# ·IIIII CISCO

### **CISCO SERVICE CONTROL SOLUTION GUIDE**



## **Cisco Service Control Online Advertising Solution Guide: Behavioral**

## Profile Creation Using RDRs, Release 5.0.x

- 1 Overview
- 2 Configuring Behavioral Targeting Support: Highlights
- **3** Configuring a Cisco SCE Platform for Exporting Behavioral Targeting Information
- 4 Anonymized HTTP Transaction Usage RDR
- 5 Hash AlgorithmObtaining Documentation and Submitting a Service Request



This document supports all 5.0.x releases.

# **1** Overview

Online Behavioral Targeting is an online advertising approach that involves presenting users with advertisements based on their interests, as deduced by monitoring their web browsing preferences. The Cisco Service Control Engine (Cisco SCE) platform can enable online behavioral targeting based on an analysis of subscriber online usage patterns.

Such behavioral targeting does not require the analysis of each and every HTTP request on the line, because such analysis would result in numerous excess information. The Cisco SCE platform performs the first level of analysis in the behavioral targeting chain by inspecting the user browsing sessions, detecting the particular requests that are triggered by the actual user browsing (these events are termed ClickStream), and generating Raw Data Records (RDRs) that contain a digest of these events. To avoid compromising subscriber privacy, the RDRs can be configured to not include any Personally Identifiable Information (PII), in which case all elements containing PII are hashed. The RDRs are typically received by an entity that analyzes the nature of usage and creates a profile of the subscriber to be used later for targeting. The way the greater solution works is outside the scope of this document.

Figure 1 illustrates the high-level overview of an RDR-based behavioral targeting solution.





ClickStream detection is a fundamental capability of the solution, because it can detect which specific requests, out of the enormous number of HTTP requests generated throughout the subscriber web activity, were actually triggered by the subscriber browsing the web. When a subscriber clicks a link, or enters a URL in the browser address bar, an HTTP request is generated to fetch this URL. Typically, an HTML page is returned, which constitutes the outline of the contents requested. For the browser to be able to render this page, it must download multiple objects (tens or sometimes around a hundred for a single page viewed), which in turn results in multiple HTTP requests for obtaining these objects.

To carry out behavioral targeting, it is sufficient to understand what the user was trying to do (represented by the initial request, such as biz.publisher.com/ap/081120/world\_markets.html > global markets), rather than looking at each object downloaded as a secondary result of such a request (such as: http://ads.adnetwork.com/a/a/in/interbroke/300x250\_yah.jpg > broker advertisement).

ClickStream detection makes exactly this distinction, allowing the number of requests to be analyzed to be greatly reduced, which is necessary to enable a scalable analysis solution.

The information that is collected per such transaction is exported from the Cisco SCE using Extended Transaction Usage RDRs, which includes information on the transaction that has been performed, in addition to information on the subscriber that performed the transaction.

PII, such as subscriber ID and IP address, is protected by hashing this information with the use of preconfigured "salt". In this way, the subscriber record can be matched only by another system configured with the same salt, in which case it has the original PII to match against. Salt is configurable to the Cisco SCE platform.

## **2** Configuring Behavioral Targeting Support: Highlights

This section provides the highlights of configuring the main components of behavioral targeting on the Cisco SCE platform. For configuration details, see the "Configuring a Cisco SCE Platform for Exporting Behavioral Targeting Information" section on page 6.

## **Creating a ClickStream Service**

ClickStream signatures are mapped by default to the HTTP Browsing protocol and consequently to the browsing service. To be able to act on them separately, first move them to a protocol of their own, then assign this protocol to a service of its own.

Figure 2 and Figure 3 illustrate configuring the ClickStream protocol and service.

