1 Creating Remediation Scripts for TRITON AP-DATA

Creating Remediation Scripts | Data Protection | Version 8.3.x

This document describes how discovery and data loss prevention (DLP) incidents are created. It also provides examples of how to write and use discovery remediation scripts so you can write your own remediation scripts or remediation programs for discovery or DLP incidents.

Contents:

Discovery

Creating Remediation Scripts | Data Protection | Version 8.3.x

What is discovery?

Discovery is the act of automatically scanning data at rest, classifying it, and potentially enforcing an action on the classification.

Discovery can be performed by 2 components in the Forcepoint™ TRITON® AP-DATA system:

- TRITON AP-DATA crawler, which performs Network discovery.
- TRITON AP-ENDPOINT DLP, which performs Endpoint discovery.

Both components perform essentially the same activity.

Once the documents are classified as having matched a policy rule, an action plan associated with this rule is executed. The action plan specifies what happens next, and this can include running a remediation script.

What is a discovery incident?

An incident is an object that contains information about the file which was matched by a Discovery Policy Rule. It also contains information about the rules it matched, and other meta-data such as the file's permissions, the discovery policy engine name, and so on.

When running remediation scripts, an XML file containing the incident's detail is generated. The full path name is presented to the script as the first command line parameter.

Data Loss Prevention (DLP)

Creating Remediation Scripts | Data Protection | Version 8.3.x

What is DLP?

Data Loss Prevention (DLP) is the activity of classifying real-time data that is communicated on various channels, by various means. Once classified, the data sent over the communication channel might trigger a policy and generate an incident.

Once an incident occurs, an action plan is executed. The action plan may specify which remediation scripts to run.

Classification is performed by the Policy Engine. This means it can be executed on Windows or Linux policy engines, or executed on servers or endpoints (including Linux endpoints, although this functionality is currently not available).

What is a DLP incident?

An incident is an object that contains information about the file which was matched by a DLP Policy Rule. It also contains information about the rules it matched, and other meta-data such as the source and destination, the policy engine name, and so on

When running remediation scripts, an XML file containing the incident details is generated and the full path name is presented to the script as the first command line parameter.

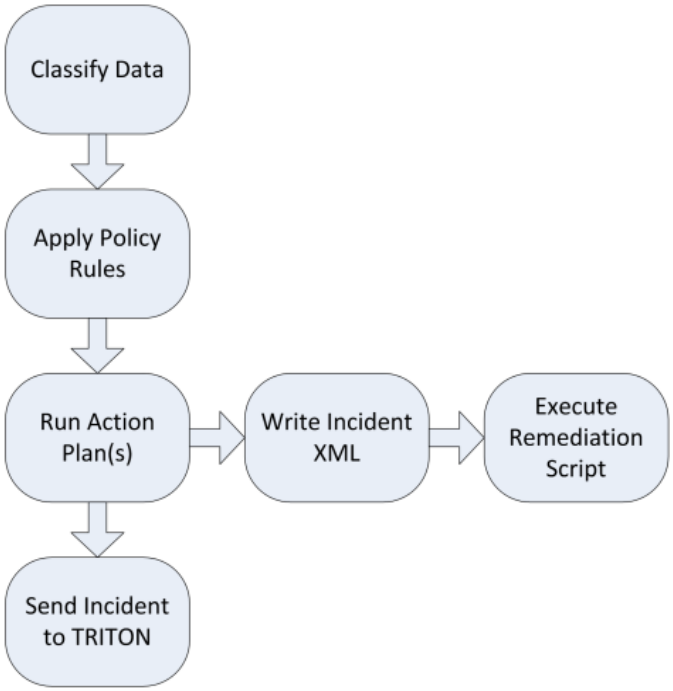
Remediation scripts

Creating Remediation Scripts | Data Protection | Version 8.3.x

What are remediation scripts?

Remediation scripts, in the context of discovery or DLP incidents, lets you extend the functionality of discovery or DLP.

A remediation script is an executable that is run by a Policy Engine or an Endpoint Agent whenever an incident is triggered. Whenever an incident is generated, the system creates an XML file that contains details of the incident. The absolute path to that file is provided as the first command line argument.



How do I configure remediation scripts?

You can configure remediation scripts via the Data Security module of the TRITON Manager. A remediation script is considered a Resource, and is configured under the Resources sections.

For information on how to configure the remediation scripts, refer to the Data Security manager Help section Remediation scripts.

What is the operating system context of a remediation script?

