ESM API Examples - Find Events Matching a Dynamic Condition

- 1 Objective
- 2 Solution
  - 2.1 Creation of ArcSight Query and Report
  - 2.2 Related ESM API calls

Objective

Find all Events that match a dynamic condition that could be prepared based on results returned by external tools. Here's an example in Logger's syntax

```
start_time=-48h
end_time=now
(deviceReceiptTime IS NOT NULL) AND ( "192.168.0.10" OR "192.168.0.40")
AND (deviceEventClassId = "Microsoft-Windows-Security-Auditing:4624")
AND (destinationNtDomain IS NOT NULL) AND (sourceAddress  IS NOT NULL)
AND (destinationUserName IS NOT NULL) AND (deviceVendor ="Microsoft")
AND (deviceProduct  = "Microsoft Windows") | cef deviceReceiptTime,
deviceEventClassId, destinationNtDomain, sourceAddress,
destinationUserName, deviceVendor, deviceProduct
```

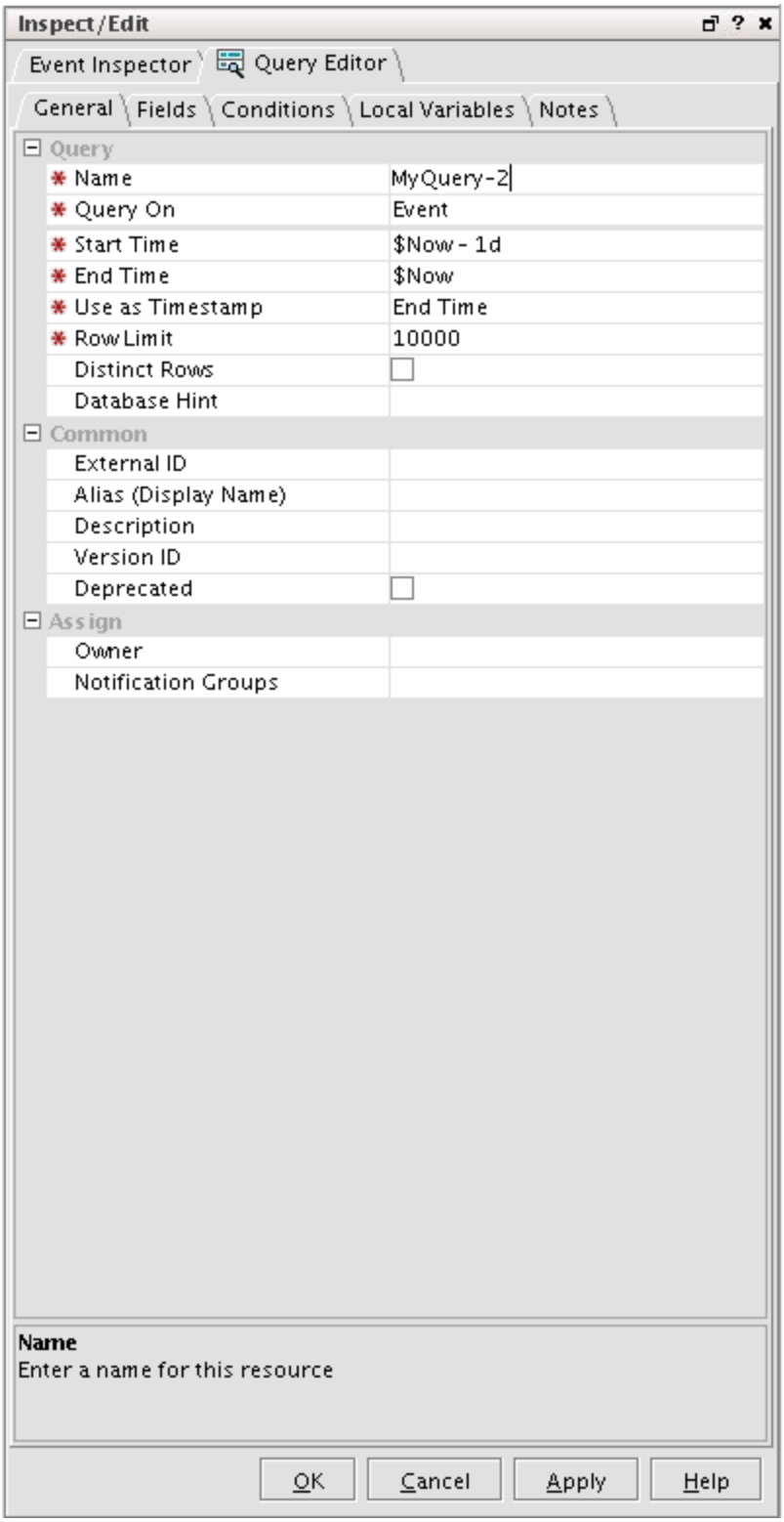
Solution

Create ArcSight Report to report Events that match a condition. Use Query parameters for those values that could be changed dynamically based on results received from external tools. If reports' Query has relative conditions (e.g. $Now – 3h), then the condition is recalculated based on report's execution time.
Then use ESM API's ArchiveReportService to run a report and retrieve the IDs of the Events. Finally use SecurityEventService for Events retrieval.

Note: The solution requires manager-service of the version **1.4.0.release.189** or later.