Figure 2 Configuring the ClickStream Protocol

Protocol Settin	igs							×
Protocols					Protocol Elements			
(Show All)	*	÷	×	1			4	×
worldfusion				^	Signature	IP Protocol	Port Range	
😻 wpages					In-Domain Click Stream	*	*	
😻 wpgs					In-Domain Click Stream - Uni	*	*	
😻 wta-wsp-s					Cross-Domain Click Stream	*	*	
😻 x-bone-ctl					Cross-Domain Click Stream			_
😻 xact-backup								
😻 xbox live								
😻 xdmcp								
😻 xdtp								
😻 xfer								
😻 xfr								
😻 xns-auth								
😻 xns-ch								
😻 xns-courier								
😻 xns-mail								
😻 xns-time								
😻 xvttp								
😻 xyplex-mux								
😻 yak-chat								
😻 z39.50								
😻 zannet								
😻 zserv								
🐼 ClickStream				~				
?							Close	

#### Figure 3 Configuring the ClickStream Service

File Tools Preferences Wedow Help	¥ Service Configuration Editor - ≤new se	rvice configuration> -	SCA BB Console				
Subscription Honorger	File Tools Preferences Window Help						
Subscriber Manager Subscriber M	i 🗋 😂 🔛 i 🏶						
	🗈 🙀 Subscriber Manager 🛛 Eignature Editor	Service Configuration E	ditor 🕐 Network Navigal	or			
Clusticution:       Clusticution:<	S * <new configuration="" service=""> 🛛</new>						- 0
Peldes Configuration  CeldStream 1 CidStream  Protocol Initiating Side Zones Flovor  CidStream Initiation Side Zones Flovor  CidStream  Flovor  CidStream  Flovor  CidStream  Flovor  CidStream  Flovor  CidStream  Flovor  CidStream  Flovor	Classification : ClickStream						
Configuration  Config	Policies 📕 Classification						🗣 🔶 🔶 🗙 📐
Provining     Protocol     Initiating Side     Zones     Flavor       Image: Strategy of the strate	🛷 🐥 🗶 🔪 Configuration 🗸	ClickStream : ClickStream					
ClubStream Initiated by eather side	Representation	Protocol	Initiating S	ide	Zones	Flavor	
Consele Concess     Concesse      Conce	HTTP	ClickStream	Initiated by	either side	•	•	
	HTTPS						
	Location Based Services						
Pie Shaing     Dotant Messaging     Dotant Video     Dotant     Dotant Video     Dotant	. E-Mail						
	🗄 🖳 File Sharing						
	Gaming     Jostant Messaging						
Protect Vdeo     Protect Vdeo     Protect Vdeo     Protect Advin     Protect Advin     Protect Advin     Protect Advin     Other     Other     Other     Other     Other     Protect Advin	Internet Privacy						
Bet Admin         Mexagouss         Bet Admin         Newsgouss         Bet Admin         Other         Other         Other         Console         Decross, 0 warnings, 0 infos         Report Only         Encode	🗄 📃 Internet Video						
Voce and Video Cals  Voce and Video Cals  Console  Console  Problems  D errors, 0 warnings, 0 infos  Report Only	Net Admin     Newscroups						
Concole Concole Problems O errors, 0 wernings, 0 infos Report Only	Voice and Video Calls						
© Console (%, Problems 0 errors, 0 warnings, 0 infos	🛓 🖷 Other						
Console [1] Problems Cerrors, 0 warnings, 0 linfos Report Only							
Console [1] Problems 0 errors, 0 wernings, 0 infos							
Console C Problems 0 errors, 0 warnings, 0 infos							
Console  Problems O errors, 0 warnings, 0 infos Report Only							
Console [1] Problems 0 errors, 0 warnings, 0 infos							
© Console [1] Problems 0 errors, 0 wernings, 0 infos							
Console R Problems O errors, 0 warnings, 0 infos Report Only : : : : : : : : : : : : : : : : : : :							
Console  Problems  Concole Report Only  Report Only							
Console [1] Problems O errors, 0 warnings, 0 infos Report Only							
Console [L] Problems 0 errors, 0 warnings, 0 infos Report Only	L	1					
0 errors, 0 warnings, 0 infos Report Only	📮 Console 🔝 Problems						
Report Only .	0 errors, 0 warnings, 0 infos						
			Report Only				1

## **Configuring Extended Transaction Usage RDRs**

Extended Transaction Usage RDRs (TURs) contain extra protocol-specific information (see the "Anonymized HTTP Transaction Usage RDR" section on page 14). These RDRs are configured from the Transaction Usage RDRs tab of the RDR Settings dialog box of the Cisco Service Control Application for Broadband (Cisco SCA BB) console.

