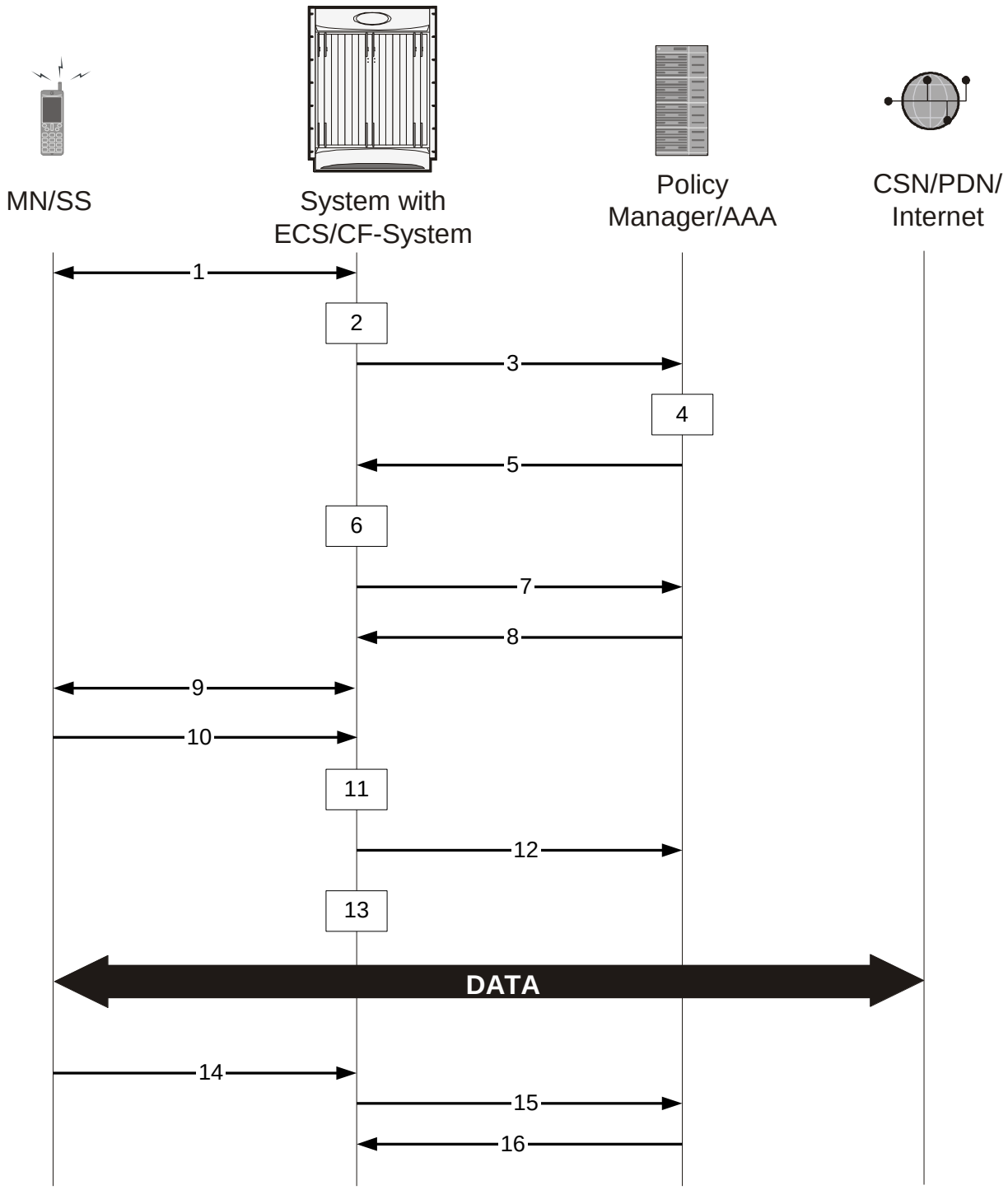
How Category-based Content Filtering Works

The Content Filtering Subsystem which is integrated into the ECS subsystem consists of an onboard static categorization database. The filtering service uses the Deep Packet Inspection (DPI) capabilities of the ECS subsystem to classify and partition application or protocol specific flows into virtual sessions.

Content analyzers are used to identify various types of flows such as HTTP, MMS/WAP, and POP3 E-mail. A typical HTTP request for a Web page, for example, invokes TCP and HTTP traffic analyzers. Any HTTP field including URLs or URIs can be identified. When a subscriber session is bound by CSS to an ECS running content filtering service, the URL/URI is extracted and compared against the static categorization database.

The following figure and the steps describe how Category-based Content Filtering works during a subscriber call:

Figure 3. Content Filtering Call Flow



Step 1 MS requests for registration to the system.

Step 2 System processes MS-related information with Content Filtering subsystem.

Step 3 System sends the AAA Access Request to AAA server for MS.

Step 4 AAA server processes the AAA Access Request from the Content Filtering subsystem to create the session, and the Policy Manager in AAA server uses subscriber identification parameters including NAI (*username@domain*), Calling Station ID (IMSI, MSID) and Framed IP Address (HoA) as the basis for subscriber lookup.

Step 5 The Policy Manager and AAA generate and send an Access Accept message including all policy and other attributes to establish the session to the Content Filtering subsystem.

The Policy Manager and/or AAA include following attributes in the Access Accept message:

- **Filter ID or Access Control List Name**: Applied to subscriber session. It typically contains the name of the Content Service Steering (CSS) ACL. The CSS ACL establishes the particular service treatments such as Content Filtering, ECS, Traffic Performance Optimization, Stateful Firewall, VPN, etc. to apply to a subscriber session and the service order sequence to use in the inbound or outbound directions. Real-time or delay sensitive flows are directly transmitted to the Internet with no further processing required. In this case, no CSS ACL or Filter ID is included in the Access Response.
- **SN-CF-Category-Policy**: Applied to the subscriber content flow. Policy ID included in this attribute overrides the policy identifier applied to subscriber through rulebase or APN/Subscriber configuration. This content filtering policy determines the action to be taken on a content request from subscriber on the basis of its category. At anytime only one content filtering policy can be associated with a rulebase.
- **SN1-Rulebase Name**: This custom attribute contain information such as consumer, business name, child/adult/teen, etc.). The rulebase name identifies the particular rule definitions to apply. Rulebase definitions are used in ECS as the basis for deriving charging actions such as prepaid/postpaid volume/duration/destination billing and charging data files (EDRs/UDRs). Rulebase definitions are also used in content filtering to determine whether a type of user class such as teenagers should be permitted to receive requested content belonging to a particular type of category such as adult entertainment, gambling or hate sites. Rulebase definitions are generated in the Active Charging Configuration Mode and can be applied to individual subscribers, to domains or on per-context basis.

Step 6 Content Filtering subsystem creates a new session for MS.

Step 7 Content Filtering subsystem sends Accounting-Start messages to AAA server.

Step 8 AAA server sends Accounting-Start response message to Content Filtering subsystem.

Step 9 Content Filtering subsystem establishes data flow with MS.

Step 10 MS requests for data with URL name.

Step 11 Within the system access control list (ACL) processes the request and directs the request to ECS/Content Filtering subsystem based on the subscriber configuration.

Step 12 System performs ECS action on the content and then applies content filtering if required.

Within the system, if the bearer flow is treated by Content Filtering or other in-line services, the SessMgr feeds it to the Content Service Steering (CSS) API. If Content Filtering is the first service touch point, TCP and HTTP traffic analyzers within a given SessMgr utilize deep-packet inspection to extract the requested URL.

Step 13 The Content Filtering subsystem processes the URL access request.

When only Static Content Filtering is enabled, first the URL is looked-up in the cache maintained at SessMgr for static URL requests, if there is a hit, the category is returned, if its a miss, a URL look-up is performed by an onboard SRDB for static rating.

- If a category is returned, action is taken as configured for that category in the subscriber's Content Filtering policy:
  - allow: If the category is permitted by the subscriber's content filtering policy, the request is sent to the server, and the response transmitted to the subscriber's mobile.
  - content-insert: The system notifies the subscriber's mobile of the blocked content by inserting a specified message within the IP data stream, and prevents access to the requested content. The insert string is as specified in the subscriber's content filtering policy.
  - discard: The system silently discards the request packet(s).
  - redirect-url: The system inserts a specified redirect server address in the bearer data stream and returns an HTTP error message to the subscriber's mobile. The redirect address is as specified in the subscriber's content filtering policy.