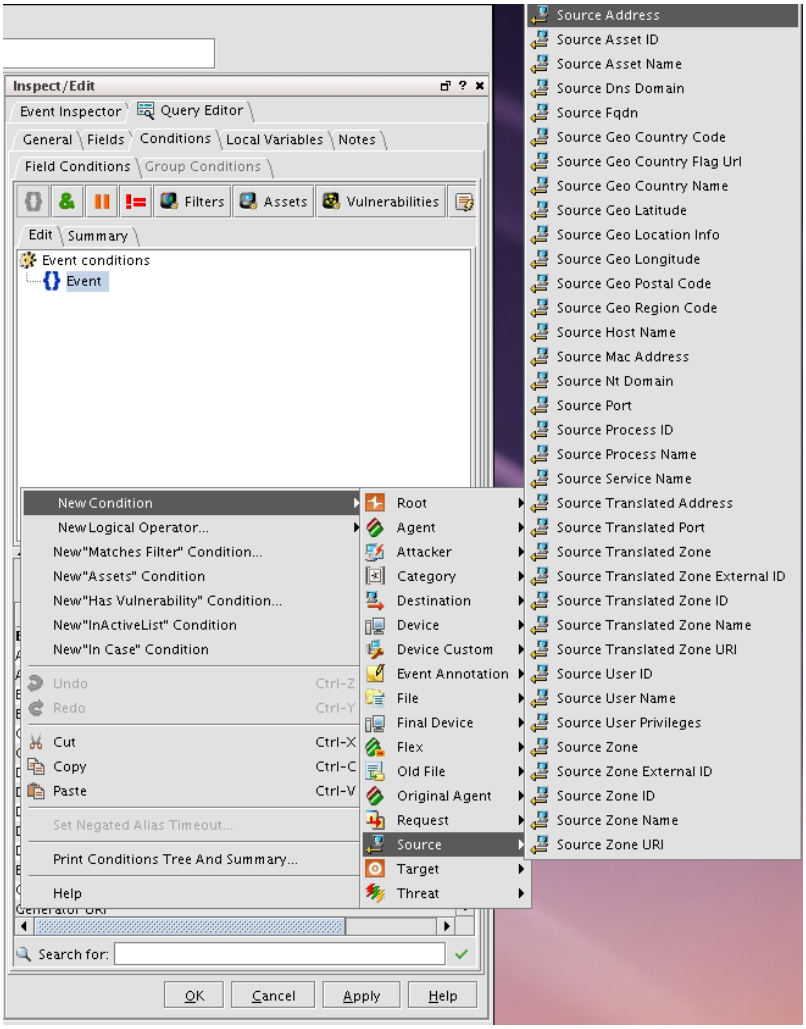
Creation of ArcSight Query and Report

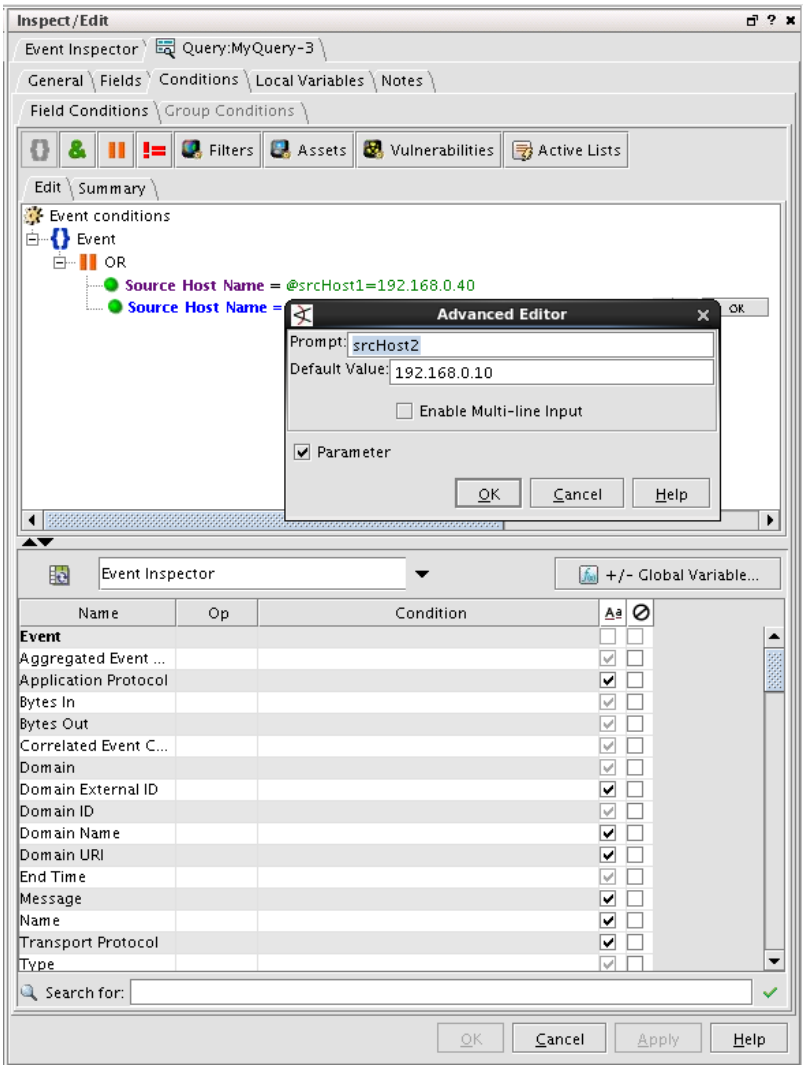
1. Create a Query of Event type (note "**Query On"** field)



