IBM Security QRadar
Version 7.1.0 MR2

AQL Flow and Event Query Guide

Note: Before using this information and the product that it supports, read the information in Notices and Trademarks on page 21.

CONTENTS

ABOUT THIS GUIDE

Intended audience 1
Documentation conventions 1
Technical documentation 1
Contacting customer support 1
Statement of good security practices 2

1 THE AQL QUERY COMMAND-LINE INTERFACE

About the AQL command-line interface (CLI) 3
Accessing AQL from the command-line 4
Search for event or flow data in AQL 5
Identifying AQL fields 16
Querying for event or flow field names 17

A NOTICES AND TRADEMARKS

Notices 21
Trademarks 23

About This Guide

The AQL Event and Flow Query CLI Guide provides you with information for using the AQL command-line shell. This guide assumes you have advanced knowledge of Linux command line functionality.

Intended audience

This guide is intended to inform users of the commands required to view event or flow data stored in the Ariel database.

Documentation conventions

The following conventions are used throughout this guide:

Note: Indicates that the information provided is supplemental to the associated feature or instruction.

CAUTION: *Indicates that the information is critical. A caution alerts you to potential loss of data or potential damage to an application, system, device, or network.*

WARNING: *Indicates that the information is critical. A warning alerts you to potential dangers, threats, or potential personal injury. Read any and all warnings carefully before proceeding.*

Technical documentation

For information on how to access more technical documentation, technical notes, and release notes, see the ***Accessing IBM Security QRadar Documentation Technical Note***.
(http://www.ibm.com/support/docview.wss?rs=0&uid=swg21614644)

Contacting customer support

For information on contacting customer support, see the ***Support and Download Technical Note***.
(http://www.ibm.com/support/docview.wss?rs=0&uid=swg21612861)

Statement of good security practices

IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM DOES NOT WARRANT THAT ANY SYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY.

1 THE AQL QUERY COMMAND-LINE INTERFACE

You can use the AQL Event and Flow Query Command Line Interface (CLI) to access flows and events stored in the Ariel database on your QRadar Console.

The AQL shell is a read-only interface for viewing events or flows based on the time they were written to disk. This interface does not support data imports for event or flow data.

About the AQL command-line interface (CLI)

The AQL Event and Flow Query CLI allows you to access raw flows and events stored in the Ariel database. The AQL query CLI includes syntax that is a subset of the SQL92 standard and provides support for two tables: events and flows.

Note: The AQL CLI does not provide support for joining tables.

The AQL Event and Flow Query CLI functions in the following modes:

- **Interactive mode** - This is the default mode. Using a command-line shell, you can enter queries interactively and view the results in a standard output. At the query prompt, any valid AQL statement is accepted. If time is not specified (using -start and -end options), the last minute is assumed as the time range. The start and end time for all searches is based on when Ariel writer wrote the event or flow to the disk. You can access previous queries by using the Up arrow key.
- **Non-interactive mode** - You can enter the non-interactive mode by adding the -execute <AQL query> parameter to the command. The -execute command must be followed by a valid AQL query surrounded by double quotes. Non-interactive mode does not include a prompt enabling you to redirect the output to a file using regular UNIX pipe syntax. By default, the results are sent to a standard output.

Accessing AQL from the command-line

You can access the AQL shell from the command-line of your QRadar Console.

Procedure

Step 1 Using SSH, log in to your QRadar Console as the root user.

Username: **root**

Password: **<password>**

Step 2 Type the following command to start the arielClient shell:

```
/opt/qradar/bin/arielClient
```

Step 3 Optional. Type any additional command-line options to restrict the results returned by your AQL query.

Optional query command are list in **Table 1-1**.

Step 4 Press Enter to start the AQL shell.

The shell is active when the command-line prompt displays Query >>. You must exit the AQL shell to return to the standard command-line interface.

Step 5 Type a select statement to query for event or flow data.