    The redirect server may prompt the subscriber to send additional security credentials in order to access the requested content.
  - terminate-flow: The system gracefully terminates the TCP connection between the subscriber and server, and sends a TCP FIN to the subscriber and a TCP RST to the server.
  - www-reply-code-and-terminate-flow: The system terminates the flow with a specified reply code to the subscriber's mobile. The reply code is as specified in the subscriber's content filtering policy.
- If a category is not returned / the URL is not present in the database, the system takes the action as configured for the UNKNOW category in the subscriber's Content Filtering policy.
- If for the category returned there is no action configured in the subscriber's content filtering policy, the default action is taken.

If the SRDB task is timed out or some other failure happens, the action configured for failure is taken.

Step 14 MS requests for session termination.

Step 15 System sends Accounting-Stop Request to the AAA server.

Step 16 AAA server stops the accounting for the MS for content filtering session and sends Accounting-Stop-Response to the system.

How URL Blacklisting and Category-based Content Filtering Work Concurrently

Both URL Blacklisting and Category-based Content Filtering can be concurrently enabled in a system. The following describes how URL blacklisting and content filtering are performed on HTTP/WAP traffic when concurrently enabled on a system:

Step 1 If both URL Blacklisting and Category-based Content Filtering are enabled, first URL blacklist matching is performed, and then, if required, content filtering is performed.

When an HTTP/WAP request comes for ECS processing, a check is made to see if the URL Blacklisting feature is enabled. If enabled, the URL is extracted from the incoming request and is matched with the local Blacklist database.

- If a match is found for the URL in the Blacklist database, the packets are subjected to the blacklisting action configured in the rulebase—Discard, Redirect, or Terminate flow. In case of multiple HTTP requests in the same TCP packet, if any of the URLs is blacklisted, then action is taken on the packet.
- If a match is not found in the Blacklist database, then Category-based Content Filtering is performed.
    - If Category-based Static Content Filtering is enabled, static rating is performed and action taken as configured for the category returned in the subscriber's content filtering policy.

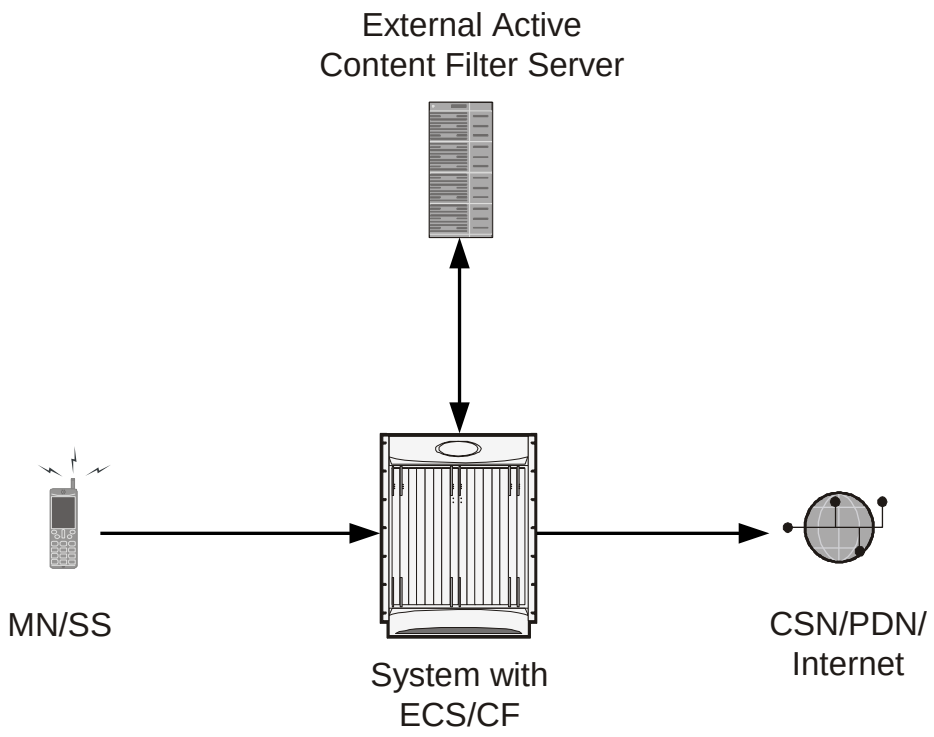
Step 2 If URL Blacklisting is enabled and Category-based Content Filtering is disabled, and a match is not found for the URL in the Blacklist database, the request is allowed to pass through, and no Content Filtering EDRs are generated for those flows.

Content Filtering Server Group Support

ECS supports the streamlined ICAP interface to leverage Deep Packet Inspection to enable external application servers to provide their services without performing DPI, and without being inserted in the data flow. For example, with an external Active Content Filtering (ACF) platform.

A high-level view of the streamlined ICAP interface support for external ACF is shown in the following figure.

Figure 4. High-Level View of Streamlined ICAP Interface with External ACF



The system with ECS is configured to support DPI and the system uses this capability for content charging as well.

If a subscriber initiates a WAP (WAP1.x or WAP2.0) or Web session, the subsequent GET/POST request is detected by the DPI function. The URL of the GET/POST request is extracted and passed, along with subscriber identification information and the subscriber request, in an ICAP message to the application server.

In the case of Category-based Content Filtering solution, the application server checks the URL on the basis of its category and other classifications like type, access level and content category and decides if the request should be authorized, blocked, or redirected by answering to the GET/POST with:

- A 200 OK message if the request is accepted.
- A 302 Redirect message in case of redirection. This redirect message includes the URL to which the subscriber should be redirected.

- A 403 Denied message is the request should be blocked.

Depending on the response received, the system with ECS will either pass the request unmodified, or discard the message, and respond to the subscriber with the appropriate redirection or block message.

Content Charging is performed by the ECS only after the request has been controlled by the application server. This guarantees the appropriate interworking between the external application and content-based billing. In particular, this guarantees that charging will be applied to the appropriate request in case of redirection, and that potential charging based redirections (i.e. Advice of Charge, Top Up page, etc.) will not interfere with the decisions taken by the application server.

The ACF performs the following functions:

- Retrieval of subscriber policies based on the subscriber identity passed in the ICAP message.
- Determining the appropriate action (permit, deny, redirect) to take for this type of content based on subscriber profile.
- Communication of the action (permit, deny, or redirect) decision for the URL back to the ECS subsystem.

For information on configuring the ICAP interface support for external ACF servers, refer to the *ICAP Interface Support* chapter of the *System Enhanced Feature Configuration Guide*.

External Storage System

ESS supports generation of EDR/UDR/FDR (xDR) files from the chassis. To store generated xDR files, on the ASR 5000 chassis, the system allocates 512 MB of memory on the packet processing card's RAM. The generated xDRs are stored in CSV format in the */records* directory on the packet processing card RAM. These generated xDRs can be used for billing as well as for generation of reports to analyze network usage and subscriber trends. As this temporary storage space (size configurable) reaches its limit, the system deletes older xDRs to make room for new xDRs. Setting gzip file compression extends the storage capacity by approximately 10:1.

Because of the volatile nature of the memory, xDRs can be lost due to overwriting, deletion, or unforeseen events such as power or network failure or unplanned chassis switchover. To avoid loosing charging and network analysis information, configure the CDR subsystem in conjunction with the External Storage System (ESS) to offload the xDRs for storage and analysis.

For more information on the ESS, refer to the *ESS Installation and Administration Guide*.

Minimum System Requirements and Recommendations

This section identifies the minimum system requirements for components of the URL Blacklisting / Category-based Content Filtering solutions.

IMPORTANT: The hardware required for these components may vary, depending on the number of clients that require access, components managed, and other variables like EDR generation rate or CDR storage and processing requirements.

Certain basic server requirements are recommended for WEM and inPilot to exploit the CF solution. For information on these system requirements, refer to *WEM Installation and Administration Guide* and *inPilot Installation and Administration Guide*.

MCRDBS System Requirements

This section provides information on the system requirements for MCRDBS.

IMPORTANT: You must ensure that the minimum system requirements are met before proceeding with the MCRDBS installation.