Enable these RDRs explicitly by checking the appropriate check boxes, and choosing the packages you want them generated for (you only need to enable these RDRs for the ClickStream service in each package).

Optionally, these RDRs can be anonymized, in which case all PII in the RDR is hashed using Message Digest 5 (MD5). Hashing is done using the "salt" configured on the SCE platform. For GUI options on anonymizing the RDRs, see Transaction Usage RDR Settings (Figure 4).

#### Figure 4 Transaction Usage RDR Settings

Usage RDRs		Transaction RDI	હ	Quota RDRs
ransaction Usage RDRs	Log RDRs	Real-Time Su	bscriber RDRs	Real-Time Signaling RDRs
ansaction Usage RDRs				
ansaction Usage RDRs rep ir specific services and sub	resent single netw scribers of specific	ork transactions. packages.	They can be used to	generate detailed usage logs
Generate the following	types of Transacti	on Usage RDRs (	'URs):	
Generic TUR				
Protocol specific TL	IRs (include protoci	ol specific informa	tion)	
Extended TURs (in	clude extended pro	tocol specific info	rmation)	
Anonymize - Enc	rypt Personally Ide	ntifiable Informat	on	
chages and services for w	hich Transaction U: Select/Deselect Default Pad Default Default Default Default Unknown St	sage RDRs should All kage Service Sam Jbscriber Traffic	be generated:	
Transaction Usage RDR	s can be filtered ac for sessions excee	cording to the se	sion's volume Bytes	
Transaction Usage RDR	s can be generated	at fixed interval	, during long flows	
Enable Interim TUR	to be generated e	wery	Minutes (from 5 min	utes up to 2880 minutes)

## **Enabling Deep HTTP Inspection**

To ensure comprehensive detection of the ClickStream events in the traffic stream, it is important to enable deep inspection of HTTP, which configures the Cisco SCE platform to analyze and classify all HTTP requests within a single flow.

Some browsers, in conjunction with some web server implementations, use the same TCP flow to carry multiple requests triggered by clicks that target the same host. Such events are not detected if the classification is done only at the beginning of the flow (which is the default for Cisco SCA BB).

To enable deep HTTP inspection, in the Cisco SCA BB Console Service Configuration Editor, choose Configuration > System Settings > Advanced Options > Advanced Service Configuration Options.

Note

Enabling deep HTTP inspection is expected to impact the Cisco SCE performance because of the excessive processing associated with it, the actual figure depending on the amount and on the nature of HTTP traffic. We recommend that you monitor SCE platform performance when enabling this capability.

## **Configuring MD5 Salt to the SCE Platform**

When the Anonymize option is selected, Extended TURs are generated with all personally identifying fields (subscriber ID and subscriber IP address) hashed using MD5.

The Cisco SCE platform applies a "salt" to the personally identifying field before hashing it. The salt is 128 bits (16 bytes) long, and it is configured to the SCE platform in four separate 4-byte arguments represented in HEX. The default value of the salt is 0x12345678 0x12345678 0x12345678 0x12345678.

The salt is configured to the SCE by using these CLI commands with four 4-byte arguments in HEX:

```
SCE10000# config
SCE10000(config)# interface linecard 0
SCE10000(config if)# salt 0x12345678 0x00004321 0xfafafafa 0xafafafaf
```

To return the salt to the default value use these CLI commands:

```
SCE10000# config
SCE10000(config)# interface linecard 0
SCE10000(config if)# default salt
```

Save the running configuration by using this command:

SCE10000# copy running-config startup-config

## **Configuring RDR Routing to the Profiling Server**

Ensure that only ClickStream RDRs are sent to the designated server, even if other RDRs are enabled on the system. This is done by directing the extended HTTP TUR to a separate category, and routing this category to a dedicated server.

