# Application Assurance — Application Identification and User-Defined Applications

## In This Chapter

This section describes Application Assurance (AA) Application Identification and User-Defined Applications configurations.

Topics in this section include:

- Applicability on page 1376
- Overview on page 1377
- Configuration Examples on page 1387
- Conclusion on page 1409

Applicability

## Applicability

This example is applicable to all 7750, 7450 and 7750-SRc chassis supporting Application Assurance and was tested on release 12.0.R4.

There are no specific pre-requisites for this example.

## Overview

This chapter is intended for Application Assurance (AA) network architects and engineers. It provides best practice information to customize the AA policy and classify any type traffic to meet the service provider reporting, charging or control requirements.

In addition to the signatures built and supported by Alcatel-Lucent, service providers can create their own application signatures based on various criteria. This customization capability can be used to classify traffic hosted on the provider network (web portal, streaming service) or hosted on the Internet and not yet covered by the default AA signature set.

## **Basics and Terminology**

The following main components are used for AA classification:

- Application Filters App-filters are used to define applications based on Layer 3 to Layer 7 criteria. They provide a mapping between one or more protocol signatures or customized traffic patterns into an application of interest.
- Application Such as BitTorrent<sup>®</sup>, Netflix<sup>®</sup>. Traffic is classified into applications using app-filters.
- Application Group Such as peer-to-peer, multimedia streaming. For the purpose of reporting and control, applications of similar type/function can be grouped together in Application Groups (App-Group).
- Charging Group Such as zero rating, default. For the purpose of charging or control, applications and app-group can be grouped together in charging groups.

The following table is a high level example to illustrate how app-filters are used to defined applications and show their logical grouping into app-group and charging group.

#### **Basics and Terminology**

Criteria	App-Filter (ordered list of entries, ACL like)	Application	Application Group	Charging Group		
- Protocol - Expression: (HTTP, SIP, H323, TLS, RTSP)	Expression - http: yahoo.com	Yahoo	Web	CG#1 - Default		
	Expression - http: maps.google.com	Google Maps	Web	CG#2 - Zero Rating		
	Expression - http: facebook.com	Facebook	Social Networking			
- L4 Server Port - IP Server Address	Protocol: ftp_control, ftp_data	FTP	File Transfer	- CG#1 - Default		
- Custom Protocol	Protocol: bittorrent, dht, utp	BitTorrent	Deserte Deser			
	Protocol: emule	Emule	Peer to Peer			

Maximum Elavibility	to Idontify	(Standard	and Custom	Applications	of Intoract
	y to luentin	y Stanuaru a		Applications	or interest

Flexible classification/identification rules (apps-filters) to identify:

- Standard applications

Flexible applications/app-group creation and mapping for:

- Custom defined applications

- Reporting - Control (redirect, enrichment, policing...) Independent charging group mapping for differentiated billing.

al 0680

#### Figure 203: App-Filters/Applications/AppGroup

- BitTorrent<sup>®</sup> and Emule<sup>®</sup> applications are defined using their protocol signature and ٠ grouped in the P2P app-group.
- FTP application is defined using both ftp data and ftp control protocol signatures, the ٠ app is mapped in the file transfer app-group.
- Google Maps<sup>®</sup> and Yahoo<sup>®</sup> web sites are defined using http expression and grouped ٠ together in the Web app-group.

## Configuration

## **Classification Criteria (App-Filter)**

The operator can take full advantage of the flexible AA policy configuration to classify traffic from any application of interest using various criteria ranging from Layer 3 to Layer 7 expressions.

Expression match criteria allows to further refine traffic classification by identifying traffic from HTTP, HTTPS (SSL/TLS), SIP, H323, RTSP, Citrix protocol signatures.

The different app-filter match criteria are listed below:

- L7 Expression
  - ç HTTP: Host, URI, User Agent, Referer
  - ç SSL/TLS: Certificate Org Name, Common Name, SNI
  - ç H323: Product-ID
  - ç SIP: URI, User Agent, Media Type
  - ç RTSP: Host, URI, User Agent
  - ç Citrix: Application Published Name
  - ç RTMP: Page-host, page-uri, swf-host, swf-uri
- IP Protocol Number
- IP Server Address
- TCP/UDP Server Port
- Custom Protocol
- Protocol Signature

The following operators are supported to define expression based app-filters:

- ^ : Expression start with
- \$ : Expression end with
- \*: Wildcard anything before or after
- \I: Forces case sensitivity
- \d: Any single decimal digit [0-9]
- \.: Any single character
- \\*: Asterisk character

#### 7750 SR Advanced Configuration Guide

#### Classification Criteria (App-Filter)

Examples of expression match combinations:

- ^abcd\*: match 'abcd' at beginning, can end with anything
- \*abcd\*: match 'abcd' anywhere
- \*abcd\$: match 'abcd' at the end
- ^abcd\$: exact expression match 'abcd'
- ^ab\*cd\$: string starts with 'ab', ends with 'cd' (anything else in between)
- ^ab\dcd\$: string starts with 'ab', followed by a decimal digit, ends with 'cd'

Note: It is possible to combine different criteria or expressions within the same filter in which case an implicit AND operation between the criteria within the same filter is done by the system.

## **Application Definition Example**

The example below provides a basic configuration example with the application FTP made of two protocol signatures ftp\_control and ftp\_data; the application is mapped into the application group file transfer:

Create the application group.

```
configure application-assurance group 1:1 policy
    app-group "File Transfer"
    exit
```

Create the application.

```
configure application-assurance group 1:1 policy
application "FTP"
app-group "File Transfer"
exit
```

#### Create the app-filters.

```
configure application-assurance group 1:1 policy
    app-filter
    entry <1..65535> create
        protocol eq "ftp_data"
        application "FTP"
        no shutdown
    exit
    entry <1..65535> create
        protocol eq "ftp_control"
        application "FTP"
        no shutdown
exit
```

Note: Once the application is created the operator is expected to configure the collection of statistics at the subscriber level for this new application (usually only for business VPNs).