Hardware Requirements

- Dell PowerEdge 1950 server:
  - 1.86 GHz Dual quad-core Intel Xeon CPU
  - 8 GB RAM
  - 2 * 146 GB RAID hard disk drive. The hard disk can be expanded up to 300 GB.
  - Gigabit Ethernet interfaces
  - CD-ROM Drive
- Operating Environment:
  - Debian Linux with all recommended patches from vendor

Additional Requirements on Chassis

The chassis requires the following additional hardware and memory to handle the Content Rating Master Databases; for example, for Category-based Content Filtering OPTCMDB. The memory required may vary with the size of rating databases used for content rating service.

- Minimum of two active packet processing cards s are required
- Minimum 4 GB memory:
    - in ASR5000 on Flash memory

Chapter 2
Content Filtering Service Configuration

This chapter describes how to configure content filtering support with ECS.

In this chapter, only the minimum set of configurations required to make the system operational with content filtering services are provided. Additional configuration commands specific to the content filtering service are available in the *Command Line Interface Reference*.

The following topics are described in this chapter:

Configuring the System for Content Filtering Support

This section lists the high-level steps to configure a system with Content Filtering service in conjunction with the Enhanced Charging Services.

CAUTION: Before proceeding with the configuration, refer the *Additional Requirements on Chassis for Content Filtering* section of the *Content Filtering Support Overview* chapter for the minimum system requirements. If the system has fewer than two processing cards, Content Filtering service cannot be activated on the system.

To configure the system for Content Filtering service:

Step 1 Set the initial configuration parameters such as activating the processing cards and creating the VPN context by applying the example configurations in the Initial Configuration section.

Step 2 Enable the Enhanced Charging Service with Content Filtering, and configure Content Filtering parameters:

- For URL Blacklisting support, enable the Enhanced Charging Service by applying the example configurations presented in the URL Blacklisting Configuration section.

–and/or–

- For Category-based Content Filtering support, enable the Enhanced Charging Service by applying the example configurations presented in the Category-based Content Filtering Configuration section.

Step 3 Save the changes to system configuration by applying the example configuration in the Saving the Configuration section.

Initial Configuration

Step 1 Specify the role of the processing cards in the chassis by applying the example configuration in the Activating Processing Cards section.

Step 2 Set local system management parameters by applying the example configuration in the Modifying the Local Context section.

Step 3 Create the context where the service will reside by applying the example configuration in the Creating the VPN Context section.

Step 4 Create the service within the newly created context by applying the example configuration in the *Service Configuration* chapter of the *System Administration Guide*.

Activating Processing Cards

The following example activates two processing cards, placing one in active mode and labeling the other as redundant:

```
configure
   card <slot_number>
      redundancy card-mode
      exit
   card <slot_number>
      mode active pac
      end
```

Modifying the Local Context

The following example sets the default subscriber in the local context:

```
configure
   context local
      interface <local_ctx_iface_name>
         p address <ip_address> <ip_mask>
         exit
      server ftpd
         exit
      server telnetd
         exit
      subscriber default
         exit
      administrator <name> encrypted password <password> ftp
      ip route <ip_addr> <ip_mask> <next_hop_addr> <local_ctx_iface_name>
      exit
   port ethernet <slot#/port#>
      no shutdown
      bind interface <local_ctx_iface_name> local
```

```
        exit
    end
```

Creating the VPN Context

The following example creates the VPN context and interface and binds the VPN interface to a configured Ethernet port.

```
configure
    context <vpn_context_name> -noconfirm
        interface <vpn_interface_name>
            ip address <ip_address> <ip_mask>
            exit
        subscriber default
            exit
        ip route 0.0.0.0 0.0.0.0 <next_hop_address> <vpn_interface_name>
        exit
    port ethernet <slot_number/port_number>
        no shutdown
        bind interface <vpn_interface_name> <vpn_context_name>
        end
```

URL Blacklisting Configuration

This section describes steps to configure the system for URL Blacklisting support.

Step 1 Enable the ACS subsystem by applying the example configuration in the Enabling ACS Subsystem section.

Step 2 Configure URL Blacklisting database parameters by applying the example configuration in the Configuring URL Blacklisting Database Parameters section.

Step 3 Create the Active Charging Service, and set URL Blacklisting matching method by applying the example configuration in the Creating Active Charging Service and Setting URL Blacklisting Matching section.

Step 4 Enable URL Blacklisting functionality in a rulebase, and configure the action to be taken by applying the example configuration in the Enabling URL Blacklisting in Rulebase and Configuring Blacklisting Action section.

Step 5 Load/upgrade URL Blacklisting database by applying the example configuration in the LoadingUpgrading URL Blacklisting Database section.

Enabling ACS Subsystem

Use the following configuration to enable the Active Charging Service subsystem for URL Blacklisting:

```
configure
   require active-charging
   end
```

Configuring URL Blacklisting Database Parameters

Use the following configuration to configure URL Blacklisting database parameters:

```
configure
   url-blacklisting database directory path <directory_path>
   url-blacklisting database max-versions <max_versions>
   url-blacklisting database override file <file.extension>
   end
```

Creating Active Charging Service and Setting URL Blacklisting Matching

Use the following configuration to create the Active Charging Service and set URL Blacklisting match:

```
configure
   active-charging service <service_name> [ -noconfirm ]
      url-blacklisting match-method { exact | generic }
      end
```

Enabling URL Blacklisting in Rulebase and Configuring Blacklisting Action

Use the following configuration to enable URL Blacklisting in a rulebase and configure the blacklisting action:

```
configure
   active-charging service <service_name>
      rulebase <rulebase_name> [ -noconfirm ]
         url-blacklisting action { discard | redirect-url <url> | terminate-flow
| www-reply-code-and-terminate-flow <reply_code> }
         end
```

Loading/Upgrading URL Blacklisting Database

Use the following command to load/upgrade the URL Blacklisting database:

```
upgrade url-blacklisting database [ -noconfirm ]
```

Testing URL Blacklisting Functionality

The URL Blacklisting functionality can be tested by appending test URLs/URIs to the blacklist file. The test URLs/URIs must be added to the *testurldb.pub* file in the *<WEM_Install_Dir>/flash/blacklist/testurldb* directory.

The *testurldb.pub* file must have one URL per line without space. If space is included in the URL entries, the WEM ignores the URLs with space.

Category-based Content Filtering Configuration

This section describes the steps to configure the system for Category-based Content Filtering support.

Step 1 Enable the Enhanced Charging mode for Category-based Static by applying the example configuration in the Enabling ACS Subsystem section.

Step 2 Configure the global parameters like database path and version for Content Filtering service by applying the example configuration in the Configuring Content Rating Rule Database Parameters section. This is an optional step. In case this configuration is not performed, the default values will be used.

Step 3 Create the Active Charging Service and Content Filtering Policy by applying the example configuration in the Creating Active Charging Service and Content Filtering Policy section.

Step 4 Configure the Content Filtering Policy Identifier and actions by applying the example configuration in the Configuring Content Filtering Policy section.

Step 5 *Optional*. Create billing and charging actions by applying the example configuration in the *Configuring Enhanced Charging Services* chapter of the *Enhanced Charging Services Administration Guide*.

Step 6 *Optional*. Define rule definitions by applying the example configuration in the *Configuring Enhanced Charging Services* chapter of the *Enhanced Charging Services Administration Guide*.

Step 7 Create and configure the rulebases by applying the example configuration in the Configuring Rulebase for Content Filtering section. For more information on rulebase configuration, refer to the *ECS Configuration* chapter in the *Enhanced Charging Services Administration Guide*.

Step 8 Apply the Content Filtering service to subscribers/APNs by applying the example configuration in the APN Configuration /Subscriber Configuration section.

Step 9 Create the EDR format and configure attributes by applying the example configurations in the Configuring Event Detail Record (EDR) section.

IMPORTANT: Category-based Static-and-Dynamic Content Filtering is not supported in this release.