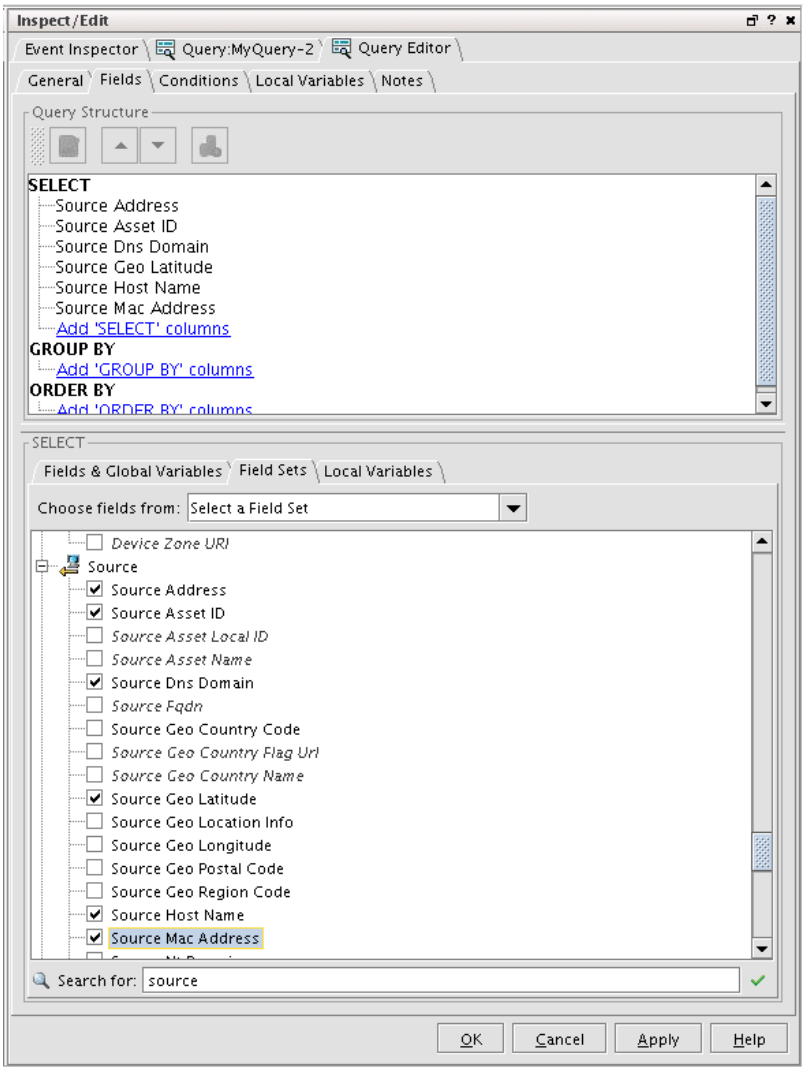
2. Specify a condition(s). Use parameters for those fields that could be changed dynamically later. To match one of several options (e.g. "192.168.0.10" OR "192.168.0.40") add several conditions joined via OR logical operator. The following example demonstrates how to set "Source Host Name" IN ("192.168.0.10" OR "192.168.0.40").

   When specifying a condition you can also use values like ($Now - 3h) that are recalculated for every execution based on execution time. For these values you don't even need to use parameters.

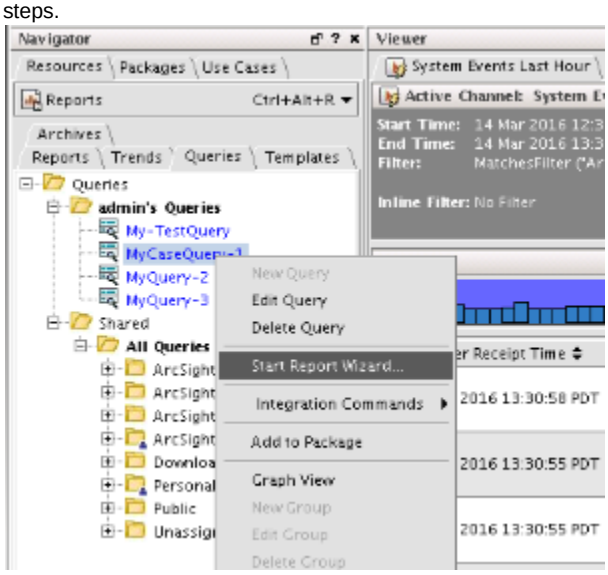

Inspect/Edit
Event Inspector
Query Editor
General
Fields
Conditions
Local Variables
Notes
Field Conditions
Group Conditions
Filters
Assets
Vulnerabilities
Edit
Summary
Event conditions
Event
New Condition
New Logical Operator...
New "Matches Filter" Condition...
New "Assets" Condition
New "Has Vulnerability" Condition...
New "InActiveList" Condition
New "In Case" Condition
Undo Ctrl-Z
Redo Ctrl-Y
Cut Ctrl-X
Copy Ctrl-C
Paste Ctrl-V
Set Negated Alias Timeout...
Print Conditions Tree And Summary...
Help
Root
Agent
Attacker
Category
Destination
Device
Device Custom
Event Annotation
File
Final Device
Flex
Old File
Original Agent
Request
Source
Target
Threat
Source Address
Source Asset ID
Source Asset Name
Source Dns Domain
Source Fqdn
Source Geo Country Code
Source Geo Country Flag Url
Source Geo Country Name
Source Geo Latitude
Source Geo Location Info
Source Geo Longitude
Source Geo Postal Code
Source Geo Region Code
Source Host Name
Source Mac Address
Source Nt Domain
Source Port
Source Process ID
Source Process Name
Source Service Name
Source Translated Address
Source Translated Port
Source Translated Zone
Source Translated Zone External ID
Source Translated Zone ID
Source Translated Zone Name
Source Translated Zone URI
Source User ID
Source User Name
Source User Privileges
Source Zone
Source Zone External ID
Source Zone ID
Source Zone Name
Source Zone URI
Search for:
OK
Cancel
Apply
Help