## **User-Defined Applications**

#### **General Recommendations**

In order to classify traffic properly it is recommended to follow the guidelines and best practices defined in this section before creating a new application:

- Analyze the application traffic
  - $\varsigma$  Identify what type traffic is used (Wireshark<sup>®</sup>).
  - **ç** Use the application the same way the end user would use it, the same application can create various flows.
- Configure the appropriate App-Filters
  - **ç** Following the analysis of the application done above, create the application.
  - **ç** Follow the App-Filter best practices chapter to understand in which range to add the filters.
  - **ç** More than one App-Filters can be required to identify a single application.

### AppDB/Default AA Policy

The default AA policy called AppDB (Application Database) is provided by Alcatel-Lucent and should be used on most deployments. Contact your regional support organization for more details on how to obtain it.

This configuration includes applications and application-groups most Providers can use by default and is designed to allow the addition of any custom entries required by Service Providers to identify additional services/applications.

Before adding new entries to the template and customizing the configuration it is recommended to follow the next guidelines on app-filters and ranges. These guidelines are key to allow an easy upgrade path from the policy configuration provided by Alcatel-Lucent.

## **App-Filters**

App-Filters are an ordered list of entries. It is important to keep the order of this list consistent with the classification objective.

For instance a common configuration mistake is to configure a filter rule for the HTTP protocol signature before HTTP expression filters. If that was the case then app-filters using HTTP expressions would not be used as the system would find an acceptable match with the protocol signature before walking the list of expressions configured. This mistake is described in the example below:

```
entry 100 create
  description "Default HTTP Protocol"
  protocol eq "http"
  application "HTTP" This is an incorrect AppFilter order. App-filter entry
  no shutdown #100 will always match before the http expression entry
  exit #110.
entry 110 create
  description "Google"
  expression 1 http-host eq "*.google.com$"
  application "Google"
  no shutdown
exit
```

Note: It is not necessary to specify a protocol when defining an expression filter, the protocol is implicit based on the type of expression match criteria used (for instance, http, sip, h323).

## **App-Filters Ranges**

The App-Filter list is an ordered list, it is key to configure each app-filter in the right order and in the proper range.

The operator can customize the policy and create applications and app-filters by using the following ranges shown in Table 9 (other ranges are used by the Alcatel-Lucent default policy):

Range Name	Description	Start	End		
Top range	Top range, matches before any other filters	1	1499		
High priority	Matches before the other filters.	2000	3999		
Expression range A	HTTP Host, Host+URI ; optionally with IP/ Port match	19000	22999		

**Table 9: Customer Reserved App-Filter Ranges** 

Range Name	Description	Start	End		
Expression range B	Other Expression Match ; optionally with IP/ Port match	33000	34999		
Extended protocols	Protocol-signature + Port IP Dir. match	40000	41999		
Custom protocols	Custom protocol signature match	61000	61499		
Trusted/validate ports	1st packet validate, 1st packet trusted match	61500	61999		

Table 9: Customer Reserved App-Filter Ranges (Continued)

Ordering Basics:

- Layer 7 expressions based filters are located before their parent protocol signature (for example, expression matches on http are located before the http protocol app-filter; the same applies to TLS, SIP, H323, RTSP, Citrix).
- HTTP Host and URI are located before the HTTP referer for accounting accuracy (for example, YouTube<sup>®</sup> from within Facebook<sup>®</sup> is classified as YouTube<sup>®</sup>)
- App-filters combining protocol signatures with Layer 4 port, IP protocol, IP address or flow direction are always located before the protocol signature only filter range.

## HTTP

#### Protocol

HTTP is a client/server protocol using TCP/IP at the transport layer to deliver resources such as HTML files, images, videos and more.

HTTP 1.1 enables HTTP clients to use a persistent connection to a server allowing them to reuse the same TCP session for multiple HTTP transactions. Text, images, video, scripts and other objects can be downloaded individually in different transactions through the same TCP session.

Figure 204 describes a typical persistent HTTP connection between a web client and a server with multiple HTTP transactions within the same TCP session:



Figure 204: HTTP Persistent Connection

User-defined expression-based HTTP applications will use the first HTTP transaction to classify the flow (optionally this behavior can be modified).

#### HTTP

#### **HTTP Request**

The example below shows the content of a typical HTTP request to wikipedia.org which includes the following header fields: HTTP Host, HTTP URI, HTTP User Agent and HTTP referer fields:

Web Browser URL: http://en.wikipedia.org/wiki/Main\_Page

Host URI

HTTP Request Header Host: en.wikipedia.org URI: /wiki/Main\_Page User Agent: Mozilla/5.0(Windows NT 6.1; WOW64) Referer: http://www.google.com/

- HTTP Host Represents the domain name (does not include "http://").
- HTTP URI The URL trailer after the host domain name (begins with slash "/").
- HTTP Referer The address of the previous web page from which a link to the currently requested page was followed (in this example the referer is www.google.com which means the user clicked on a link from a Google search pointing to wikipedia.org).
- HTTP User Agent This identifies the web browser or application making the HTTP request.

## **Configuration Examples**

#### HTTP Host (Wikipedia)

Classifying HTTP traffic from this web site can be done using a single expression tail anchored on the HTTP host:

```
configure application-assurance group 1:1 policy app-filter
    entry <1..65535> create
        description "Wikipedia Web Access" expression 1 http-host eq "*.wikipedia.org$"
         application "Wikipedia"
         no shutdown
    exit
```

This can be confirmed using Wireshark<sup>®</sup>.