Step 6 Type **exit** to leave the AQL shell.

Optional command-line search parameters

You can enhance the results returned by your AQL query by defining parameters for your search results.

Table 1-1 AQL query result options

Option	Description
-range <first record> <last record>	Restricts the number of records sent to the output within the specified range. This is used for viewing a selection of records generated by an ordered query. For example, if you want to view the first ten records, you must specify **-range 1 10**.
-debug	Generates debugging output during execution.
-start <time>, -end <time>	Specifies the start and end time of the query. Where **<time>** specifies the time. You must specify the time as either a UNIX timestamp or a date using the following format: yyyy/mm/dd-hh:mm:ss. For example: **/opt/qradar/bin/arielClient - start 2012/08/11-01:15:00 -end 2012/08/11-01:17:00** The start and end times for queries are based on the time the events were written to the disk. If you are using the Store and Forward feature your AQL query start and end time should reflect the start and end time of your forwarding schedule.
-exectime <time limit>	Specifies the maximum period of time, in seconds, a single query will continue processing.

Table 1-1 AQL query result options (continued)

Option	Description
-execute <AQL query>	Enables non-interactive mode that is used to process a query that is sent to standard output. The query must include double quotes. If you do not include this option, the command is entered in interactive mode.
-f <output format>	Specifies the output format for the query results. The table format is an ASCII drawing of a multi-column table while the csv format provides a comma separated list. Where **<output format>** indicates the output format. The options are **table** or **csv**.
-remote <host:port>	Specifies the connection to a specific Ariel query host and port.

Examples

To enter a command in interactive mode:

```
/opt/qradar/bin/arielClient -start 2012/08/11-01:15:00 -end
2012/08/11-01:17:00 -exectime 60
```

```
/opt/qradar/bin/arielClient
```

```
/opt/qradar/bin/arielClient -start 2012/08/11-01:15:00 -end
2012/08/11-01:17:00  -f csv
```

To enter a command in non-interactive mode:

```
/opt/qradar/bin/arielClient -start 2012/08/11-01:15:00 -end
2012/08/11-01:17:00 -exectime 60 -execute "select * from flows
where sourceIP = '231.12.37.17' and protocol != 'TCP.tcp_ip'"
```

Search for event or flow data in AQL

The AQL shell allows you to use select statements to query specific data from the events or flows table in the Ariel database.

Select statement examples

You can use a select statement that includes one or more fields from the flow or event tables. You can also use an asterisk (*) to denote all columns. All field names are case sensitive, however, the terms select and from are not case sensitive.

The full list of event and flow fields that can be queried by the AQL shell is listed in **Table 1-2**.

The idlist.sh shell script can provide users with a list of the most commonly used event and flow fields, along with their data types. The use of the describe command is used. For more information on using the describe command, see **Querying for event or flow field names**.

Examples

```
select sourceIP, destinationIP, application from flows where
protocol = ‘TCP.tcp_ip’
```

```
select category, credibility from events where severity > 8
select * from events where credibility >=9
```