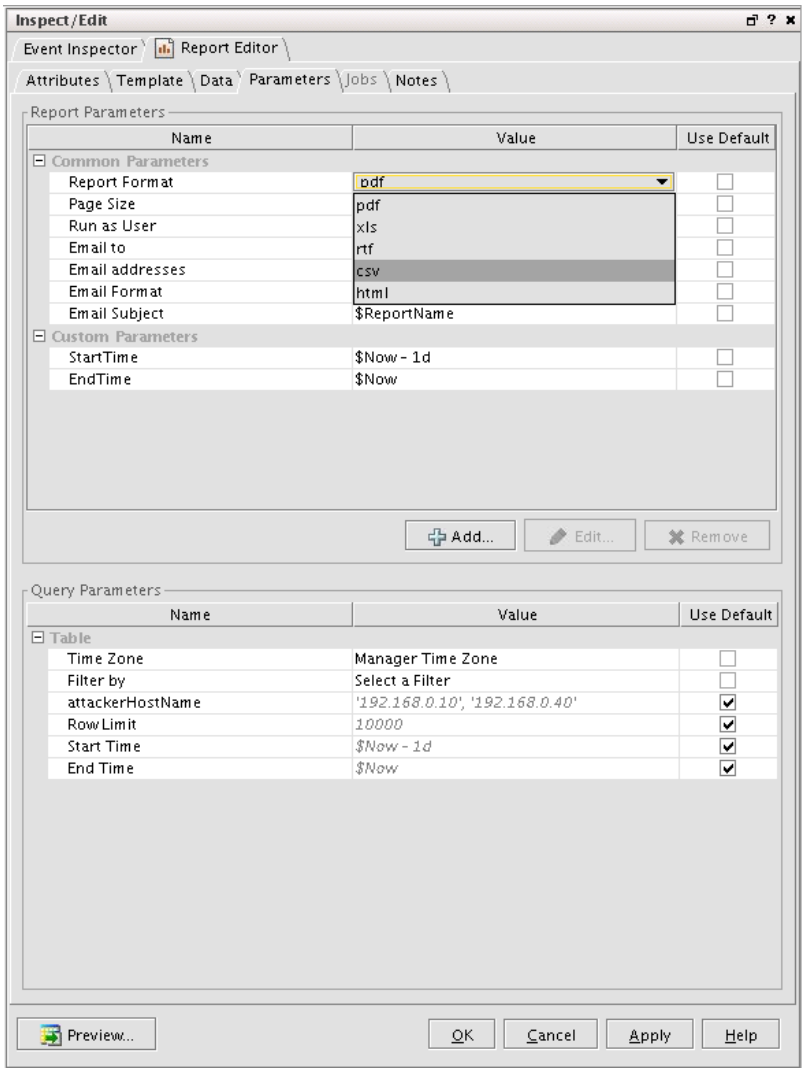
3. Specify Fields that should be returned by report. You probably would only need a Event Id field.



4. Create a Report based on the above Query. That could be easily done using "Report Wizard" that walks you through all configuration steps.



5. When Report is created specify report's format (pdf, xls, rtf, csv, or html). Consider using **CSV**. That's how you will receive the results of Query execution later.