Enabling ACS Subsystem

Use the following configuration to enable the Active Charging Service subsystem:

```
configure
   require active-charging content-filtering category
   end
```

Configuring Content Rating Rule Database Parameters

Use the following configuration to configure Content Rating Rule database parameters:

```
configure
   content-filtering category database directory path <directory_path>
   content-filtering category database max-versions <max_versions>
   content-filtering category database override file <file.extension>
   end
upgrade content-filtering category database
```

Creating Active Charging Service and Content Filtering Policy

Use the following configuration to create the Active Charging Service and Content Filtering Policy:

```
configure

   active-charging service <service_name> [ -noconfirm ]

      content-filtering category policy-id <cf_policy_id> [ description
<description> ] [ -noconfirm ]

      end
```

Configuring Content Filtering Policy

Use the following configuration to configure the content filtering policy:

```
configure

   active-charging service <service_name>

      content-filtering category policy-id <cf_policy_id>

         analyze priority <priority> { all | category <category> | x-category
<x-category> } action { allow | content-insert <content_string> | discard |
redirect-url <url> | terminate-flow | www-reply-code-and-terminate-flow
<reply_code> } [ edr <edr_format> ]

         failure-action { allow | content-insert <content_string> | discard |
redirect-url <url> | terminate-flow | www-reply-code-and-terminate-flow
<reply_code> } [ edr <edr_format> ]

         end
```

Notes:

- To configure runtime categories not present in the CLI, use the following command:

  analyze priority *<priority>* **x-category** *<x-category>* **action { allow | content-insert** *<content_string>* **| discard | redirect-url** *<url>* **| terminate-flow | www-reply-code-and-terminate-flow***<reply_code>* **} [edr** *<edr_format>* **]**

- To configure the action to take for any match, and the default action to take when the category returned after rating is not configured in the subscriber's content filtering policy, use the following command:

  analyze priority *<priority>* **all action { allow | content-insert** *<content_string>* **| discard | redirect-url** *<url>* **| terminate-flow | www-reply-code-and-terminate-flow** *<reply_code>* **} [edr** *<edr_format>* **]**

Configuring Rulebase for Content Filtering

Use the following configuration to configure the rulebase:

```
configure
   active-charging service <service_name>
      rulebase <rulebase_name>
         route priority <route_priority> ruledef <ruledef_name> analyzer
<analyzer_name> [ description <description> ]
         action priority <priority> { [ dynamic-only | static-and-dynamic ] {
group-of-ruledefs <group_name> | ruledef <ruledef_name> } charging-action
<charging_action_name> [ description <description> ] }
         flow end-condition content-filtering edr <edr_format_name>
         billing-records { egcdr | radius | udr udr-format <format_name> }+
         content-filtering category policy-id <cf_policy_id>
         content-filtering mode category static-only
         end
```

Enabling Category-based Content Filtering Support

APN Configuration

Use the following configuration to apply Content Filtering configuration to an APN through policy identifier:

```
configure
   context <context_name>
      apn <apn_name>
         content-filtering category policy-id <cf_policy_id>
         end
```

Subscriber Configuration

Use the following configuration to apply Content Filtering configuration to a subscriber through policy identifier:

```
configure
   context <context_name>
      subscriber name <user_name>
```

```
content-filtering category policy-id <cf_policy_id>
end
```

IMPORTANT: Category Policy ID applied to APN or subscriber in this mode overrides the Category Policy ID configured using the "**content-filtering category policy-id** *cf_policy_id*" command in the Configuring Rulebase for Content Filtering section.

Configuring Event Detail Record (EDR)

This section describes how to configure Category-based Content Filtering EDR settings. The system does not generate URL Blacklisting specific EDRs.

To configure Category-based Content Filtering EDR settings:

Step 1 Enable the EDR module and file format for EDR in context configuration mode by applying the example configuration in the EDR Module Configuration section.

Step 2 Define attributes and rule variables by applying the example configuration in the EDR Attribute Configuration section.

Step 3 *Optional*. Enable charging record retrieval by applying the example configuration in the *Charging Record Retrieval* section of *Enhanced Charging Services Administration Guide*.

EDR Module Configuration

Use the following configuration to enable EDR module and configure the file for EDR generation in Content Filtering services:

```
configure
   context <context_name>
      edr-module active-charging-service
         file [ edr-format-name ] [ name <file_name> ]+
         end
```

Notes:

- For more information on keywords/options available with the **file** command, refer to the *EDR Module Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

EDR Attribute Configuration

Use the following configuration to configure attributes and rule-variables for EDRs for Content Filtering services:

```
configure
   active-charging service <service_name>
```

```
edr-format <edr_format_name>
   attribute <attribute> priority <priority>
   rule-variable <protocol> <rule> priority <priority>
   end
```

Notes:

- For more information on options available with **attribute** and **rule-variable** commands, refer to the *EDR Format Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Saving the Configuration

To save the changes made to the system configuration for Content Filtering service, refer to the *Verifying and Saving Your Configuration* chapter.

Verifying the Configuration

This section describes how to review the configurations after saving them in a *.cfg* file as described in the *Verifying and Saving Your Configuration* chapter, and to retrieve errors and warnings within an active configuration for a service.

Viewing System Configuration

Use the following configuration to view the active configuration for a service:

```
configure
   context <context_name>
   end
show configuration
```

Viewing Service Configuration Errors

Use the following configuration to view the errors in configuration for a service:

```
configure
   context <context_name>
   end
show configuration errors verbose
```

This command also shows the ambiguities in configurations with Content Filtering service, category, and rulebase configuration. Warnings/errors are displayed in the following scenarios:

- Warning: When "**`require active-charging content-filtering category`**" CLI command is not activated and any Content Filtering configurations are done.
- Error: When Content Filtering is enabled, but no Content Filtering Policy ID is configured in the Active Charging Service.
- Error: A rulebase uses an undefined Content Filtering Policy ID.
- Error: A rulebase has Content Filtering Category Mode set, but Content Filtering Policy ID is not set.
- Warning: A rulebase has Content Filtering Policy ID set, but Content Filtering Category Mode is not set.
- Error: An APN uses a Content Filtering Policy ID not defined in the Active Charging Service.
- Error: A subscriber uses a Content Filtering Policy ID not defined in the Active Charging Service.

- Warning: When no default analyze rule is configured in Content Filtering Policy ID.
- Warning: When default analyze rule is configured in the Content Filtering Policy ID, but not at the lowest priority.
- Warning: When no analyze rule is configured in Content Filtering Policy ID.

Gathering Statistics

This section explains how to gather statistics and configuration information for:

- URL Blacklisting Statistics
- Category-based Content Filtering Statistics

URL Blacklisting Statistics

This section explains how to gather URL Blacklisting statistics and configuration information.

In the following table, the first column lists what statistics to gather, the second column lists the action to perform, and the third column describes what information is displayed or what information to look for in the resulting output.

Table 1. Gathering URL Blacklisting Statistics and Configuration Information

Statistics Wanted	Action to Perform
To view URL Blacklisting statistics, optionally for rulebase(s)	**show active-charging url-blacklisting statistics [rulebase { all | name** *<rulebase_name>* **}] [verbose] [| { grep** *<grep_options>* **| more }]**
To view URL Blacklisting static database configuration	**show url-blacklisting database [all | url** *<url>* **| facility acsmgr { all | instance** *<instance>* **}] [| { grep** *<grep_options>* **| more }]**
To view total Blacklisting URL hits and misses statistics, optionally for rulebase(s) or specific ACS instance	**show active-charging subsystem { all | facility acsmgr [all | instance** *<instance>* **] | full } | [rulebase name** *<rulebase_name>* **] | [| { grep** *<grep_options>* **| more }]**
To view information for rulebase(s) configured in a system or service	**show active-charging rulebase { all [service name** *<svc_name>* **] | name** *<rulebase_name>* **[service name** *<svc_name>* **] | statistics [name** *<rulebase_name>* **] } | [| { grep** *<grep_options>* **| more }]**
To view ACS session statistics	**show active-charging sessions all [| { grep** *<grep_options>* **| more }]**

Category-based Content Filtering Statistics

This section explains how to gather Category-based Content Filtering statistics and configuration information.

In the following table, the first column lists what statistics to gather, the second column lists the action to perform, and the third column describes what information is displayed or what information to look for in the resulting output.

IMPORTANT: For more information on Content Filtering statistics collection, refer to the *Exec Mode Commands* chapter of the *Command Line Interface Reference*.