This is accomplished in two steps:

- Direct the HTTP Extended TUR to an exclusive RDR category. Map the Extended TUR for HTTP (tag 0xF0F0F53C / 4042323260) to RDR category 2 by using this CLI command: SCE10000(config)# RDR-formatter rdr-mapping tag-ID 0xF0F0F53C category-number 2
- Send the RDRs in this category to the designated server.
   Configure RDR category 2 to the desired destination by using this CLI command:
   SCE10000(config)# RDR-formatter destination 10.10.10.10 port 33000 category number 2 priority 100

You can configure a second destination for the category 2 RDRs to function as a backup destination.

- Configure a secondary destination for backup as needed.
   Configure RDR category 2 to a secondary destination by using this CLI command:
   SCE10000(config)# RDR-formatter destination 10.10.10.11 port 33000 category number 2 priority 90
- Save the running configuration by using this CLI command: SCE10000# copy running-config startup-config

# **3** Configuring a Cisco SCE Platform for Exporting Behavioral Targeting Information

This section explains in detail how to configure an Cisco SCE to generate ClickStream RDRs.

- Step 1 In the Cisco SCA BB Policy Editor, click the Classification tab (left pane), click Configuration, and select Protocols.
- Step 2 In the Protocol Settings window (see Figure 5), select the HTTP Browsing service.
- **Step 3** On the **Protocol Elements** tab, remove the ClickStream-related protocol elements:
  - In-Domain ClickStream
  - In-Domain ClickStream Unidirectional Client Request
  - Cross-Domain ClickStream
  - Cross-Domain ClickStream Unidirectional Client Request

#### Figure 5 Protocol Settings

	Protocol Elements		
(Show All) 🕑 🐈 🗙 📐			÷ 🗙
🔊 Generic TCP 🔗	Signature	IP Protocol	Port Range
Generic UDP	Generic	TCP	80
≶ Gnutella File Transfer	Generic	TCP	8080
🚿 Gnutella Networking	HTTP	*	*
Google Talk	HTTP - Unidirectional Server	*	*
Soogle Talk File Transfer	HTTP - Unidirectional Client	*	*
Google Talk Voice	Cross-Domain Click Stream	*	*
	Cross-Domain Click Stream	*	*
	In-Domain Click Stream	*	*
∭r H323	In-Domain Click Stream - Uni	*	*
HTTP Browsing HTTP Tunnel HULU Heretic II Heretic II Hopster Hopster I-NLSP IATP			

**Step 4** In the Protocol Settings window, on the Protocols tab, click the Add (+) icon to add a new protocol.

**Step 5** Enter a name for the new protocol ClickStream Event, and click OK (see Figure 6).

#### Figure 6 Protocol Settings-Name

Protocol Settings			Þ
Protocols	Protocol Elements		
(Show All) 🕑 🖶 🗙 📐			÷ ×
😻 Generic TCP 🛛 🧖	Signature	IP Protocol	Port Range
🧭 Generic UDP	Generic	TCP	80
😻 Gnutella File Transfer	Generic	TCP	8080
👹 Gnutella Networking	▶ 1 ¥ Protocol Sett	ings 🔣 🔜	*
😻 Google Talk	F Convert		*
😻 Google Talk File Transfer	General		*
🐼 Google Talk Voice	Name:		*
🤝 GoogleEarth	ClickStream Ever	əti 👘	*
😻 H323	I		*
🤝 HMP	Description:	1	
W HOPOPT			
🐼 HTTP Browsing			
W HTTP Tunnel		8	
🐼 HULU			
😻 Heretic II	Protocol ID: 50	00	
🤝 Hexen			
😻 Hopster		Canad	
😹 Hotline			
🐼 I-NLSP			
W IATP			
S ICMP			
ICQ			
😻 ICQ File Transfer			
0			Class
U			Close

**Step 6** On the Protocol Elements tab, click the Add (+) icon to add protocol elements to the ClickStream Protocol.

Step 7 For the new protocol element created, click the "..." button in the Signature column.