No	. Time	Source	Destination	Protocol	Info						
	149 4.474276	192.168.1.4	208.80.154.225	TCP	57881 > http [SYN] Seq-O Win-8192 Le						
	172 4.508432	208.80.154.225	192.168.1.4	TCP	http > 57881 [SYN, ACK] Seq-0 Ack-1						
	173 4.508543	192.168.1.4	208.80.154.225	TCP	57881 > http [ACK] Seq-1 Ack-1 Win-62						
	204 4.568615	192.168.1.4	208.80.154.225	HTTP	GET / HTTP/1.1						
	207 4.615704	208.80.154.225	192.168.1.4	2.168.1.4 TCP http > 57881 [ACk							
	208 4.615807	208.80.154.225	192.168.1.4	TCP	[TCP segment of a reassembled PDU]						
	209 4.615635	208.80.154.225	192.168.1.4	HTTP	HTTP/1.0 301 Moved Permanently						
	210 4.617685	192.168.1.4	208.80.154.225	TCP	57881 > http [ACK} Seq-956 Ack-614 w						
•											
+	Frame 204: 1039 byt	es on wire (8312 bits	), 1039 bytes capture	d (8312 bits)							
+	Ethernet II, Src: Hon	HaiPr_77:bf:c8 (4c:0f	:6e:77:bf:c8), Dst: Ne	tgear_d8:68:78 (c0:3f:0e	:d8:68:78)						
+	Internet Protocol, Sro	: 192.168.1.4 (192.1	68.1.4), Dst: 208.80.1	54.225 (208.80.154.225	)						
+	Transmission Contro	I Protocol, Src Port: 5	57881 (57881), Dst Po	ort: http (80), Seq: 1, Ack	: 1, Len: 985						
-	Hypertext Transfer P	rotocol									
	+ GET / HTTP/1.1\r	<u>\n</u>									
	Host: en.wikipedia	a.org\r\n									

Connection: keep-alive \r\n

User-Agent: Mozilla/5.0 (Windows NT 6,1: WOW64) AppleWebKit/537.4 (KHTML, like Gecko) Chrome/22.0.1229.

al\_0682W

#### Figure 205: Wireshark<sup>®</sup> www.wikipedia.org

## **Classification per URI within the Same Host**

Operators may need to apply different charging rules to different content located on the same HTTP domain (different URI, same HOST).

Table 10 displays an example of classification rules for the ISP ON-NET content services:

URL	Charging Rule	AA Application
www.ispdomain.com/video	Rule #1 – 0 Rating	ISP-Portal-Video
www.ispdomain.com/images	Rule #2 – Charge X	ISP-Portal-Images
www.ispdomain.com/*	Rule #3 – Charge Y	ISP-Portal-Default

Table 10: Classification Rules for the ISP ON-NET Content Services

HTTP 1.1 can reuse the same TCP connection for many transactions to the same server. Classifying each HTTP transaction to www.ispdomain.com independently requires a specific AA configuration.

Prior to SR OS 12.0.R1 the system can be configured to enable traffic classification for all http requests at the AA partition level only therefore affecting all HTTP flows within this partition. SR OS 12.0R1 allows to selectively enable "http-match-all-requests" in app-filters to improve the system performance and limit the HTTP analysis per domain.

The SROS 12.0.R1 configuration example below allows traffic classification of different URIs of the same domain (www.ispdomain.com) independently therefore allowing differentiated charging and control:

- http-match-all-req is enabled on all host+uri app-filters to www.ispdomain.com
- default app-filter required to match any traffic to www.ispdomain.com

```
configure application-assurance group 1:1 policy
    app-filter
    entry <1..65535> create
        description "Zero rated content"
        expression 1 http-host eq "^www.ispdomain.com$"
        expression 2 http-uri eq "^/video*"
        http-match-all-req
        application "ISP Portal Video"
        no shutdown
    exit
    entry <1..65535> create
        description "Image charging"
        expression 1 http-host eq "^www.ispdomain.com$"
        expression 1 http-host eq "^www.ispdomain.com$"
        expression 2 http-uri eq "^/images*"
        http-match-all-req
```

```
application "ISP Portal Images"
no shutdown
exit
entry <1..65535> create
  description "Default charging"
  expression 1 http-host eq "^www.ispdomain.com$"
  http-match-all-req
  application "ISP Portal Default"
  no shutdown
exit
```

## SSL/TLS (HTTPs)

#### Protocol

HTTPS uses SSL/TLS to encrypt traffic between the client and the server. Since this communication is encrypted it is not possible to identify the HTTP Host or URI. However, AA can still identify the service requested by the subscriber by looking at the TLS certificate information or Server Name Indication exchanged in the clear before the TLS session is established.

Note: SSL/TLS expression based app-filters are not limited to HTTPS. HTTPS is not a protocol in itself, but is HTTP traffic-tunnelled encrypted into SSL/TLS on port 443.

## SSL/TLS Certificates

The snapshot (Figure 206) from Wireshark shows the SSL/TLS certificate exchanged using the mobile application **whatsapp**<sup>®</sup>.