Table 2. Gathering Category-based Content Filtering Statistics and Configuration Information

Statistics Wanted	Action to Perform
To view Category-based Content Filtering database statistics/configuration	**show content-filtering category database [active | all | facility srdbmgr { all | instance** *<instance>* **} | url** *<url_string>* **] [verbose] [| { grep** *<grep_options>* **| more }]**
To view Category-based Content Filtering category statistics	**show content-filtering category statistics [facility srdbmgr { all | instance** *<instance>* **}] [| { grep** *<grep_options>* **| more }]**
To view information of a database URL for Category-based Content Filtering application in a service	**show content-filtering category url** *<url_string>* **[policy-id** *<cf_policy_id>* **| rulebase** *<rulebase_name>* **] [verbose] [| { grep** *<grep_options>* **| more }]**
To view Content Filtering Server Group (CFSG) details configured in the service	**show content-filtering server-group [statistics] [name** *<cfsg_name>* **] [| { grep** *<grep_options>* **| more }]**
To view Category-based Content Filtering category policy definitions	**show active-charging content-filtering category policy-id { all | id** *<policy_id>* **} [| { grep** *<grep_options>* **| more }]**
To view Category-based Content Filtering statistics, optionally for rulebase(s)	**show active-charging content-filtering category statistics [rulebase { name** *<rulebase_name>* **| all }] [verbose] [| { grep** *<grep_options>* **| more }]**
To view details of Content Filtering Server Group (CFSG) configured in the service	**show active-charging content-filtering server-group [statistics [verbose]] [name** *<cfsg_name>* **] [| { grep** *<grep_options>* **| more }]**
To view information for rulebase(s) configured in a system or service	**show active-charging rulebase { all [service name** *<svc_name>* **] | name** *<rulebase_name>* **[service name** *<svc_name>* **] | statistics [name** *<rulebase_name>* **] } | [| { grep** *<grep_options>* **| more }]**
To view Active Charging Service session statistics	**show active-charging sessions all [| { grep** *<grep_options>* **| more }]**

Supported Bulk Statistics

For information on bulk statistics configuration and collection, and the list of bulk statistics for the Content Filtering service, refer to the *Bulk Statistics Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Supported Thresholds and SNMP Traps

For information on the SNMP traps and thresholds for the Content Filtering service, see the *Content Filtering Application MIB* chapter of the *SNMP MIB Reference*.

Chapter 3
Verifying and Saving Your Configuration

This chapter describes how to save the system configuration.

Verifying the Configuration

You can use a number of command to verify the configuration of your feature, service, or system. Many are hierarchical in their implementation and some are specific to portions of or specific lines in the configuration file.

Feature Configuration

In many configurations, specific features are set and need to be verified. Examples include APN and IP address pool configuration. Using these examples, enter the following commands to verify proper feature configuration:

show apn all

The output displays the complete configuration for the APN. In this example, an APN called apn1 is configured.

```
access point name (APN): apn1

authentication context: test

pdp type: ipv4

Selection Mode: subscribed

ip source violation: Checked drop limit: 10

accounting mode: gtpp No early PDUs: Disabled

max-primary-pdp-contexts: 1000000 total-pdp-contexts: 1000000

primary contexts: not available total contexts: not available

local ip: 0.0.0.0

primary dns: 0.0.0.0 secondary dns: 0.0.0.0

ppp keep alive period : 0 ppp mtu : 1500

absolute timeout : 0 idle timeout : 0

long duration timeout: 0 long duration action: Detection

ip header compression: vj

data compression: stac mppc deflate compression mode: normal

min compression size: 128

ip output access-group: ip input access-group:

ppp authentication:

allow noauthentication: Enabled imsi

 authentication:Disabled
```

Enter the following command to display the IP address pool configuration:

show ip pool

The output from this command should look similar to the sample shown below. In this example, all IP pools were configured in the *isp1* context.

```
context : isp1:

+-----Type: (P) - Public (R) - Private

 | (S) - Static (E) - Resource

 |

 |+----State: (G) - Good (D) - Pending Delete (R)-Resizing

 ||

 ||++--Priority: 0..10 (Highest (0) .. Lowest (10))

 ||||

 ||||+-Busyout: (B) - Busyout configured

 |||| |||||| vvvvv Pool Name Start Address Mask/End Address Used Avail

 ----- --------- --------------- --------------- -------- --------

PG00 ipsec 12.12.12.0 255.255.255.0 0 254 PG00

pool1 10.10.0.0 255.255.0.0 0 65534 SG00

vpnpool 192.168.1.250 192.168.1.254 0 5 Total Pool Count: 5
```

IMPORTANT: Many features can be configured on the system. There are show commands specifically for these features. Refer to the *Command Line Interface Reference* for more information.

Service Configuration

Verify that your service was created and configured properly by entering the following command:

show *<service_type> <service_name>*

The output is a concise listing of the service parameter settings similar to the sample displayed below. In this example, a P-GW service called pgw is configured.

```
Service name : pgw1

Service-Id : 1

Context : test1
```

```
Status : STARTED

Restart Counter : 8

EGTP Service : egtp1

LMA Service : Not defined

Session-Delete-Delay Timer : Enabled

Session-Delete-Delay timeout : 10000(msecs)

PLMN ID List : MCC: 100, MNC: 99

Newcall Policy : None
```

Context Configuration

Verify that your context was created and configured properly by entering the following command:

show context name *<name>*

The output shows the active context. Its ID is similar to the sample displayed below. In this example, a context named *test1* is configured.

```
Context Name     ContextID     State

------------     ---------     -----

test1            2          Active
```

System Configuration

Verify that your entire configuration file was created and configured properly by entering the following command:

show configuration

This command displays the entire configuration including the context and service configurations defined above.

Finding Configuration Errors

Identify errors in your configuration file by entering the following command:

show configuration errors

This command displays errors it finds within the configuration. For example, if you have created a service named “service1”, but entered it as “srv1” in another part of the configuration, the system displays this error.

You must refine this command to specify particular sections of the configuration. Add the **section** keyword and choose a section from the help menu:

show configuration errors section ggsn-service

or

show configuration errors section aaa-config

If the configuration contains no errors, an output similar to the following is displayed:

```
##############################################################################

Displaying Global
AAA-configuration errors
##############################################################################

Total 0 error(s) in this section !
```

Saving the Configuration

Save system configuration information to a file locally or to a remote node on the network. You can use this configuration file on any other systems that require the same configuration.

Files saved locally can be stored in the SPC's/SMC's CompactFlash or on an installed PCMCIA memory card on the SPC/SMC. Files that are saved to a remote network node can be transmitted using either FTP, or TFTP.

Saving the Configuration on the Chassis

These instructions assume that you are at the root prompt for the Exec mode:

```
[local]host_name#
```

To save your current configuration, enter the following command:

```
save configuration url [-redundant] [-noconfirm] [showsecrets] [verbose]
```

Keyword/Variable	Description
url	Specifies the path and name to which the configuration file is to be stored. *url* may refer to a local or a remote file. *url* must be entered using one of the following formats: • **{ /flash | /pcmcia1 | /pcmcia2 }** *[/dir] /file_name* • **file:/{ /flash | /pcmcia1 | /pcmcia2 }** *[/dir] /file_name* • **tftp://**{ *ipaddress*|*host_name*[*:port#*]}[*/directory*]*/file_name* • **ftp://**[*username*[*:pwd*] *@*]{ *ipaddress*|*host_name*}[*:port#*][*/directory*] */file_name* • **sftp://**[*username*[*:pwd*] *@*]{ *ipaddress*|*host_name*}[*:port#*][*/directory*] */file_name* **/flash** corresponds to the CompactFlash on the SPC/SMC. **/pcmcia1** corresponds to PCMCIA slot 1. **/pcmcia2** corresponds to PCMCIA slot 2. *ipaddress* is the IP address of the network server. *host_name* is the network server's *hostname*. *port#* is the network server's logical port number. Defaults are: • tftp: 69 - data • ftp: 20 - data, 21 - control • sftp: 115 - data Note: *host_name* can only be used if the **networkconfig** parameter is configured for DHCP and the DHCP server returns a valid nameserv er.dx *username* is the username required to gain access to the server if necessary. *password* is the password for the specified username if required. */directory* specifies the directory where the file is located if one exists. */file_name* specifies the name of the configuration file to be saved. Note: Configuration files should be named with a .cfg extension.
-redundant	Optional: This keyword directs the system to save the CLI configuration file to the local device, defined by the url variable, and then automatically copy that same file to the like device on the Standby SPC/SMC, if available. Note: This keyword will only work for like local devices that are located on both the active and standby SPCs/SMCs. For example, if you save the file to the /pcmcia1 device on the active SPC/SMC, that same type of device (a PC-Card in Slot 1 of the standby SPC/SMC) must be available. Otherwise, a failure message is displayed. Note: If saving the file to an external network (non-local) device, the system disregards this keyword.

Keyword/Variable	Description
-noconfirm	Optional: Indicates that no confirmation is to be given prior to saving the configuration information to the specified filename (if one was specified) or to the currently active configuration file (if none was specified).
showsecrets	Optional: This keyword causes the CLI configuration file to be saved with all passwords in plain text, rather than their default encrypted format.
verbose	Optional: Specifies that every parameter that is being saved to the new configuration file should be displayed.