**Step 8** In the Select a Signature dialog box (see Figure 7), add the In-Domain Click Stream signature, and click OK.



3	🗲 Select a Signature	
	Select a signature for the service element	
	in	L
	In-Domain Click Stream In-Domain Click Stream - Unidirectional Client Request	l
		l
		l
		l
		l
	2	L
	OK Cancel	273993

**Step 9** Repeat Step 6 through Step 8 for the rest of the ClickStream signatures:

- In-Domain Click Stream Unidirectional Client Request
- Cross-Domain Click Stream

- Cross-Domain Click Stream Unidirectional Client Request
- Step 10 In the Cisco SCA BB Policy Editor, click the Classification tab (left pane), and highlight the Browsing service.
- **Step 11** Click the Add (+) icon to add a new service under the Browsing service.
- Step 12 Name the service ClickStream (or any other name you choose) (see Figure 8).

#### Figure 8 Service Settings

Name: ClickStream Description:	
ClickStream Description:	
Rescription:	
_	

**Step 13** Click the Hierarchy tab (see Figure 9) and check the two check boxes to add a dedicated service counter to the ClickStream Service. (This is useful if you want to generate reports on the global and per-subscriber ClickStream activity.)

🗚 Service Settings for "ClickStream"
General Hierarchy Advanced
Parent Service
Parent Service: Browsing
Service Usage Counters
Per usage accounting scope (global accounting and subscriber accounting), a service can either be mapped to an exclusive usage counter, or share a usage counter with its ancestor service.
Map this Service to an exclusive Global usage counter
Global counter used by this service: ClickStream Counter Counter Index: 75
Map this Service to an exclusive Subscriber usage counter
Subscriber counter used by this service: ClickStream Counter Counter Index: 30 💌
(?) OK Cancel

#### Figure 9 Hierarchy

#### Step 14 Click OK.

**Step 15** In the right pane, click the Add (+) icon to add a service element.

- **Step 16** In the dialog box that opens, click **Select** next to the Protocol field and select the **ClickStream Event** protocol (or whatever you named your ClickStream protocol) from the list (see Figure 10).
- Step 17 Click OK.

Edit Service E	×	
Edit any of the se	Select	
Assign the ser Service: Choose which Protocol:	cl ClickStream Event cloanto-net-1 Club Box cmip-agent cmip-man coauthor codauthor codauth2	ct perties:
Initiating Side: Zone: Flavor:	collaborator commerce Compaq-Peer compressnet comscm con con	₫ 
	conference connendp	OK Cancel rel

- **Step 18** In the Cisco SCA BB Policy Editor, click the **Policies** tab (left pane), and then select the package for which to generate the ClickStream RDRs.
- **Step 19** In the right pane, click the Add (+) icon to add the ClickStream service.
- **Step 20** In the window that opens (see Figure 11), select ClickStream from the drop-down list.

#### Figure 11 Add New Rule to Package

Add New Rule to Package "Default Package"
General Control Usage Limits Breach Handling
Service Service to which the Rule will relate:
Rule State         Define the state of this Rule:            • Enable reporting and active actions             Disable reporting and active actions

- Step 21 Click OK.
- Step 22 In the Cisco SCA BB Policy Editor, repeat Step 18 through Step 21 for every package for which you want to enable ClickStream RDRs.
- **Step 23** In the right pane, click **Configuration** and choose **Classification** > **RDR** Settings.
- Step 24 Click the Transaction Usage RDRs tab (see Figure 12).