Table 1-2 Supported flow and event fields

Table	Field Name	Description	Data Type	Visible using describe
Flows	anyDestinationFlag	Destination Flags	Numeric	No
	anySourceFlag	Source Flags	Numeric	No
	application	Application	String	Yes
	applicationId	Application	Numeric	Yes
	bytesIn	Bytes In	Numeric	No
	bytesOut	Bytes Out	Numeric	No
	destinationAssetName	Destination Asset Name	String	No
	destinationASN	Destination ASN	Numeric	Yes
	destinationBytes	Destination Bytes	Numeric	Yes
	destinationByteRatio	Destination Byte Ratio	Numeric	No
	destinationDSCP	Destination DSCP	Numeric	Yes
	destinationDscpOnly	Destination DSCP	Numeric	No
	destinationFlags	Destination Flags	Numeric	Yes
	destinationIP	Destination IP	String	Yes
	destinationIPSearch	Destination IP Search	String	No
	destinationIfIndex	Destination If Index	Numeric	Yes
	destinationNetwork	Destination Network	String	Yes
	destinationPackets	Destination Packets	Numeric	Yes
	destinationPacketRatio	Destination Packet Ratio	Numeric	No
	destinationPayload	Destination Payload	String	Yes
	destinationPayloadHex	Destination Payload As Hex	Hexadecimal	No
	destinationPort	Destination Port	Numeric	Yes
	destinationPrecedence	Destination Precedence	Numeric	Yes
	destinationPrecedanceOnly	Destination Precedence	Numeric	No
	destinationTOS	Destination QoS	Composite	Yes
	destinationv6	IPv6 Destination	Hexadecimal	No
	firstPacketTime	First Packet Time	Numeric	Yes
	flowBias	Flow Bias	String	Yes
	flowDirection	Flow Direction: • Local-to-Local (L2L) • Local-to-Remote (L2R) • Remote-to-Local (R2L) • Remote-to-Remote (R2R)	String	Yes

Table 1-2 Supported flow and event fields (continued)

Table	Field Name	Description	Data Type	Visible using describe
	flowSource	Flow Source	Numeric	Yes
	flowType	Flow Type	Numeric	Yes
	geographic	Matches Geographic Location	String	Yes
	hasDestinationPayload	Has Destination Payload	Boolean	No
	hasSourcePayload	Has Source Payload	Boolean	No
	icmpType	ICMP Type/Code	Numeric	Yes
	interface	Flow Interface	String	Yes
	intervalId	Interval ID	Numeric	Yes
	lastPacketTime	Last Packet Time	Numeric	Yes
	packetsIn	Packets In	Numeric	No
	packetsOut	Packets Out	Numeric	No
	protocol	Protocol	String	Yes
	protocolId	Protocol	Numeric	Yes
	remoteHost	Remote Host	String	No
	remoteNet	Matches Remote Network	String	No
	remoteServices	Matches Remote Service	String	No
	sourceASN	Source ASN	Numeric	Yes
	sourceAssetName	Source Asset Name	String	No
	sourceByteRatio	Source Byte Ratio	Numeric	No
	sourceBytes	Source Bytes	Numeric	Yes
	sourceDSCP	Source DSCP	Numeric	Yes
	sourceDscpOnly	Source DSCP	Numeric	No
	sourceFlags	Source Flags	Numeric	Yes
	sourceIP	Source IP	String	Yes
	sourceIPSearch	Source IP Search	String	No
	sourceIfIndex	Source If Index	Numeric	Yes
	sourceNetwork	Source Network	String	Yes
	sourceOrDestinationIP	Source or Destination IP	String	No
	sourcePackets	Source Packets	Numeric	Yes
	sourcePacketRatio	Source Packet Ratio	Numeric	No
	sourcePayload	Source Payload	String	Yes
	sourcePayloadHex	Source Payload As Hex	Hexadecimal	No
	sourcePort	Source Port	Numeric	Yes
	sourcePrecedence	Source Precedence	Numeric	Yes
	sourcePrecedanceOnly	Source Precedence	Numeric	No

Table 1-2 Supported flow and event fields (continued)