6. Save report.

Related ESM API calls

Use previously created report to get relevant Case IDs later using the following API calls:

1. Login using LoginService

```
https://HOST:8443/www/core-service/rest/LoginService/login?login=admi
n&password=password
```

Add "@alt=json" if you want receive a response in JSON. Otherwise you would get something like

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<ns3:loginResponse>

<ns3:return>7JMM0FGEttlOdfI9kJNheNUBnMLWy1ojM03yO8cfIHQ.</ns3:return>
</ns3:loginResponse>
```

Use the returned token (<authToken> - below) in all other requests.

2. If you don't know Id of your report use ReportService to get Id of all Reports.

3. Use ArchiveReportService to execute report. Every time you call this method, new data that matches Report's conditions will be collected and ESM will prepare ArchivedReport for you. ArchiveReport could be downloaded using instructions provided in the next step.

   Here you have couple options for executing report. When it's enough to run report with the default parameters (or without any parameters) use simpler call "**initDefaultArchiveReportDownloadById**". That could be executed as GET HTTP request:

   ```
   https://HOST:8443/www/manager-service/rest/ArchiveReportService/initD
   efaultArchiveReportDownloadById?reportId=974vFXVMBABCHFyt7PkCdwA%3D%3
   D&reportType=Manual&authToken=<authToken>
   ```

   For the parameters with dynamic values use another method (executed as POST HTTP request only)- **"initDefaultArchiveReportDownloadWithOverwrite**"

   ```
   https://HOST:8443/www/manager-service/rest/ArchiveReportService/initD
   efaultArchiveReportDownloadWithOverwrite
   ```

   with request body containing previously received authentication token and the map of dynamic values, containing values for report parameters (srcHost1 and/or srcHost2)

   ```
   {
     "arc.initDefaultArchiveReportDownloadWithOverwrite": {
       "arc.authToken": "2kFqmzpyvuzgD5yVte7dNcIaehAEL23oI-WdxACxPFs.",
       "arc.reportId": "974vFXVMBABCHFyt7PkCdwA==",
       "arc.reportType": "Manual",
       "arc.fieldValueList": {
         "key": "srcHost1",
         "value": "192.168.0.11"
       },
       {
         "key": "srcHost2",
         "value": "192.168.0.12"
       }
     }
   }
   ```

   By default you would receive response in XML.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:initDefaultArchiveReportDownloadWithOverwriteResponse

xmlns:ns2="http://ws.v1.service.resource.manager.product.arcsight.com
/activeListService/"

xmlns:ns3="http://ws.v1.service.resource.manager.product.arcsight.com
/archiveReportService/"
    ...
xmlns:ns29="http://ws.v1.service.manager.product.arcsight.com/manager
SearchService/"

xmlns:ns30="http://ws.v1.service.manager.product.arcsight.com/infoSer
vice/">

<ns3:return>QZuy_5qEUr6nQhLHRzO0i_VSpTBMqua38R6A66pEPxM.</ns3:return>
</ns3:initDefaultArchiveReportDownloadWithOverwriteResponse>
```

Again, use "&alt=json" or "accept" HTTP header with the value "application/json", if you want to receive the response in JSON.

```
{
  "arc.initDefaultArchiveReportDownloadWithOverwriteResponse": {
    "arc.return": "QZuy_5qEUr6nQhLHRzO0i_VSpTBMqua38R6A66pEPxM."
  }
}
```

Use the returned value on the next step to download the prepared report.

4. Use fileservlet web-service to download the results. Here you would simply read the response of the HTTP request similar to the following

```
https://HOST:8443/www/manager-service/fileservlet?file.command=downlo
ad&file.id=QZuy_5qEUr6nQhLHRzO0i_VSpTBMqua38R6A66pEPxM.
```

For report format CSV and fields as above you would receive a strings like this (3 matching events in this example)

```
Source Address,Source Asset ID,Source Dns Domain,Source Geo
Latitude,Source Host Name,Source Mac
Address,,,,192.168.0.11,,,,,192.168.0.11,,,,,192.168.0.11,
```

5. Parse the output to retrieve Events that match your (dynamic) condition.
6. Do not forget to logout at the end

```
https://HOST:8443/www/core-service/rest/LoginService/logout?authToken
=<AUTH_TOKEN>
```

For successful executions this call doesn't return any response body (response code 204)

7. ...