IMPORTANT: The **`-redundant`** keyword is only applicable when saving a configuration file to local devices .This command does not synchronize the local file system. If you have added, modified, or deleted other files or directories to or from a local device for the active SPC/SMC, then you must synchronize the local file system on both SPCs/SMCs.

To save a configuration file called system.cfg to a directory that was previously created called cfgfiles on the SPC's/SMC's CompactFlash, enter the following command:

```
save configuration /flash/cfgfiles/system.cfg
```

To save a configuration file called simple_ip.cfg to a directory called host_name_configs using an FTP server with an IP address of 192.168.34.156 on which you have an account with a username of administrator and a password of secure, use the following command:

```
save configuration
ftp://administrator:secure@192.168.34.156/host_name_configs/
simple_ip.cfg
```

To save a configuration file called init_config.cfg to the root directory of a TFTP server with a hostname of config_server, enter the following command:

`save configuration` *`tftp://config_server/init_config.cfg`*

Appendix A
Category List

The Category-based Content Filtering solution uses categories to categorize content and URLs for content filtering.

The following table lists the category codes used in the Category-based Content Filtering application. Apart from these categories, in the Content Filtering Policy Configuration Mode, runtime categories not present in the CLI can also be configured for rating.

Table 3. Category Codes and Descriptions

Category	Description
ABOR	Abortion
ADULT	Adult Related Material
ADVERT	Advertising site
ANON	Anonymizer
ART	Art, Museums
AUTO	Automotive site
BLACK	Inappropriate Content
BLOG	Blogging
BUSI	Business
CAR	Career, Job Search
CHAT	Chatting site
CMC	Virtual Community
CRIME	Criminal Skills
CULT	Cult
DRUG	Drug
DYNAM	Dynamic site, Wireless
EDU	Educational site
ENERGY	Energy
ENT	Entertainment, Music
FIN	Finance
FORUM	Forum and Messageboard
GAMB	Gambling

Category	Description
GAME	Gaming
GLAM	Glamour
GOVERN	Government site
HACK	Hacking
HATE	Hate Site, Hate Speech
HEALTH	Health site
HOBBY	Hobby
HOSTS	Hosting site
KIDS	Kids
LEGAL	Legal, Law site
LIFES	Lifestyle
MAIL	Webmail
MIL	Military
NEWS	News site
OCCULT	Occult
PEER	File Sharing
PERS	Personals and Dating
POLTIC	Politics
PORN	Pornography, Nudism, Naturism
PORTAL	Portal site
PROXY	Proxy, Test
REF	Reference site
REL	Religion
SCI	Science
SEARCH	Search, Web Search
SHOP	Online Shopping
SPORT	Sport
STREAM	Streaming Media
SUIC	Suicide
SXED	Sexual Education
TECH	Technology and Telecommunication
TRAV	Travel

Category	Description
VIOL	Violence
VOIP	Internet Telephony
WEAP	Weapons
WHITE	Clean Content
UNKNOW	Unknown URLs which are not present in the optimized content category database

Appendix B
Sample Content Filtering Service Configuration

This appendix includes the following sample configuration files for Content Filtering configuration within an ECS service:

- URL Blacklisting Configuration
- Category-based Content Filtering Configuration

URL Blacklisting Configuration

This section presents a sample configuration file with URL Blacklisting configuration within an ECS service.

```
config
   license key "\
VER=1|C1M=SanDiskSDCFJ-4096|C1S=016816D2597X4624|C2M=SanDiskSDCFJ-4096\
FAA=Y|FCP=Y|LCF=30000|SIG=MC0CFQC2Zp+qSGqGR+VQ5QdhkHksZgXxgAIUN7+bT/OL\
qeFwAMiJbb4acy33JsU"
   aaa large-configuration
   timestamps
   autoconfirm
   clock timezone asia-calcutta
   crash enable encrypted url 01abc234d56e7f8g01abc234d56e7f8g
   card 1
      mode active psc
   #exit
   card 2
      mode active psc
   #exit
   card 3
      mode active psc
   #exit
   require session recovery
   require active-charging
   require diameter-proxy multiple
   context local
      interface spio
```

```
            ip address 1.2.3.4 255.255.255.0
         #exit
         server ftpd
         #exit
         ssh key
f22330a765e10f40001920bf01dbf89a224dd8f09fe8d1598751401cb392f3c062f859a4335cb92f
4a352a4686dcea99e4740be8a0063da1c657c560991ec87ce06728 len 461
         server sshd
            subsystem sftp
         #exit
         server telnetd
         #exit
         server tftpd
         #exit
         subscriber default
         exit
         administrator administrator encrypted password 123abc456def789gh ftp
         aaa group default
         #exit
         gtpp group default
         #exit
         ip route 0.0.0.0 0.0.0.0 1.2.3.4 spio
         ip domain-lookup
         ip domain-name ind.star.com
         ip name-servers 1.2.3.4
      #exit
      port ethernet 24/1
         no shutdown
         bind interface spio local
      #exit
```

```
ntp
   enable
   server 1.2.3.4
#exit
snmp community private read-only
snmp community public read-only
snmp target abc1 1.2.3.4 port 162 security-name public version 2c traps
active-charging service bl_service
   ruledef clwap-dst
      udp dst-port = 9200
      rule-application routing
   #exit
   ruledef clwap-src
      udp src-port = 9200
      rule-application routing
   #exit
   ruledef cowap-dst
      udp dst-port = 9201
      rule-application routing
   #exit
   ruledef cowap-src
      udp src-port = 9201
      rule-application routing
   #exit
   ruledef default
      ip any-match = TRUE
   #exit
   ruledef ftp-ctrl-dst
      tcp dst-port = 21
```

```
   rule-application routing
#exit
ruledef ftp-ctrl-src
   tcp src-port = 21
   rule-application routing
#exit
ruledef ftp-data-dst
   tcp dst-port = 20
   rule-application routing
#exit
ruledef ftp-data-src
   tcp src-port = 20
   rule-application routing
#exit
ruledef handshake
   tcp payload-length = 0
   tcp any-match = TRUE
   tcp flag !contains fin
   tcp flag !contains reset
#exit
ruledef http-dst
   tcp dst-port = 80
   rule-application routing
#exit
ruledef http-get
   http request method = get
#exit
ruledef http-pkts
   http any-match = TRUE
```

```
#exit
ruledef http-proxy-dst
   tcp dst-port = 3128
   rule-application routing
#exit
ruledef http-proxy-src
   tcp src-port = 3128
   rule-application routing
#exit
ruledef http-route
   tcp either-port = 80
   rule-application routing
#exit
ruledef http-src
   tcp src-port = 80
#exit
ruledef http-wap2-dst
   tcp dst-port = 8799
   rule-application routing
#exit
ruledef http-wap2-src
   tcp src-port = 8799
   rule-application routing
#exit
ruledef https-dst
   tcp dst-port = 443
   rule-application routing
#exit
ruledef https-src
```

```
   tcp src-port = 443
   rule-application routing
#exit
ruledef pop3-dst
   tcp dst-port = 110
   rule-application routing
#exit
ruledef pop3-src
   tcp src-port = 110
   rule-application routing
#exit
ruledef rtsp-dst
   tcp dst-port = 554
   rule-application routing
#exit
ruledef rtsp-src
   tcp src-port = 554
   rule-application routing
#exit
ruledef rule2
   http uri starts-with http://1.2.3.4/test/service/2000/
#exit
ruledef rule3
   http uri starts-with http://1.2.3.4/test/service/3000/
#exit
ruledef rule4
   http uri starts-with http://1.2.3.4/test/service/4000/
#exit
ruledef rule5
```

```
   http uri starts-with http://1.2.3.4/test/service/5000/
#exit
ruledef rule6
   http uri starts-with http://1.2.3.4/test/service/6000/
#exit
ruledef rule7
   http uri starts-with http://1.2.3.4/test/service/7000/
#exit
ruledef rule8
   http uri starts-with http://1.2.3.4/test/service/8000/
#exit
ruledef rule9
   http uri starts-with http://1.2.3.4/test/service/9000/
#exit
ruledef sdp_route
   sip content type = application/sdp
   rule-application routing
#exit
ruledef sip-dst
   udp dst-port = 5060
   rule-application routing
#exit
ruledef sip-src
   udp src-port = 5060
   rule-application routing
#exit
ruledef smtp-dst
   tcp dst-port = 25
   rule-application routing
```

```
      #exit
      ruledef smtp-src
         tcp src-port = 25
         rule-application routing
      #exit
      ruledef tcp
         ip protocol = 6
         rule-application routing
      #exit
      ruledef udp
         ip protocol = 17
         rule-application routing
      #exit
      charging-action standard
         content-id 10
         retransmissions-counted
      #exit
      url-blacklisting method exact-match
      rulebase rulebase1
         action priority 1 ruledef http-get charging-action standard
         action priority 65000 ruledef default charging-action standard
         url-blacklisting action discard
         route priority 80 ruledef http-route analyzer http
         no transport-layer-checksum verify-during-packet-inspection
      #exit
      rulebase default
      #exit
   #exit
   context source
```

```
interface chassis1_2_CLIENT
   ip address 1.2.3.4 255.255.255.0
   ip address 1.2.3.5 255.255.255.255 secondary
   ip address 1.2.3.6 255.255.255.255 secondary
#exit
interface chassis1_2_RADIUS
   ip address 1.2.3.4 255.255.255.0
#exit
subscriber default
   ip access-group acl1 in
   ip access-group acl1 out
   ip context-name dest
   active-charging rulebase rulebase1
exit
aaa group default
   radius attribute nas-ip-address address 1.2.3.4
   radius server 1.2.3.4 encrypted key 01abc234d56e7f8g port 1812
   radius accounting server 1.2.3.4 encrypted key 01abc234d port 1813
#exit
gtpp group default
#exit
ha-service HA         mn-ha-spi spi-number 1000 encrypted secret
01abc234d56e7f8g hash-algorithm md5
   fa-ha-spi remote-address 1.2.3.4 spi-number 256 encrypted secret
01abc234d56e7f8g hash-algorithm md5
   fa-ha-spi remote-address 1.2.3.4 spi-number 256 encrypted secret
01abc234d56e7f8g hash-algorithm md5
   no reg-lifetime
   bind address 1.2.3.4
#exit
```

```
      edr-module active-charging-service
      #exit
      ip igmp profile default
      #exit
   #exit
   context dest
      ip access-list acl1
         redirect css service srv1   any
      #exit
      ip pool callgen_A11 1.2.3.4 255.255.0.0 static
      ip pool callgen_B11 1.2.3.5 255.255.0.0 static
      ip pool dpool00 1.2.3.6 255.255.0.0 public 0
      ip pool dpool01 1.2.3.7 255.255.0.0 public 0
      interface chassis1_2_SERVER
         ip address 1.2.3.4 255.255.255.0
      #exit
      subscriber default
      exit
      aaa group default
      #exit
      gtpp group default
      #exit
      ip igmp profile default
      #exit
   #exit
   port ethernet 17/1
      no shutdown
      vlan 4000
         no shutdown
```

```
      bind interface chassis1_2_SERVER dest
   #exit
#exit
port ethernet 18/1
   no shutdown
   vlan 2000
      no shutdown
      bind interface chassis1_2_CLIENT source
   #exit
   vlan 3000
      no shutdown
      bind interface chassis1_2_RADIUS source
   #exit
#exit
port ethernet 18/5
   no shutdown
#exit
port ethernet 18/6
   no shutdown
#exit
port ethernet 18/7
   no shutdown
#exit
port ethernet 18/8
   no shutdown
#exit
port ethernet 19/1
   no shutdown
#exit
```

```
    task facility sessmgr start aggressive
    task facility acsmgr start aggressive
end
```