Remediation scripts can be supplied with optional credentials. The table below describes the credentials available to the remediation scripts on multiple platforms:

	Windows Server Policy Engine	**Linux Server Policy Engine**	**Windows Endpoint**	**Linux Endpoint**
With supplied credentials	Impersonate the supplied credentials	*Currently not supported*	Impersonate the supplied credentials	*Currently not supported*
Without supplied credentials	Impersonate the user running the Policy Engine	Effective UID of root	LocalSystem	*Currently not supported*

Please note that DLP remediation scripts do not have access to forensic information (the data that caused the incident).

How are discovery remediation scripts written?

A remediation script can be an executable, or it can be a script written in an interpreted language. Listed below are scripts and language interpreters you can use to write remediation scripts without installing additional software.

- Windows - CMD Shell (.bat, .cmd)
- Windows - VBScript (.vbs) - by running through a CMD shell script that runs the VBScript interpreter cscript.exe explicitly
- Windows & Linux - Python 2.5.4 (.py)
- Windows executable (.exe, .com)
- Linux ELF executable
- Linux BASH scripts

You can also use other interpreted languages, however you will need to install additional software. Ensure the language interpreter is correctly installed on the server. In Network discovery, the language interpreter must be installed on all Crawler machines. In Endpoint discovery, the language interpreter must be installed on the entire endpoint population.

Please note that because DLP and discovery incidents can occur on both Windows and Linux machines. It is highly desirable to write these scripts in Python since Python is available on both Windows and Linux servers and also on all endpoints. In most cases you can use the exact same script for these 2 operating systems without amending the script in such a way to account for changes.

Due to the writer's preference, the samples provided in this document are written in Python.

Discovery details structure

Creating Remediation Scripts | Data Protection | Version 8.3.x

The discovery detail structure is an XML file with no DTD. Following is a sample XML file, taken from File System discovery:

```
<?xml version="1.0" encoding="UTF-8"?>
<ns1:pa-xml-rpc xmlns:ns1="http://www.portauthoritytech.com/
schmea/xml-rpc/1.0" xmlns:evt="http://www.portauthoritytech.com/
schmea/incident/1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <ns1:request>
    <ns1:service-name>insertCrawlerService</ns1:service-name>
    <ns1:params>
      <evt:incident>
        <evt:dataAtRest>
          <evt:incidentInfo>
            <evt:incidentId>5371106770671816417</evt:incidentId>
            <evt:serviceId isSecured="false">1800221564</
evt:serviceId>
            <evt:analyzedBy>NLCTR.nolosscorp.com</evt:analyzedBy>

<evt:subject>\\10.4.228.150\DiscoveryTarget\TestFile.txt</
evt:subject>
            <evt:localDetectedTime>2011-07-18T14:54:11+10:00</
evt:localDetectedTime>
            <evt:installVersion>7.6</evt:installVersion>
            <evt:resourceType>NETWORK</evt:resourceType>
            <evt:totalSize>125</evt:totalSize>
          </evt:incidentInfo>
          <evt:rules>
            <evt:rule id="170998" type="1" policyID="170893">
              <evt:severity>2</evt:severity>
              <evt:actionSettings id="172003"/>
              <evt:numOfMatches>1</evt:numOfMatches>
              <evt:classifierMatches>
                <evt:classifierMatch id="171094">
                  <evt:numberOfMatches>1</evt:numberOfMatches>
                  <evt:isTruncated>false</evt:isTruncated>
                  <evt:breachContent>
                    <evt:contentInfo>
                      <evt:pathPartInfo order="0">

<evt:path>\\10.4.228.150\DiscoveryTarget\TestFile.txt</evt:path>
                        <evt:partType>3</evt:partType>
                        <evt:fileType>2</evt:fileType>
                      </evt:pathPartInfo>
                    </evt:contentInfo>
                    <evt:detectedValues>
                      <evt:detectedValue>
```

```
                          <evt:unMasked>WebsenseTestKeyword</
evt:unMasked>
                        </evt:detectedValue>
                      </evt:detectedValues>
                      <evt:numberOfMatches>1</evt:numberOfMatches>
                    </evt:breachContent>
                  </evt:classifierMatch>
                </evt:classifierMatches>
              </evt:rule>
            </evt:rules>
            <evt:actionTaken type="2097152">
            </evt:actionTaken>
            <evt:properties>
              <evt:property>
                <evt:name>acl</evt:name>

<evt:value>NLC\Administrator:wr,BUILTIN\Administrators:wr,NLC\webs
ense:r,NT AUTHORITY\SYSTEM:wr</evt:value>
              </evt:property>
              <evt:property>
                <evt:name>checksum</evt:name>
                <evt:value>7a0627c2efa25daedb56f19b79c22ab7</
evt:value>
              </evt:property>
              <evt:property>
                <evt:name>fileOwner</evt:name>
                <evt:value>BUILTIN\Administrators</evt:value>
              </evt:property>
              <evt:property>
                <evt:name>folderOwner</evt:name>
                <evt:value>BUILTIN\Administrators</evt:value>
              </evt:property>
              <evt:property>
                <evt:name>jobID</evt:name>
                <evt:value>172104</evt:value>
              </evt:property>
              <evt:property>
                <evt:name>jobName</evt:name>
                <evt:value>RemediationTest</evt:value>
              </evt:property>
              <evt:property>
                <evt:name>resourceSubType</evt:name>
                <evt:value>NETWORK</evt:value>
              </evt:property>
            </evt:properties>
            <evt:file>
              <evt:filepath>cifs://10.4.228.150/DiscoveryTarget/
TestFile.txt</evt:filepath>
              <evt:filesize>39</evt:filesize>
              <evt:filetype>2</evt:filetype>
              <evt:encodeType>N/A</evt:encodeType>
              <evt:ip>10.4.228.150</evt:ip>
              <evt:dateAccessed>2011-07-18T14:51:54</
evt:dateAccessed>
              <evt:dateCreated>2011-07-18T14:51:54</evt:dateCreated>
              <evt:dateModified>2011-07-18T14:52:16</
evt:dateModified>
              <evt:owner>
                <evt:incidentUser>
```

```
                <evt:detail type="5" value="BUILTIN\Administrators"
isLookedUp="false"/>
              </evt:incidentUser>
            </evt:owner>
            <evt:folderOwner>
              <evt:incidentUser>
               <evt:detail type="5" value="BUILTIN\Administrators"
isLookedUp="false"/>
              </evt:incidentUser>
            </evt:folderOwner>
          </evt:file>
          <evt:jobId>172104</evt:jobId>
          <evt:jobName></evt:jobName>
          <evt:scanStartTime>2011-07-18T14:54:06</
evt:scanStartTime>
          <evt:discoveryEndpointInfo>
            <evt:endpointType>Unknown</evt:endpointType>
          </evt:discoveryEndpointInfo>
        </evt:dataAtRest>
      </evt:incident>
    </ns1:params>
  </ns1:request>
</ns1:pa-xml-rpc>
```