No.	Time	Source	Destination	Protocol	Info									
	42 44.854067	192.11.231.83	50.23.142.168	TCP	33084 > https [SYN] Seq-0 Win-64240 L									
	43 44.933347	50.23.142.168	192.11.231.83	TCP	https > 33084 [SYN, ACK] Seq-0 Ack-1									
	44 45.213335	192.11.231.83	50.23.142.168	0.23.142.168 TCP 33084 > https [ACK]										
	45 45.342530	192.11.231.83	50.23.142.168	SSLv3	Client Hello									
	46 45.448230	50.23.142.168	192.11.231.83	TCP	https > 33084 [ACK] Seq-1 Ack-75 Win-6									
	47 45.851643	50.23.142.168	192.11.231.83	SSLv3	Server Hello									
	48 45.853122	48 45.853122 50.23.142.168 192.11.231.83 TCP [TCP se												
	49 45.853231	50.23.142.168	192.11.231.83	TCP	[TCP segment of a reassembled PDU]									
	50 46.042243	192.11.231.83	50.23.142.168	TCP	33084 > https [ACK] Seq-75 Ack-2777 w									
	51 46.245518	192.11.231.83	50.23.142.168	TCP	33084 > https [ACK] Seq-75 Ack-4097 w									
	52 46.334985	50.23.142.168	192.11.231.83	SSLv3	Certificate, Server Hello Done									
•														
+	<ul> <li>[Reassembled TCP Segments (4686 bytes): #47(1309), #48(1388), #49(1320), #52(669)]</li> </ul>													
· S	Secure Socket Laye	er												
	SSLv3 Record L	ayer: Handshake Pr	otocol: Certificate											
	Content Type:	Handshake (22)												
	Version: SSL	3.0 (0x0300)												
	Length: 4672													
	<ul> <li>Handshake Pr</li> </ul>	otocol: Certificate												
	Handshake	Type: Certificate (1	1)											
	Length: 466	68												
	Certificates	Length: 4665												
	<ul> <li>Certificates</li> </ul>	(4665 bytes)												
	Certificat	e Length: 1377												
	<ul> <li>Certificat</li> </ul>	cid-at-commonnan	ne-*.whatsapp.net.pl-a	at-organizationalU	InitName-Domain Control validated, id									
	Certificat	e Length: 1250			al 0683									
		-			<i>u</i> _0005									

Figure 206: Wireshark<sup>®</sup> HTTPS www.whatsapp.com

The certificate information can be found in the Server Hello message sent by the server, capturing SSL/TLS (HTTPS) traffic from this application can be done using a single app-filter entry tail anchored on the TLS Common Name Certificate:

#### **Server Name Indication**

SSL/TLS traffic can optionally be identified using the Server Name Indication (SNI) which is an extension to the TLS protocol.

The SNI is found in the TLS Client Hello, the http-host expression in the app-filter is reused to classify this traffic:

No.	Time Source							Des	stina	tion			Prot	ocol		Info								
	4	0.08	8936	6	19	92.11	1.23′	1.82		98.	138.	6.52			TCF	•		iclpv-	nlc > https [SYN	] Seq-0 Win	-1			
`	5	0.16	5069	9	98	3.138	8.6.5	2		192	2.11.	231.	82		TCP			https	https > iclpv-nlc [SYN, ACK] Seq-0					
•	6	0.16	5136	6	19	92.11	.231	1.82		98.	138.	6.52			TCP			iclpv-	nlc > https ACK]	Seq-1 Ack	-1			
`	8	0.38	3867	7	19	2.11	.231	.82		98.	138.0	6.52			TLS	/1		Clien	t Hello					
•																								
	+	Cinł	her S	Suites	: (36	suit	es)																	
	_	Con	nnreg	ssior	n Met	thod	sle	nath	1															
	4	Con	nnre	ssior		thod	c (1	meth	(hou															
		Gunta Forda	ansio	one l	onat	h · 56	5 (1	<u> </u>	100)															
	/	Evto	neio	n	nyor	n. 50	) mo																	
	Extension: server_name																							
	~		ype.	(20 1			5 (0)	.0000	<i>y</i>															
	Data (30 bytes)     Extension: elliptic_curves																							
	<ul> <li>Extension: elliptic_curves</li> <li>Extension: ec_point_formats</li> </ul>																							
	<ul> <li>Extension: ec_point_formats</li> <li>Extension: SessionTicket TLS</li> </ul>																							
	+	EXIE	ensio	n. 5	essic	DITIC	cket	ILS																
0000	00	1e	e5	7a	96	5f	00	0c	29	7e	53	CC	08	00	45	00	Z	)~sE.						
0010	00	da	80	dS	40	00	80	06	69	2c	c0	0b	e7	52	62	8a	0	iRb.						
0020	06	34	05	72	01	bb	1f	6f	07	aS	3e	de	f1	43	50	18	.4.ro	>CP.						
0030	fc	00	6e	15	00	00	16	03	01	00	ad	01	00	00	a9	03	n							
0040	01	4d	80	1d	b4	c7	OC	86	06	8d	17	70	14	6c	85	ed	.M	p.1						
0050	ff	a3	30	5c	56	87	c3	09	98	d3	e0	b3	9e	a1	45	04	O\v	E.						
0060	S1	00	00	48	00	tt o t	c0	0a	c0	14	00	88	00	87	00	38	QH	8						
0070	c0	0f	c0	05	00	84	00	35	00	39	c0	07	c0	09	c0	11	5	.9						
0080	CU	13	00	45	00	44	00	33	00	32	CU	0C	CU	Ue	CU	02	E.D.3	.2						
0090	CU	04	00	96	00	41	00	04	00	05	00	21	CU	80	CU	12	A	/						
00a0	00	16	00	13	CU	00	CU	03	te	Π 75	00	0a	01	00	00	38								
0000	00	74	00	Te	00	1C	00	00	19	15	13	Ze	64 Cf	01	74	61	toolbox.	.us.data	<b>`</b>					
0000	-ze	-74 6d	00		00	02	00	06	Ze	17	00	100	00	10	Ze	03 0b		.yanoo.c	)					
0000	00	02	01	00	00	23	00	00	00	17	00	10	00	19	00	υD	UIII				al 0694			
0000	00	UΖ	01	00	00	23	00	00													ui_0084			

#### Figure 207: HTTPS SNI

```
configure application-assurance group 1:1 policy
    app-filter
    entry <1..65535> create
        description "Yahoo HTTP or TLS SNI"
        expression 1 http-host eq "*.yahoo.com$"
        application "Yahoo"
        no shutdown
        exit
```

## SIP

#### Protocol

SIP is a signaling protocol used for controlling multimedia communication sessions such as voice and video over RTP. AA automatically monitors SIP control flows and associates RTP/RTCP media flows accordingly in the sip\_rtp protocol signature.