Table	Field Name	Description	Data Type	Visible using describe
	sourceTOS	Source QoS	Composite	Yes
	sourcev6	IPv6 Source	Hexadecimal	No
	token	Associated With Offense	Numeric	Yes
	totalBytes	Total Bytes	Numeric	No
	totalPackets	Total Packets	Numeric	No
	viewObjectPair	View/Object	String	No
Events	category	Low Level Category	Numeric	Yes
	creEventList	Matched Custom Rule	String	No
	credibility	Credibility	Numeric	Yes
	destinationAssetName	Destination Asset Name	String	No
	destinationIP	Destination IP	String	Yes
	destinationMAC	Destination MAC	Hexadecimal	Yes
	destinationNetwork	Destination Network	String	Yes
	destinationPort	Destination Port	Numeric	Yes
	destinationv6	IPv6 Destination	Hexadecimal	No
	device	Log Source	Numeric	Yes
	deviceGroup	Log Source Group	Numeric	Yes
	deviceTime	Log Source Time	Numeric	No
	deviceType	Log Source Type	Numeric	Yes
	duration	Duration	Numeric	Yes
	endTime	End Time	Numeric	Yes
	eventCount	Event Count	Numeric	Yes
	eventDirection	Event Direction: • Local-to-Local • Local-to-Remote • Remote-to-Local • Remote-to-Remote	String	Yes
	eventProcessor	Event Processor	Numeric	Yes
	hasIdentity	Has Identity	Boolean	Yes
	hasOffense	Associated With Offense	Boolean	Yes
	highLevelCategory	High Level Category	Numeric	Yes
	isCREEvent	Is CRE Event	Boolean	No
	identityExtendedField	Identity Extended Field	String	Yes
	identityGroupName	Identity Group Name	String	Yes
	identityHostName	Identity Host Name	String	Yes

Table 1-2 Supported flow and event fields (continued)

Table	Field Name	Description	Data Type	Visible using describe
	identityIP	Identity IP	String	Yes
	identityMAC	Identity MAC	Hexadecimal	Yes
	identityNetBiosName	Identity NetBIOS Name	String	Yes
	identityUserName	Identity User Name	String	Yes
	magnitude	Magnitude	Numeric	Yes
	payload	Payload	String	Yes
	payloadHex	Payload As Hex	Hexadecimal	No
	postNatDestinationIP	Post NAT Destination IP	String	Yes
	postNatDestinationPort	Post NAT Destination Port	Numeric	Yes
	postNatSourceIP	Post NAT Source IP	String	Yes
	postNatSourcePort	Post NAT Source Port	Numeric	No
	preNatDestinationIP	Pre NAT Destination IP	String	Yes
	preNatDestinationPort	Pre NAT Destination Port	Numeric	Yes
	preNatSourceIP	Pre NAT Source IP	String	Yes
	preNatSourcePort	Pre NAT Source Port	Numeric	Yes
	protocol	Protocol	String	Yes
	qid	Event Name ID	Numeric	Yes
	relevance	Relevance	Numeric	Yes
	severity	Severity	Numeric	Yes
	sourceAssetName	Source Asset Name	String	No
	sourceIP	Source IP	String	Yes
	sourceMAC	Source MAC	Hexadecimal	Yes
	sourceNetwork	Source Network	String	Yes
	sourcePort	Source Port	Numeric	Yes
	sourcev6	IPv6 Source	Hexadecimal	No
	startTime	Start Time	Numeric	Yes
	token	Token Associated With Offense	Numeric	Yes
	unparsed	Event Is Unparsed	Boolean	Yes
	userName	Username	String	Yes

Additional examples

You can also use CIDR-based queries using the select statement. To query by source IP address (sourceIP) or by destination IP address (destinationIP) using a CIDR, use the following format:

```
select <query item> from <flows|events> where
<sourceCIDR|destinationCIDR> = '<CIDR Range>'
```

For example:

```
select * from flows where sourceCIDR = '10.100.100/24'
```

To return all flows coming from the 10.100.100 subnet or capture flows coming from and into the subnet, use the regular OR expression as follows:

```
select * from flows where sourceCIDR = '10.100.100/24' OR
destinationCIDR = '10.100.100/24'
```

You can use the following statements to filter AQL queries.