#### Figure 12 RDR Settings—Transaction Usage RDRs

Usage RDRs	1	Transaction RDRs	Quota RDRs
ransaction Usage RDRs	Log RDRs	Real-Time Subscriber RDRs	Real-Time Signaling RDRs
ansaction Lisage RDRs			
ransaction Usage RDRs rep	resent single netv	work transactions. They can be used	to generate detailed usage logs
or specific services and sub-	scribers of specific	c packages.	
Generate the following	types of Transact	tion Usage RDRs (TURs):	
Generic TUR			
Protocol specific TU	Rs (include proto	col specific information)	
Extended TURs (inc	lude extended or	otocol specific information)	
Anonymize - Encr	ypt Personally Id	entifiable Information	
ackages and services for w	hich Transaction L	Jsage RDRs should be generated:	
20	Select/Deselec	t All	
	Deraul Pa	t Service	
		ream	
	City Originowing	subscriber framic	
Transaction Usage RDRs	can be filtered a	ccording to the session's volume	
Generate TUR only	for sessions exce	eding Bytes	
Transaction Usage RDRs	can be generate	d at fixed intervals, during long flows	R.
Enable Interim TUR	to be generated	every Minutes (from 5 m	inutes up to 2880 minutes)

- **Step 25** Enable Extended TURs by checking the relevant check box.
- **Step 26** (Optional) Enable the anonymize option as needed, by checking **Anonymize Encrypt Personally Identifiable** Information.
- **Step 27** Check the ClickStream service on all relevant packages, to enable generating RDRs for this service. Uncheck any other service for which you do not want Transaction Usage RDRs (extended or regular) generated.



Make sure that you have already created a specific rule for ClickStream on all relevant packages.

**Step 28** Choose Policies > Configuration > System Settings (see Figure 13).

Figure 13 Service Configuration Editor – Policies > Configuration > System Settings



Step 29 On the Advanced Options tab, click Advanced Service Configuration Options (see Figure 14) to enable deep inspection of HTTP flows (beyond the first transaction) by setting the highlighted value to 64000. This enables the analysis of multiple transactions within a single HTTP flow, which is important for comprehensive detection of ClickStream events.

Figure 14 Advanced Service Configuration Options

* Advanced Service Configuration Uptions					
View & edit advanced options					
<b>-</b> .					
Property	Value	<u></u>			
Bandwidth Management					
Relax BWC enforcement on netw	Use Default Service BWCs (relaxed enforcement)				
Classification	4 1000				
Enable deep HTTP inspection	64000				
Guruguru detailed inspection mod	false				
Kuro detailed inspection mode er	false				
Soribada detailed inspection mod	false				
TCP destination port signatures	1720:H323				
UDP destination port signatures	67:DHCP,1812:Radius Access,1645:Radius Access,1813:				
UDP ports for which flow should I	5060,5061,67,69,1812,1813,1645,1646,2427,2727,920				
UDP source port signatures	1812:Radius Access, 1645:Radius Access, 1813:Radius Ac				
V-Share detailed inspection mode	false				
Winny detailed inspection mode (	false				
Malicious Traffic					
Malicious Traffic RDRs enabled	true				
Number of seconds between Mal	60				
TCP port that should remain oper	80				
Multi Stage Classification					
Blocking	true				
Enable	true	~			
	ОК				

- **Step 30** Apply the Service Configuration to the SCE platform.
- **Step 31** (Optional) If the anonymize option is enabled, configure MD5 salt to the SCE platform (use a 128-bit salt value of your choice) by using this command:

SCE10000(config if)# **salt** 0xfafafafa 0xfafafafa 0xfafafafa 0xfafafafa

**Step 32** Map the Extended TUR for HTTP (tag 0xF0F0F53C / 4042323260) to RDR category 2 by using this command:

SCE10000(config)# RDR-formatter rdr-mapping tag-ID 0xF0F0F53C category-number 2

- Step 33 Direct category 2 to your server of choice by using this command: SCE10000(config)# RDR-formatter destination 10.10.10.10 port 33000 category number 2 priority 99
- **Step 34** Save the configuration by using this command:

```
SCE10000# copy running-config-all startup-config-all
```

## **4** Anonymized HTTP Transaction Usage RDR

The RDR tag of the Anonymized HTTP Transaction Usage RDR is F0 F0 F5 3C / 4042323260. Table 1 lists the RDR fields and their descriptions.