The operator can use a SIP expression match criteria in app-filter to further refine traffic classification and identify any additional application on top of the default AA policy. This can be particularly useful in business VPNs to identify voice and telepresence applications.

AA supports SIP expression match criteria on SIP URI, SIP user agent and SIP media type. The snapshot below from Wireshark<sup>®</sup>shows a SIP control exchange using the voice-video application Vonage<sup>®</sup> followed by the RTP media audio flow; the expression fields that can be matched using AA app-filters are highlighted:

Session Initiation Protocol	í
Request-Line: INVITE sip:3102951568@k.voncp.com;transport=UDP SIP/2.0	
Method: INVITE	
Request-URI: sip:3102951568@k.voncp.com; transport=UDP	
Request-URI User Part: 3102951568	
Request-URI User Part: k.voncp.com	
[Resent Packet: False]	
Message Header	
From: "613-963-0148" <sip:16139630148@k.voncp.com>;tag=1019fb60-7196c445-2710-4e9485ff-7b9cb12-4e9485ff</sip:16139630148@k.voncp.com>	5ff
To: <sip:3102951568@k.voncp.com></sip:3102951568@k.voncp.com>	
Call-ID: 101a7de0-7196c445-2710-4e9485ff-229a8c45-4e9485ff@k.voncp.com	
CSeq: 1 INVITE	
Via: SIP/2.0/UDP 69.196.150.113:10000;branch=z9hG4bK-4e9485ff-f42b6c64-49ad5933	
P-Preferred-Identity: off	
Max-Forwards: 70	
Supported: replaces.timer.100rel	
User-Agent: VTA001346FE8BF111.4.1-r060815-1.00.09-20070402170142 1248967645135/1007551373 308	
Contact: <sip:16139630148@69.196.150.113:10000;transport=udp></sip:16139630148@69.196.150.113:10000;transport=udp>	
Min-SE: 0	
Content-Type: application/sdp	
Content-Length: 294	
Message Body	
Session Description Protocol	
Session Description Protocol version (v): 0	
Owner/Creator, Session Id (o): a0000 8644 6672 IN IP4 69.196.150.113	
Session Name (s): SIP Cal	
Connection Information (c): IN IP4 69.196.150.113	
Time Description, active time (t): 0 0	
Media Description, name and address (m): audio 10050 RTP/AVP 0 101 8 2 18	
Media Type: audio	
Media Port: 10050	al 0685
	<i>ai_0005</i>

Figure 208: SIP Wireshark<sup>®</sup> Capture

## **Configuration Example**

The configuration example below provides the configuration to classify  $Vonage^{\mbox{\ensuremath{\mathbb{R}}}}$  SIP/RTP desktop traffic using SIP URI expression:

```
configure application-assurance group 1:1 policy
    app-filter
    entry <1..65535> create
        description "Vonage"
        expression 1 sip-uri eq "*voncp.com*"
        application "Vonage"
        no shutdown
    exit
```

SIP

## H323

#### Protocol

Similarly to SIP, H323 is a signaling protocol used for controlling multimedia communication sessions such as voice and video over RTP. AA automatically monitors H323 control flows and associates the RTP media flow accordingly in the h323\_rtp protocol signature.

The operator can use an H323 expression match criteria app-filter to further refine traffic classification and identify any additional application on top of the default AA policy. This can be particularly useful in business VPNs to identify voice and telepresence applications.

AA supports H323 expression match criteria on the H323 Product ID. The snapshot below from Wireshark shows an H323 control exchange using the Telepresence application LifeSize<sup>®</sup> followed by the RTP media audio flow; the expression field that can be matched using AA app-filters is highlighted:

Transmission Control Protocol, Src Port: 61505 (61505), Det Port: h323hostcall (1720), Seq: 1, Ack: 1, Ler	i: 212
TPKT, Version: 3, Length: 212	
Q.931	
Protocol discriminator: Q.931	
Call reference value length: 2	
Call reference flag: Message sent from originating side	
Call reference value: 461a	
Message type: SETUP (0x05)	
Bearer capability	
Display 'Conference Room'	
- H.225.0 CS	
H323-UserInformation	
h323-uu-pdu	
h323-message-body: setup (0)	
setup	
protocolldentifier: 0.0.8.2250.0.5 (Version 5)	
sourceAddress: 3 items	
sourceInfo	
<ul> <li>vendor</li> </ul>	
✓ vendor	
H.221 Manufacturer: Unknown (0xb500a11a)	
productld: LifeSize Express 220	
versionId: 4.7.10.14	
0mc: False	
.0 undefinedNode: False	al_0686

Figure 209: H323 Wireshark<sup>®</sup> Capture

H323

### **Configuration Example**

The configuration example below provides the configuration to classify LifeSize<sup>®</sup> H323/RTP traffic using the H323 product ID expression:

```
configure application-assurance group 1:1 policy
   app-filter
    entry <1..65535> create
        description "LifeSize H323 traffic"
        expression 1 h323-product-id eq "^LifeSize*"
        application "LifeSize"
        no shutdown
        exit
```

## RTSP

#### Protocol

RTSP is a signaling protocol used for controlling media streaming content such as audio and video over RTP/RDT. AA automatically monitors the RTSP control flows and associates its RTP/RDT media flow with the rtp\_rtsp protocol signature.