Table 1-3 Fields that use the any statement type

Table	Field Name	Description	Data Type
Flow	anyASN	Source or Destination ASN	Numeric
	anyHost	Source or Destination IP	String
	anyNetwork	Source or Destination Network	String
	destinationTOS	Destination QoS	Composite
	icmpType	ICMP Type/Code	Numeric
	sourceOrDestinationIP	Source or Destination IP	String
	sourceTOS	Source QoS	Composite
Events	anyIP	Any IP	String
	anyMac	Source or Destination MAC Address	Hex
	anyPort	Any Port	Numeric
	sourceOrDestinationIP	Source or Destination IP	String
	sourceOrDestinationNetwork	Source or Destination Network	String
	sourceOrDestinationPort	Source or Destination Port	Numeric

Where clause examples

You can restrict your AQL queries using **where** clauses. The supported logical operators in the clause include **and**, **OR**, and parentheses. AQL queries also support the following relational operators: **, =, <, >, >=, <=,** and **!=.**

Examples

```
select sourceIP, category, credibility from events where
severity > 9 and category = 5013
```

```
select sourceIP, category, credibility from events where
(severity > 9 and category = 5013) or (severity < 5 and
credibility > 8)
```

The where clause also supports the arieltime variable, which overrides the time settings passed to the AQL CLI. The arieltime variable must be used with the between keyword to specify the start and end time bounds of the query. All time constraints must be entered as either UNIX timestamps or formatted date or time

strings. The results returns are based on the time that the event or flow was written to the Ariel database.

You can only use the arieltime variable once in a single query. Therefore, you can only query a continuous span of time in a single AQL command.

The logical operator for the arieltime variable and the remainder of the where clause should be the and operator. We recommend that you use the arieltime variable as the last constraint of the query and the and operator between the arieltime variable and the rest of the where clause.

Group by clause examples

You can use the group by clause to aggregate your data. Normally, data aggregation is combined with arithmetic functions on remaining columns to provide meaningful results of the aggregation.

Examples

To enter a query to investigate the IP addresses that sent more than 1 million bytes within all flows in a specific time frame, you must enter:

```
select sourceIP, SUM(sourceBytes) from flows where sourceBytes >
1000000 group by sourceIP
```

The output resembles:

```
----------------------------------
| sourceIP       | SUM_sourceBytes |
----------------------------------
| 64.124.201.151 | 4282590.0       |
| 10.105.2.10    | 4902509.0       |
| 10.103.70.243  | 2802715.0       |
| 10.103.77.143  | 3313370.0       |
| 10.105.32.29   | 2467183.0       |
| 10.105.96.148  | 8325356.0       |
| 10.103.73.206  | 1629768.0       |
----------------------------------
```

However, if you compare this information to a non-aggregated query, the output displays all the IP addresses that are unique:

```
select sourceIP, sourceBytes from flows where sourceBytes >
1000000
```

```
-----------------------------
| sourceIP       | sourceBytes |
-----------------------------
| 64.124.201.151 | 1448629     |
| 10.105.2.10    | 2412426     |
| 10.103.70.243  | 1793095     |
| 10.103.77.143  | 1449148     |
| 10.105.32.29   | 1097523     |
| 10.105.96.148  | 4096834     |
| 64.124.201.151 | 2833961     |
| 10.105.2.10    | 2490083     |
| 10.103.73.206  | 1629768     |
| 10.103.70.243  | 1009620     |
| 10.105.32.29   | 1369660     |
| 10.103.77.143  | 1864222     |
| 10.105.96.148  | 4228522     |
-----------------------------
```

In addition to the **sum** operator, the **min**, **max**, **avgnonzero** and **avg** arithmetic aggregation functions are also supported.

For example, to view the maximum number of events:

```
select max(eventCount) from events
```

To view the number of average events from a source IP:

```
select avg(eventCount) from events group by sourceIP
```

Order by clause examples

You can add a single order by clause to the end of your AQL CLI query. Only one field can be used in the order by clause. Also, sorting can be switched between ascending or descending by appending the asc or desc keyword to the order by clause, respectively.