Table 1	Anonymized HTTP	Transaction	Usage RDR	Fields
	Anonymizcumm	mansaction	o Suge non	i icius

RDR Field Name	Туре	Description	
HASHED_SUBSCRIBER_ID	STRING	Subscriber identification string, passed through hashing algorithm as described in the "Hash Algorithm" section on page 16.	
PACKAGE_ID	INT16	ID of the package assigned to the subscriber whose traffic is being reported.	
SERVICE_ID	INT32	Service classification of the reported session.	
PROTOCOL_ID	INT16	Unique ID of the protocol associated with the reported session.	
SKIPPED_SESSIONS	UINT32	Always 1.	
SERVER_IP	UINT32	Destination IP address of the reported session. The destination is defined as the server or the listener of the networking session.	
		The IP address is in a 32-bit binary format, but the value is obtained in decimal format in RDR. For example, 3228978306.	
		If this is the subscriber IP, this field contains the short-hash of the IP, as described in the "Hash Algorithm" section on page 16.	
SERVER_PORT	UINT16	Destination port number of the networking session.	
HOST	STRING	Host extracted from the HTTP transaction.	
URL	STRING	URL extracted from the HTTP transaction.	
CLIENT_IP	UINT32	IP address of the client side of the reported session. The client side is defined as the initiator of the networking session.	
		The IP address is in a 32-bit binary format, but the value is obtained in decimal format in RDR. For example, 3228978306.	
		If this is the subscriber IP, this field contains the short-hash of the IP, as described in the "Hash Algorithm" section on page 16.	
CLIENT_PORT	UINT16	Port number of the client side (initiator) of the networking session.	
INITIATING_SIDE	INT8	Side of the SCE platform on which the initiator of the transaction resides.	
		• 0—Subscriber side	
		• 1—Network side	
REPORT_TIME	UINT32	Ending time stamp of this RDR.	
MILLISEC_DURATION	UINT32	Duration, in milliseconds, of the transaction reported in this RDR.	
TIME_FRAME	INT8	Time frame during which the RDR was generated. (The range is from 0 to 3.)	

#### Table 1 Anonymized HTTP Transaction Usage RDR Fields (continued)

RDR Field Name	Туре	Description	
SESSION_UPSTREAM_ VOLUME	UINT32	Upstream volume of the transaction, in bytes. The volume refers to the aggregated upstream volume on both links of all the flows bundled in the transaction.	
SESSION_DOWNSTREAM_ Volume	UINT32	Downstream volume of the transaction, in bytes. The volume refers to the aggregated stream volume on both links of all the flows bundled in the transaction.	
SUBSCRIBER_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 32 subscriber usage counters.	
GLOBAL_COUNTER_ID	UINT16	Each service is mapped to a counter. There are 128 global usage counters.	
PACKAGE_COUNTER_ID	UINT16	Each package is mapped to a counter. There are 1024 package usage counters.	
IP_PROTOCOL	UINT8	IP protocol type.	
PROTOCOL_SIGNATURE	UINT32	ID of the protocol signature associated with this session.	
ZONE_ID	UINT32	ID of the zone associated with this session.	
FLAVOR_ID	UINT32	For protocol signatures that have flavors, this field contains the ID of the flavor associated with this session.	
FLOW_CLOSE_MODE	UINT8	Reason for the end of flow.	
HASHED_SUBSCRIBER_IP	STRING	Subscriber IP, hashed as described in the "Hash Algorithm" section on page 16.	
USER_AGENT	STRING	User agent field extracted from the HTTP transaction.	
HTTP_REFERER	STRING	REFERER extracted from the HTTP transaction.	
HTTP_COOKIE	STRING	Cookie extracted from the HTTP transaction.	