Category-based Content Filtering Configuration

This section presents a sample configuration file with Category-based Content Filtering configuration within an ECS service.

```
config
   license key "\
VER=1|C1M=SanDiskSDCFJ-4096|C1S=016816D2597X4624|C2M=SanDiskSDCFJ-4096\
FAA=Y|FCP=Y|LCF=30000|SIG=MC0CFQC2Zp+qSGqGR+VQ5QdhkHksZgXxgAIUN7+bT/OL"
   aaa large-configuration
   timestamps
   autoconfirm
   clock timezone asia-calcutta
      crash enable encrypted url 90b248ca778edc0db4a55318525bc
   card 1
      mode active psc
   #exit
   card 2
      mode active psc
   #exit
   card 3
      mode active psc
   #exit
   card 4
      mode active psc
   #exit
   require session recovery
   content-filtering category database directory path /flash/cf/
   require active-charging content-filtering category static-and-dynamic
```

```
   context local
      interface spio
         ip address 1.2.3.4 255.255.255.0
      #exit
      server ftpd
      #exit
      ssh key
f22330a765e10f40001920bf01dbf89a224dd8f09fe8d1598751401cb392f3c062f859a59520b1a8
f0684335cb92f4a352a4686dcea99e4740be8a0063da1c657c5609 len 006
    ssh key
75f41778bab0a173ee6e4e79c1026389918dca8b9f4701078f6841add6a81a669d183107638abac6
c0de03f606736334e1f5ee618dc370636824c0c8aaffc96050ecb88 len 007 type v2-dsa
      server sshd
         subsystem sftp
      #exit
      server telnetd
      #exit
      server tftpd
      #exit
      subscriber default
      exit
      administrator test encrypted password abc123def456ghi789 ftp
      aaa group default
      #exit
      gtpp group default
      #exit
      ip route 0.0.0.0 0.0.0.0 2.3.4.5 spio
      ip domain-lookup
      ip domain-name test.ind.testing.com
      ip name-servers 10.4.5.253
```

```
#exit
port ethernet 24/1
   no shutdown
   bind interface spio local
#exit
ntp
   enable
   server 3.4.5.6
#exit
snmp community private read-only
snmp community public read-only
snmp target test 1.3.5.7 port 162 security-name public version 2c traps
active-charging service srv1
   ruledef http-dst
      tcp dst-port = 80
      rule-application routing
   #exit
   ruledef http-response-1x
      http reply code >= 100
      http reply code < 199
   #exit
   ruledef http-response-2x
      http reply code >= 200
      http reply code < 299
   #exit
   ruledef http-response-3x
      http reply code >= 300
      http reply code < 399
   #exit
```

```
ruledef http-response-4x
   http reply code >= 400
   http reply code < 499
#exit
ruledef http-response-5x
   http reply code >= 500
#exit
ruledef http-get
   http request method = get
#exit
ruledef http-post-req
   http request method = post
#exit
ruledef http-src
   tcp src-port = 80
   rule-application routing
#exit
ruledef wsp-cl-dst
   udp dst-port = 9200
   rule-application routing
#exit
ruledef wsp-cl-src
   udp src-port = 9200
   rule-application routing
#exit
ruledef wsp-co-dst
   udp dst-port = 9201
   rule-application routing
#exit
```

```
ruledef wsp-co-src
   udp src-port = 9201
   rule-application routing
#exit
ruledef wsp-get-req
   wsp pdu-type = get
#exit
ruledef wsp-post-req
   wsp pdu-type = post
#exit
ruledef wsp-put-req
   wsp pdu-type = put
#exit
edr-format web-hit
   attribute radius-user-name priority 1
   attribute radius-calling-station-id priority 2
   attribute   sn-end-time format MM/DD/YYYY-HH:MM:SS priority 3
   attribute   sn-start-time format MM/DD/YYYY-HH:MM:SS priority 4
   attribute radius-nas-ip-address priority 5
   rule-variable http url priority 6
   rule-variable wsp url priority 7
   rule-variable ip subscriber-ip-address priority 8
   attribute sn-closure-reason priority 22
   attribute sn-cf-category-policy priority 23
   attribute sn-cf-category-rating-type priority 24
   attribute sn-cf-category-classification-used priority 25
   attribute sn-cf-category-flow-action priority 26
   attribute sn-cf-category-unknown-url priority 27
   attribute sn-volume-amt ip pkts uplink priority 50
```

```
   attribute sn-volume-amt ip pkts downlink priority 51
   attribute sn-volume-amt ip bytes uplink priority 52
   attribute sn-volume-amt ip bytes downlink priority 53
   rule-variable http request method priority 54
   rule-variable http content type priority 70
   rule-variable http reply code priority 75      #exit
charging-action standard
   content-id 10
#exit
content-filtering category policy-id 1
   analyze priority 65535 all action allow
#exit
content-filtering category policy-id 2
   analyze priority 65535 all action allow
#exit
content-filtering category policy-id 3
   analyze priority 65535 all action allow
#exit
content-filtering category policy-id 4
   analyze priority 1 category ABOR    action allow edr web-hit
   analyze priority 2 category ADULT   action allow edr web-hit
   analyze priority 3 category ADVERT action allow edr web-hit
   analyze priority 4 category ANON    action allow edr web-hit
   analyze priority 5 category ART      action allow edr web-hit
   analyze priority 7 category AUTO    action allow edr web-hit
   analyze priority 8 category BLACK   action allow edr web-hit
   analyze priority 9 category BLOG    action allow edr web-hit
   analyze priority 10 category BUSI    action allow edr web-hit
   analyze priority 11 category CAR      action allow edr web-hit
```

```
analyze priority 12 category CHAT    action allow edr web-hit
analyze priority 14 category CMC      action allow edr web-hit
analyze priority 15 category CRIME   action allow edr web-hit
analyze priority 16 category CULT    action allow edr web-hit
analyze priority 17 category DRUG    action allow edr web-hit
analyze priority 18 category EDU      action allow edr web-hit
analyze priority 19 category ENT      action allow edr web-hit
analyze priority 20 category FIN      action allow edr web-hit
analyze priority 21 category FORUM   action allow edr web-hit
analyze priority 22 category GAMB    action allow edr web-hit
analyze priority 23 category GAME    action allow edr web-hit
analyze priority 24 category GOVERN action allow edr web-hit
analyze priority 25 category GLAM    action allow edr web-hit
analyze priority 26 category HACK    action allow edr web-hit
analyze priority 27 category HATE    action allow edr web-hit
analyze priority 28 category HEALTH action allow edr web-hit
analyze priority 29 category HOBBY   action allow edr web-hit
analyze priority 30 category HOSTS   action allow edr web-hit
analyze priority 31 category KIDS    action allow edr web-hit