Examples

To query AQL to return results in descending order.

```
select sourceBytes, sourceIP from flows where sourceBytes >
1000000 order by sourceBytes
```

To display results in ascending order:

```
select sourceBytes, sourceIP from flows where sourceBytes >
1000000 order by sourceBytes asc
```

Combing the group by and the order by clauses in a single query can be used to create data, such as TopN lists, to determine the most abnormal events or the most bandwidth intensive IP addresses. For example, the following query displays the most traffic intensive IP address in descending order:

```
select sourceIP, sum(sourceBytes) from flows group by sourceIP
order by sum(sourceBytes) desc
```

Count(*) clause examples

You can use the count(*) clause to count the number of records matching your query.

Example

To count all events with credibility equal to or greater than 9, type the following query:

```
select count(*) from events where credibility >= 9
```

Distinct clause examples

You can use the distinct clause to select unique rows based on a column or a group of columns. This clause is similar to the group by clause, however, the distinct clause ensures ANSI SQL compatibility.

Example

To create a query to return results with distinct values, type the following query.

```
select distinct sourceIP, sourcePort from flows where
sourceBytes > 1000000
```

Count (distinct ...) clause examples

You can use the standard **`SQL Count(Distinct ...)`** clause to obtain unique counts. Using the AQL CLI, you can only use one field.

Examples

To view all the IP addresses that are connected to a specific IP address over time:

```
select count(distinct sourceIP) from flows where destinationIP =
'192.168.61.71'
```

Or, if you want to view the number of unique source IP addresses communicating with a particular destination IP address:

```
select destinationIP, count(distinct sourceIP) from flows group
by destinationIP
```

Using UniqueCount also returns unique counts for items in the table:

```
select destinationIP, UniqueCount(sourceIP) from flows
```

Note: Using this clause could require additional system resources. Therefore, depending on the query, the amount of time to return results could vary.

Materialize view clauses

The materialize view clause allows you to produce query results as a virtual table and run subsequent queries against the view. You can also specify the period of time that the materialized view is accessible.

Note: You cannot create a materialized view statement based on a previously created materialized view.

The syntax for creating materialized view includes:

```
materialize view NameOfView <time> as select <statement>
```

Where:

- **<time>** specifies the time you want the **materialized view** to be accessible.
- **<statement>** specifies a valid select statement.

Examples

To create a materialized view containing flows with more than 1,000,000 source bytes, type the following command:

```
materialize view LargeSourceBytesFlows as select * from flows
where sourceBytes >1000000
```

To select from this view, enter the select statement as you would a valid table:

```
select * from LargeSourceBytesFlows
```

You can also use an aggregation statement on a materialized view:

```
select sourceIP, sum(sourceBytes) from LargeSourceBytesFlows
group by sourceIP
```

To create a **materialized view** to select from a record set with ambiguous column names, you can define aliases for all computed columns. For example:

```
materialize view MyView as select sourceIP, sum(sourceBytes) as
srcBytesSum from flows group by sourceIP
```

Then you can refer to the alias in a subsequent query against **MyView**:

```
select * from MyView orderBy srcBytesSum
```

Like clause queries

You can search text fields using the standard like clause. You can also use the two wild card options supported by the AQL Query CLI: percentage (%) and underscore (_). The percentage (%) wild card option matches zero or more characters while the underscore (_) wild card option only matches one character.

Examples

To match names such as Joe, Joanne, Joseph, or any other name beginning with Jo, type the following query:

```
select * from events where userName like ‘jo%’
```

To match names beginning with Jo that are three characters long, such as, Joe or Jon, type the following query:

```
select * from events where userName like ‘jo_’
```

You can enter the wild card option at any point in the command. For example:

```
select * from flows where sourcePayload like ‘%xyz’
select * from events where payload like ‘%xyz%’
select * from events where payload like ‘_yz’
```

Boolean clause queries

Boolean clauses allow you to restrict your AQL queries to return data matching values you specify using true or false with relational operators. AQL queries support the following relational operators, equals (=) and does not equal (!=).