The operator can use an RTSP expression match criteria app-filter to further refine traffic classification and identify any additional application on top of the default AA policy. This can be particularly useful to identify specific streaming applications.

AA supports RTSP expression match criteria on the RTSP Host, URI, UserAgent. The snapshot below from Wireshark<sup>®</sup> shows an RTSP setup request to YouTube<sup>®</sup> followed by the RTP media audio flow; the expression fields that can be matched in RTSP SETUP request using AA app-filters are highlighted:

HostURLRTSP Header<br/>SETUP rtsp://v3.cache7.c.youtube.com/ZTww=/0/0/0/video.3gp/trackID=13 RTSP/1.0<br/>CSeq: 3User-Agent: Mozilla/5.0 (BlackBerry; U; BlackBerry 9800; en) AppleWebKit/54.8+<br/>x-wap-profile: "http://www.blackberry.net/go/mobile/profiles/uaprof/9800\_unknown/<br/>6.0.0.rdf"<br/>Transport: RTP/AVP;unicast;client\_port=51132-51133;mode="PLAY"

#### **Configuration Example**

The configuration example below provides the configuration to classify YouTube<sup>®</sup> RTSP/RTP traffic using RTSP Host expression:

```
configure application-assurance group 1:1 policy
   app-filter
      entry <1..65535> create
        description "YouTube RTSP/RTP Video"
        expression 1 rtsp-host eq "*.youtube.com$"
        application "YouTube"
        no shutdown
   exit
```

7750 SR Advanced Configuration Guide

## Citrix

## Protocol

Independent Computing Architecture (ICA) is a Citrix Systems<sup>®</sup> protocol used in Citrix's WinFrame, Citrix XenApp (formerly called MetaFrame/Presentation Server), and Citrix XenDesktop products.

Citrix makes it possible to run applications remotely on large servers, thus making better use of server resources while at the same time allowing people using other platforms to use the applications, for example, run Microsoft<sup>®</sup> Word on a UNIX workstation.

Citrix\_ica protocol signature will detect any remote application using Citrix (the protocol needs to be unencrypted and configured to non-seamless). The Citrix ICA session is started from a client and can be anything from Remote Desktop, SAP to Microsoft<sup>®</sup> Word.

The Citrix expression match app-filter is used to classify traffic based on the Citrix-published application. This published application is configured on the server and in the example above can be for instance RDP, SAP, Word, XLS or Microsoft<sup>®</sup> Word depending how the server is configured.

### **Configuration Example**

```
configure application-assurance group 1:1 policy
    app-filter
    entry <1..65535> create
        description "Citrix SAP Application"
        expression 1 citrix-app eq "SAP"
        application "Citrix SAP"
        no shutdown
    exit
```

## **IP Address and TCP/UDP Port**

Traffic from specific server(s) can be classified using IPv4/v6 server-address app-filter rules. It is used usually to identify traffic from an internal (on-net) server as opposed to an Internet (off-net) server.

The server-address app-filter automatically detects the client from the server by identifying which side opens the connection. It implicitly classifies traffic based on the server IP address or Port number. For example, if A initiates a TCP connection to B, then flows A->B and B<-A can be classified with a match on server-address = B. Similarly a flow initiated from B to A would be classified using a match on server-address = A.

#### **Server Address**

The configuration example below uses a server-address app-filter to classify traffic from server 10.1.1.1 in the application called Application-1:

```
configure application-assurance group 1:1 policy
   app-filter
      entry <1..65535> create
        description "Server #1 10.0.0.1"
        server-address eq 10.0.0.1/32
        application "Application-1"
        no shutdown
   exit
```

IP Address and TCP/UDP Port

#### Server-Address + Server Port

The configuration example below uses server-address and server-port app-filters to classify traffic from server 10.0.0.2 on port 1234 in the application called Application-2. It is particularly useful when the same server is used to provide different services that need to be classified separately:

```
configure application-assurance group 1:1 policy
    entry <1..65535> create
        description "Server #2 10.0.0.2 port 1234 Only"
        server-address eq 10.0.0.2/32
        server-port eq 1234
        application "Application-2"
        no shutdown
    exit
```

#### Server Port and Protocol Signature

It is possible to combine a protocol signature with a port number in the same app-filter, this is typically done in business VPNs for specific internal applications not detected using existing AA protocol signatures.

The configuration below classifies a business VPN application running on TCP port 4000 and not detected by any other signatures. It combines the protocol signature unknown\_tcp with the desired port number. This allows keeping the classification untouched for the rest of the protocols/ applications and is the recommended approach:

```
configure application-assurance group 1:1 policy
   app-filter
    entry <1..65535> create
        description "Business VPN Application X Port 4000"
        server-port eq 4000
        protocol eq unknown_tcp
        application "Busines VPN Application X"
        no shutdown
    exit
```

Note: It is important to follow the app-filter range recommendations for a proper classification of traffic using IP address or port number.

#### **Flow Setup Direction**

Traffic can be classified based on flow-setup-direction app-filter. The flow setup direction can be either subscriber-to-network or network-to-subscriber.

Network side and subscriber side is AA terminology related to where AA is enabled:

- In broadband and mobile networks, AA is enabled per subscriber. This means the subscriber side represents the ESM/mobile/transit subscriber while the network side represents Internet or other subscribers.
- In business VPNs, AA is enabled on a VPN SAP/spoke SDP and the subscriber side represents the local VPN site (SAP/spoke/transit).