analyze priority 32 category LEGAL   action allow edr web-hit
analyze priority 33 category LIFES   action allow edr web-hit
analyze priority 34 category MAIL    action allow edr web-hit
analyze priority 35 category MIL      action allow edr web-hit
analyze priority 36 category NEWS    action allow edr web-hit
analyze priority 37 category OCCULT action allow edr web-hit
analyze priority 39 category PEER    action allow edr web-hit
analyze priority 40 category PERS    action allow edr web-hit
analyze priority 42 category POLTIC action allow edr web-hit
analyze priority 43 category PORN    action allow edr web-hit
```

```
    analyze priority 44 category PORTAL action allow edr web-hit
    analyze priority 45 category PROXY   action allow edr web-hit
    analyze priority 47 category REF      action allow edr web-hit
    analyze priority 48 category REL      action allow edr web-hit
    analyze priority 49 category SEARCH action allow edr web-hit
    analyze priority 50 category SCI      action allow edr web-hit
    analyze priority 52 category SHOP    action allow edr web-hit
    analyze priority 53 category SPORT   action allow edr web-hit
    analyze priority 55 category SUIC    action allow edr web-hit
    analyze priority 57 category SXED    action allow edr web-hit
    analyze priority 58 category TECH    action allow edr web-hit
    analyze priority 59 category TRAV    action allow edr web-hit
    analyze priority 60 category VIOL    action allow edr web-hit
    analyze priority 61 category WEAP    action allow edr web-hit
    analyze priority 62 category WHITE   action allow edr web-hit
    analyze priority 63 category UNKNOW action allow edr web-hit
#exit
rulebase rulebase1
    action priority 1 ruledef http-response-1x charging-action standard
    action priority 2 ruledef http-response-2x charging-action standard
    action priority 3 ruledef http-response-3x charging-action standard
    action priority 4 ruledef http-response-4x charging-action standard
    action priority 5 ruledef http-response-5x charging-action standard
    action priority 10 ruledef http-get charging-action standard
    route priority 78 ruledef http-src analyzer http
    route priority 79 ruledef http-dst analyzer http
    no transport-layer-checksum verify-during-packet-inspection
#exit
rulebase rulebase2
```

```
        content-filtering category policy-id 4
        content-filtering mode category static-and-dynamic
        content-filtering flow-any-error permit
        action priority 1 ruledef http-response-1x charging-action standard
        action priority 2 ruledef http-response-2x charging-action standard
        action priority 3 ruledef http-response-3x charging-action standard
        action priority 4 ruledef http-response-4x charging-action standard
        action priority 5 ruledef http-response-5x charging-action standard
        action priority 10 ruledef http-get charging-action standard
        route priority 78 ruledef http-src analyzer http
        route priority 79 ruledef http-dst analyzer http
        no transport-layer-checksum verify-during-packet-inspection
    #exit
    rulebase default
    #exit
#exit
context test_src
    interface TEST_CLIENT
        ip address 1.1.1.1 255.255.255.0
        ip address 1.1.1.200 255.255.255.0 secondary
    #exit
    subscriber default
        encrypted password 123abc456def789ghi
        ip context-name test_dest
    exit
    subscriber name cf
        encrypted password 123abc456def789ghi
        ip access-group acl1 in
        ip access-group acl1 out
```

```
      ip context-name test_dest
      active-charging rulebase rulebase2
   exit
   subscriber name ecs
      encrypted password 123abc456def789ghi
      ip access-group acl1 in
      ip access-group acl1 out
      ip context-name test_dest
      active-charging rulebase rulebase1
   exit
   domain cf.com default subscriber cf
   domain ecs.com default subscriber ecs
   aaa group default
      radius attribute nas-ip-address address 1.1.1.200
      radius server 1.1.1.10 key secret port 1111
      radius accounting server 1.1.1.10 key secret port 2222
   #exit
   gtpp group default
   #exit
   ha-service test_ha
      mn-ha-spi spi-number 1000 encrypted secret 123abc456def789ghi hash-
algorithm md5
      fa-ha-spi remote-address 1.1.1.100 spi-number 777 secret
123abc456def789ghi hash-algorithm md5
      no reg-lifetime
      bind address 1.1.1.200
   #exit
   pdsn-service test_pdsn
      spi remote-address 1.1.1.100 spi-number 256 encrypted secret
123abc456def789ghi
```

```
            authentication pap 1 chap 2 mschap 3
            bind address 1.1.1.200
         #exit
      #exit
      context test_dest
         ip access-list acl1
            redirect css service srv1 any
         #exit
         ip pool pool3 70.70.0.0 255.255.0.0 public 0 policy allow-static-
allocation
         interface TEST_SERVER
            ip address 1.1.1.1 255.255.255.0
            ip address 1.1.1.200 255.255.255.0 secondary
         #exit
         ssh key
75f41778bab0a1731c19851a8e68f5e9cef4cca2bd3adf9544ec64f75a8d3823028f57815369b9b7
3388f688261e49f5d200bef8c435459db536c97e4eb len 777 type v2-raa
         subscriber default
         exit
         aaa group default
         #exit
         gtpp group default
         #exit
         ip route 0.0.0.0 0.0.0.0 1.1.1.100 TEST_SERVER
         edr-module active-charging-service
            file rotation volume 123456789 headers
            cdr use-harddisk
         #exit
      #exit
      bulkstats collection
```

```
   bulkstats mode
      file 1
         schema cf format %cf-ttlsub%,%cf-cursub%
         schema cf-system format CF,PDSNSystem,%date%,%time%,%cf-static-
ratereq%,%cf-static-ratesucc%,%cf-static-rateblock%,%cf-static-ratefail%,%cf-
static-ratefail-nr%,%cf-static-ratefail-notindb%,%cf-dyn-ratereq%,%cf-dyn-
ratesucc%,%cf-dyn-rateblock%,%cf-dyn-ratefail%,%cf-cache-hits%,%cf-cache-
misses%,%cf-cache-has-path-hits%,%cf-cache-flushes%,%cf-ratereq%,%cf-
ratesucc%,%cf-rateblock%,%cf-ratefail%,%cf-cat-pkts-hit-summary%,%cf-cat-pkts-
block-summary%
      #exit
   #exit
   #exit
   port ethernet 18/4
      no shutdown
      vlan 11
         no shutdown
         bind interface TEST_CLIENT test_src
      #exit
   #exit
   port ethernet 18/8
      no shutdown
      vlan 31
         no shutdown
         bind interface TEST_SERVER test_dest
      #exit
   #exit
   task facility sessmgr start aggressive
end
```