Examples

To sort events that are unparsed, type the following query:

```
select * from events where payload = “false”
```

To sort flows to find a specific source IP address that has an offense, type the following query:

```
select * from events where sourceIP = '231.12.37.17' and
hasOffense = “true”
```

To display a list of source IP addresses and protocols that have an offense you can use BooleanTrueCount, and type the following query:

```
select sourceIP, protocol, BooleanTrueCount(hasOffense) from
events group by sourceIP
```

```
--------------------------------
| sourceIP       | protocol     |
--------------------------------
| 64.124.201.151 | TCP.tcp.ip   |
| 10.105.2.10    | UDP.udp.ip   |
| 10.103.70.243  | UDP.udp.ip   |
| 10.103.77.143  | UDP.udp.ip   |
| 10.105.32.29   | TCP.tcp.ip   |
| 10.105.96.148  | TCP.tcp.ip   |
| 64.124.201.151 | TCP.tcp.ip   |
| 10.105.2.10    | ICMP.icmp.ip |
--------------------------------
```

Identifying AQL fields

AQL queries to event or flow fields could return numeric codes making query responses or searches difficult. The shell script idlist.sh provides additional information from numeric AQL fields in the event and flow tables.

You can search the idlist.sh output using standard operators such as fowardslash. Search terms are case sensitive in the idlist.sh output.

Procedure

Step 1 Using SSH, log in to QRadar as the root user.

Username: **root**

Password **<password>**

Step 2 Type the following command:

`/opt/qradar/bin/idlist.sh <-e or -f> <field name>`

Step 3 Enter the appropriate parameters:

Table 1-4 AQL field parameters

Parameters	Description
-f	Specifies that the script is to return information from the flow table. For example, **`/opt/qradar/bin/idlist.sh -f application`** The example query returns a list of application names and the numeric identifier associated with the application from the flows table.
-e	Specifies that the script is to identify a field from the events table. For example: **`/opt/qradar/bin/idlist.sh -e category`** The example query returns a list of low level categories and the numeric identifier associated with the low level category from the events table.

Table 1-4 AQL field parameters (continued)

Parameters	Description
<field name>	Specifies the field name of the event or flow you want to identify. See **Table 1-2** for a complete list of fields that can be identified using idlist.sh.

Querying for event or flow field names

To view a list of the most commonly used field names for a select statement, you can use the describe command.

The results returned by the describe command provides you with a list of field names, a description, and the type of data for the most commonly queried events and flows types. This is helpful for customers to view fields that can be queried without reviewing the documentation.

Procedure

Step 1 Using SSH, log in to QRadar as the root user.

Username: **<root>**

Password: **<password>**

Step 2 Type the following command to start the AQL shell:

```
/opt/qradar/bin/arielClient
```

Step 3 Type one of the following commands to view common event or flow fields:

- `describe flows`
- `describe events`

For a list of fields returned, you can review the following tables:

- Common flow table fields are listed in **Table 1-5**.
- Common event table fields are listed in **Table 1-6**

Table 1-5 Flow table fields listed in describe events

Table	Field Name	Description	Data Type
Flow	application	Application	String
	destinationASN	Destination ASN	Numeric
	destinationBytes	Destination Bytes	Numeric
	destinationDSCP	Destination DSCP	Numeric
	destinationFlags	Destination Flags	Numeric
	destinationIP	Destination IP	String
	destinationIfIndex	Destination If Index	Numeric
	destinationNetwork	Destination Network	String
	destinationPackets	Destination Packets	Numeric
	destinationPayload	Destination Payload	String

Table 1-5 Flow table fields listed in describe events (continued)