The example below shows the configuration to classify http traffic hosted by AA subscribers (for example, broadband subscribers running a web server):

```
configure application-assurance group 1:1 policy
   app-filter
    entry <1..65535> create
        description "HTTP Server on the subscriber side"
        flow-setup-direction network-to-subscriber
        protocol eq http
        application "HTTP Server"
        no shutdown
        exit
```

#### **IP Protocol**

Traffic can be classified using an IP protocol number for non TCP/UDP traffic.

The example below example provides the configuration to classify ICMP IPv4/v6 traffic:

```
configure application-assurance group 1:1 policy
   app-filter
       entry <1..65535> create
           description "ICMP v4"
           protocol eq "non_tcp_udp"
           ip-protocol-num eq icmp
           application "ICMP"
           no shutdown
        exit
        entry <1..65535> create
           description " ICMP v6"
           protocol eq "non tcp udp"
           ip-protocol-num eq ipv6-icmp
           application "ICMP"
           no shutdown
        exit
```

7750 SR Advanced Configuration Guide

### **Custom Protocol**

Custom protocols can be used to classify TCP/UDP applications using hexadecimal string matching (up to 16 hex octets) at a configurable payload offset in the data payload. The expression string length and offset must not exceed 128 bytes.

To illustrate this feature the Solaris<sup>®</sup> application GoGlobal is used. It provides remote access to a server (similar to  $VNC^{\mathbb{R}}$ ). The snapshot below (Figure 208) from Wireshark<sup>®</sup> shows a TCP SYN/ ACK session establishment followed by the first data exchange:

	1 (	0.00	000	0		13	8.2	03.4	0.20	)1		138.	203	.19.2	243		TCF	m	xxrloo	in >	go-lo	gin	[SYN	1]	
	2 (	0.00	091	5		13	8.2	03.1	9.24	13		138.	203	.40.2	201	-	TCF	g	o-logir	ו > n	nxxrlo	gin	- [SYN	1]	
	3 (	0.00	092	7		13	8.2	03.4	0.20	)1		138.	203	.19.2	243	-	TCF	m	xxrlog	jin >	go-lo	gin	[ACk	[]	
	4 0.001068 138.203.40.201									138.	203	.19.2	243	-	TCF	m	xxrloc	jin >	go-lo	gin	[PSF	ł,			
	5 (	0.00	195	0		13	8.20	03.1	9.24	13		138.	203	.40.2	201	-	TCF	go	o-logir	ז > n	nxxrlo	gin	[ACk	[]	
	6 (	0.01	676	9		13	8.20	03.1	9.24	13		138.	203	.40.2	201	-	TCF	g	o-logir	n > n	nxxrlo	gin	[PSF	1.	~
◄																									
<ul> <li>Frame 4 (58 bytes on wire, 58 bytes captured)</li> </ul>																									
+	Ethernet II, Src: Vmware 43:73:71 (00:0c:29:43:73:71), Dst: TimetraN 01:1a:00 (00:03:																								
+	<ul> <li>Internet Protocol. Src: 138.203.40.201 (138.203.40.201). Dst: 138.203.19.243 (138.203</li> </ul>												3												
+	Tran	smi	ssi	on	Cor	trol	Pre	oto	col,	Sro	Pc	ort:	mx>	rlog	gin	(10;	35)	Dst	Port	: go	-logii	n (4	91),		
-	Data	(4	byt	es)																					
	D	ata:	80	DC	040	0																			
	[L	eng	jth:	4]																					_
◄																								D	רי
000	0 00	03	fa	01	1a	00	00	0c	29	43	73	71	08	00	45	00			. )Csc	E					
001	0 00	2c	1f	b7	40	00	80	06	88	c2	8a	cb	28	с9	8a	cb		.©		(					
002	0 13	f3	04	0b	01	eb	d3	fa	42	15	4b	f4	23	ed	50	18			B.K	.#. F	2.				
003	0 ff	ff	52	71	00	00	80	dc	04	00							I	₹q							
																							ai	068	37



Wireshark<sup>®</sup> shows that each TCP session payload starts with 80DC0400 (no offset) after the threeway TCP handshake, as a result the configuration required to classify this traffic is described below:

```
configure application-assurance group 1:1 policy
  custom-protocol 1 ip-protocol-num tcp create
    description "goglobal tcp"
    expression 1 eq "\x80\xdc\x04\x00" offset 0 direction client-to-server
    no shutdown
exit
    app-filter
    entry <1..65535> create
        description "GoGlobal "
        protocol eq "custom_01"
        application "GoGlobal"
        no shutdown
    exit
```

## **Typical Configuration Mistakes**

An operator creating new user-defined applications can make a few typical mistakes which are listed below:

- App-filters in shutdown state The default app-filter state is shutdown. A no shutdown command must be executed in order for it to be enabled.
- App-filters with no match criteria This is a more troublesome mistake as it will catch all the traffic entering the filter in a particular application.

## **Troubleshooting Application Identification**

#### Show Commands

#### **Router/Partition Statistics**

Partition level statistics are not updated in real time. Instead, statistics for a particular flow are updated either at flow closure or every five minutes. The fiveminute sliding window interval is a common interval for all flows in a given ISA MDA. Different ISA MDAs will have a different five minute windows as this interval is set at the MDA boot time.