Further elaboration on RDR fields:

• HASHED\_SUBSCRIBER\_ID—Subscriber identification string, introduced through the subscriber management interfaces, passed through hashing algorithm as described in the "Hash Algorithm" section on page 16.

The field is a 32-byte-long string, containing a Hexadecimal notation of the 128-bit hash result.

- PACKAGE\_ID—ID of the package assigned to the subscriber whose traffic is being reported. An assigned Package ID is an integer value between 0 and maximum\_number\_of\_packages. The maximum\_number\_of\_packages value is reserved for unknown subscribers.
- HASHED\_SUBSCRBIER\_IP—IP address of the subscriber side of the reported session, after passing through the hashing algorithm as described in the "Hash Algorithm" section on page 16.

The field is a 32-byte-long string, containing a hexadecimal notation of the 128-bit hash result.

- CLIENT\_PORT—For TCP/UDP-based sessions, the port number of the client side (initiator) of the networking session. For non-TCP/UDP sessions, this field has the value 0.
- CONFIGURED\_DURATION—For periodic RDRs, the configured period, in seconds, between successive RDRs.
- END\_TIME—Ending time stamp of this RDR. The field is in UNIX time\_t format, which is the number of seconds since midnight of January 1, 1970.
- FLAVOR\_ID—For protocol signatures that have flavors, this field contains the ID of the flavor associated with this session.
- PROTOCOL\_ID—Contains the unique ID of the protocol associated with the reported session.

# Note

The PROTOCOL\_ID is the Generic IP, Generic TCP, or Generic UDP protocol ID value, according to the specific transport protocol of the transaction, unless a more specific protocol definition (such as a signature-based protocol or a port-based protocol), which matches the reported session, is assigned to a service.

- PROTOCOL\_SIGNATURE—Contains the ID of the protocol signature associated with this session.
- REPORT\_TIME—Ending time stamp of this RDR. The field is in UNIX time\_t format, which is the number of seconds since midnight of January 1, 1970.
- SERVER\_IP—Contains the destination IP address of the reported session. (The destination is defined as the server or the listener of the networking session.) The IP address is in a 32-bit binary format.
- SERVER\_PORT—For TCP/UDP-based sessions, this field contains the destination port number of the networking session. For non-TCP/UDP sessions, this field contains the IP protocol number of the session flow.
- SERVICE\_ID—Indicates the service classification of the reported session. For example, in the Transaction RDR, this field indicates which service was accessed, and in the Breaching RDR, this field indicates which service was breached.
- TIME\_FRAME—System supports time-dependent policies, by using different rules for different time frames. This field indicates the time frame during which the RDR was generated. The value can be in the range from 0 to 3, indicating which of the four time frames was used.
- ZONE\_ID—Contains the ID of the zone associated with this session.

### 

**Note** All volumes in RDRs are reported in Layer 3 bytes.

## **5** Hash Algorithm

Hashing is done using the MD5 hash function, using a "salt" prepended to the hashed value.

- "salt" means adding a predefined value before the to-be hashed fields, to avoid simple backtracking of the source argument. The salt currently used by the SCE platform is 128 bits long and is configured to the SCE in four separate 4-byte arguments represented in HEX.
- Setting the salt is done through CLI using a hexadecimal notation. It can be updated periodically. (By default, it set to 0x12345678 0x12345678 0x12345678 0x12345678.)
- Hash result is 128 bits long, and is represented in RDRs as a 32-byte string. Each byte represents 4 bits of the result. To make it printable, add 64. (for example, 0000 mapped to ASCII(64) = '@'; 1111 mapped to ASCII(127) = '\_').
- Short-hash that appears in RDRs is composed of the lower 32 bits of the hash result.

# **Obtaining Documentation and Submitting a Service Request**

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation* at: http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html.

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental. © 2014 Cisco Systems, Inc. All rights reserved.



Americas Headquarters Cisco Systems, Inc. San Jose, CA Asia Pacific Headquarters Cisco Systems (USA) Pte. Ltd. Singapore **Europe Headquarters** Cisco Systems International BV Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.