Table	Field Name	Description	Data Type
	destinationPort	Destination Port	Numeric
	destinationPrecedence	Destination Precedence	Numeric
	destinationTOS	Destination QOS	Composite
	firstPacketTime	First Packet Time	Numeric
	flowBias	Flow Bias	String
	flowDirection	Flow Direction: • Local-to-Local (L2L) • Local-to-Remote (L2R) • Remote-to-Local (R2L) • Remote-to-Remote (R2R)	String
	flowSource	Flow Source	Numeric
	flowType	Flow Type	Numeric
	geographic	Matches Geographic Location	String
	icmpType	ICMP Type/Code	Numeric
	interface	Flow Interface	String
	intervalId	Interval ID	Numeric
	lastPacketTime	Last Packet Time	Numeric
	protocolId	Protocol ID	Numeric
	sourceASN	Source ASN	Numeric
	sourceBytes	Source Bytes	Numeric
	sourceDSCP	Source DSCP	Numeric
	sourceFlags	Source Flags	Numeric
	sourceIP	Source IP	String
	sourceIfIndex	Source If Index	Numeric
	sourceNetwork	Source Network	String
	sourcePackets	Source Packets	Numeric
	sourcePayload	Source Payload	String
	sourcePort	Source Port	Numeric
	sourcePrecedence	Source Precedence	Numeric
	sourceTOS	Source QOS	Composite
	token	Associated With Offense	Numeric

Table 1-6 Event table fields listed in describe events

Table	Field Name	Description	Data Type
Event	category	Low Level Category	Numeric
	credibility	Credibility	Numeric
	destinationIP	Destination IP	String
	destinationMAC	Destination MAC	Hexadecimal
	destinationNetwork	Destination Network	String
	destinationPort	Destination Port	Numeric
	device	Log Source	Numeric
	deviceGroup	Log Source Group	Numeric
	deviceType	Log Source Type	Numeric
	duration	Duration	Numeric
	endTime	End Time	Numeric
	eventCount	Event Count	Numeric
	eventDirection	Event Direction: • Local-to-Local • Local-to-Remote • Remote-to-Local • Remote-to-Remote	String
	hasIdentity	Has Identity	Boolean
	hasOffense	Associated With Offense	Boolean
	highLevelCategory	High Level Category	Numeric
	identityExtendedField	Identity Extended Field	String
	identityGroupName	Identity Group Name	String
	identityHostName	Identity Host Name	String
	identityIP	Identity IP	String
	identityMAC	Identity MAC	Hexadecimal
	identityNetBiosName	Identity NetBIOS Name	String
	identityUserName	Identity User Name	String
	magnitude	Magnitude	Numeric
	payload	Payload	String
	postNatDestinationIP	Post NAT Destination IP	String
	postNatDestinationPort	Post NAT Destination Port	Numeric
	postNatSourceIP	Post NAT Source IP	String
	postNatSourcePort	Post NAT Source Port	Numeric
	preNatDestinationIP	Pre NAT Destination IP	String
	preNatDestinationPort	Pre NAT Destination Port	Numeric

Table 1-6 Event table fields listed in describe events (continued)

Table	Field Name	Description	Data Type
	preNatSourceIP	Pre NAT Source IP	String
	preNatSourcePort	Pre NAT Source Port	Numeric
	protocol	Protocol	String
	qid	Event Name ID	Numeric
	relevance	Relevance	Numeric
	severity	Severity	Numeric
	sourceIP	Source IP	String
	sourceMAC	Source MAC	Hexadecimal
	sourceNetwork	Source Network	String
	sourcePort	Source Port	Numeric
	startTime	Start Time	Numeric
	token	Token Associated With Offense	Numeric
	unparsed	Event Is Unparsed	Boolean
	userName	Username	String

A NOTICES AND TRADEMARKS

What's in this appendix:

- **Notices**
- **Trademarks**

This section describes some important notices, trademarks, and compliance information.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
170 Tracer Lane,
Waltham MA 02451, USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the

capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at *http:\\www.ibm.com/legal/copytrade.shtml*.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.