The following command can be used to view the statistics for all applications configured in the ISA Group 1, Partition 1:

show application-assurance group 1:1 application count

Alternatively it is possible to sort the display by octets, packets, flows:

```
show application-assurance group 1:1 application count top [octets|packets|flows] [max-count <max-count>]
```

The operator can also identify which app-filters are being hit by the AA policy per partition (this command is not available per subscriber), it is particularly useful to identify which filters are used and optionally prune unnecessary app-filters from user-defined applications:

show application-assurance group 1:1 policy app-filter

Note: The app-filter policy is usually relatively large, in which case additional 7x50 SR CLI functionality can be used to filter out the output and only show the relevant information. The example below was created for the application FTP:

```
A:PE# show application-assurance group 1:1 policy app-filter | match "application \"FTP\""

pre-lines 3 post-lines 2

exit

entry 44300 create (2 flows, 1205 B)

protocol eq "ftp_control"

application "FTP"

no shutdown

exit

entry 44301 create (2 flows, 1401 B)

protocol eq "ftp_data"

application "FTP"

no shutdown

exit
```

Because partition level statistics are not updated in real time it is recommended for troubleshooting purposes to use subscriber statistics or sub-study statistics.

#### **Subscriber Statistics**

Subscriber level statistics can be updated in real time. AA is usually configured by the Operator to collect subscriber level statistics for all application groups in residential and Wifi, while business VPNs typically collect Application group and all applications for each site with AA enabled.

The commands below can be used to view per subscriber statistics for all app-groups or applications configured in ISA Group 1, Partition 1 for the ESM subscriber "Bob" or business VPN SAP 1/1/1:10:

show application-assurance group 1:1 aa-sub esm "bob" app-group count show application-assurance group 1:1 aa-sub sap 1/1/1:10 application count

In case only app-group statistics are collected per subscriber, the aa-sub-study feature can be used to collect per application level statistics for selected subscribers, see configuration example below:

```
A:PE# configure application-assurance group 1:1 statistics aa-sub-study application A:PE>config>app-assure>group>statistics>aa-sub-study# aa-sub esm "bob"
```

Once done, the system will show all application level statistics for this subscriber:

show application-assurance group 1:1 aa-sub-study esm "bob" application count

Similarly to partition level statistics, aa-sub and aa-sub-study statistics can be sorted by octets, packets, flows:

show application-assurance group 1:1 aa-sub-study esm "bob" application count top [octets|packets|flows] [max-count <max-count>]

Note: When the number of flows per ISA card reaches a threshold then per subscriber statistics are not available in real time anymore and only the snapshot command can be used to display the statistics recorded in the previous 5 minute interval window:

show application-assurance group 1:1 aa-sub-study esm "bob" snapshot application count

#### **AppFilterMiss**

The default policy configuration provides a failsafe application at the very end of the app-filter list to classify any remaining traffic in the AppFilterMiss application. There should never be any traffic in this application. This failsafe filter is used as a debug to make sure that there are no major issues in the configuration.

Note: Traffic can typically be classified as AppFilterMiss when not all protocol signatures are mapped to a particular application. This could happen when upgrading to a new ISA software and enabling new protocol signature detection while not ensuring first that the correct application was provisioned. See the 7x50 SR Release Note upgrade section for more details on AA signature upgrade.

## Tools

#### Flow-Record-Search

Traditional show commands may not provide enough information when troubleshooting flow identification and the operator can use the ISA flow-record-search tool to dump the ISA flow table for more information. This feature comes with a large number of filtering options documented in the user guide.

Each flow gives visibility into: Flow ID, Sub-Type, Sub-Name, Initiator, Direction, Source IP, Dest. IP, IP Protocol, Source Port, Dest. Port, FC, DSCP, Classified, Protocol, Application, App-Group, Charging Group, Packets tx, Bytes Tx, Packets-discarded, Bytes-discarded etc.

See below for the most commonly used commands.

To show all the flows in a given ISA card per ISA group:partition (can be a very long output, up to 3M entries):

tools dump application-assurance group 1:1 flow-record-search is a 1/2

To show all the flows per AA subscriber in a given group:partition:

tools dump application-assurance group 1:1 flow-record-search aa-sub esm "bob"

To show all the active flows per AA subscriber in a given group:partition:

tools dump application-assurance group 1:1 flow-record-search aa-sub esm "bob" flow- status active

The flow-record-search command is also available with additional details by adding search-type detail at the end of the command line. Note that due to the length of the output it is recommended to paste the CLI output content in a notepad file.

**HTTP Host Recorder** 

SR OS 11.0.R1 introduced the comprehensive cflowd feature which allows AA to export the HTTP domain extracted from HTTP flows to the 5670 RAM reporting solution. As such, it is the preferred functionality to understand which HTTP Hosts are visible in the network.

Prior to SR OS 11.0.R1, the HTTP host recorder is a tool function available in the 7750 to record HTTP Hosts seen by AA. See the 7750 SR OS Multi-Service Integrated Services Adapter Guide for more details.

```
A:PE# show debug
debug
   application-assurance
       group 1:1
           http-host-recorder
               filter
                    default-filter-action record
                    record http-host-app-filter-candidates
                exit
                rate 100
               no shutdown
           exit
       exit
    exi†
exit
A:PE# tools dump application-assurance group 1:1 http-host-recorder top bytes
```

#### **Port Recorder**

This function is particularly useful in business VPN (it can also be used in residential networks). The port-recorder AA tool function is similar to the http-recorder. It allows the operator to record which ports are used on selected applications.

It is most commonly used with the applications Unidentified TCP and Unidentified UDP but it can be configured to record any other applications:

```
A:PE# show debug

debug

application-assurance

group 1:1

port-recorder

application "Unidentified TCP"

application "Unidentified UDP"

rate 100

shutdown

exit

exit

exit

exit

A:PE# tools dump application-assurance group 1:1 port-recorder top bytes
```

Tools

## Conclusion

This example, which is intended for Application Assurance (AA) network architects and engineers, provides the information required to modify an existing AA policy following AA best practices and guidelines, and provides the necessary troubleshooting information to better understand application classification using Application Assurance.

Conclusion