Some interesting nodes in this file:

`/ns1:pa-xml-rpc/ns1:request/ns1:params/ evt:incident/ evt:dataAtRest/evt:incidentInfo/ evt:resourceType`	Contains NETWORK or ENDPOINT – depending on the type of discovery incident. Can enable the creation of scripts that work on both types.
`/ns1:pa-xml-rpc/ns1:request/ns1:params/ evt:incident/ evt:dataAtRest/evt:rules/`	A list of the violated rules.
`…evt:rules/evt:rule/evt:classifierMatches /ns1:pa-xml-rpc/ns1:request/ns1:params/ evt:incident/`	A list of the matched classifier in a specific rule.
`evt:dataAtRest/evt:properties/ /ns1:pa-xml-rpc/ns1:request/ns1:params/ evt:incident/`	A list of properties such as file owner, file ACL, discovery task name.
`evt:dataAtRest/evt:file/evt:filepath /ns1:pa-xml-rpc/ns1:request/ns1:params/ evt:incident/`	The path to the file, in a URI format. Converting it back to a UNC format means that the slashes needs to be turned around into backslashes.
`evt:dataAtRest/evt:file/evt:dateModifies /ns1:pa-xml-rpc/ns1:request/ns1:params/ evt:incident/`	The date the file was last modified in YYYY-MM-DDTHH:MM:SS.
`evt:dataAtRest/evt:file/evt:owner/evt: incidentUser/ evt:detail`	In the network it contains an evt:detail node with a value attribute of DOMAIN\Username (type 5). On the endpoint it contains a value attribute of the user's SID (type 8).

Here is a sample of Exchange discovery:

```
<?xml version="1.0" encoding="UTF-8"?>
<ns1:pa-xml-rpc xmlns:ns1="http://www.portauthoritytech.com/
schmea/xml-rpc/1.0" xmlns:evt="http://www.portauthoritytech.com/
schmea/incident/1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <ns1:request>
    <ns1:service-name>insertCrawlerService</ns1:service-name>
    <ns1:params>
      <evt:incident>
        <evt:dataAtRest>
          <evt:incidentInfo>
            <evt:incidentId>4679778800686204169</evt:incidentId>
            <evt:serviceId isSecured="false">1800221564</
evt:serviceId>
            <evt:analyzedBy>NLCTR.nolosscorp.com</evt:analyzedBy>
            <evt:subject>ismith/Deleted Items/DSS Incident
[ID:12564].EML</evt:subject>
            <evt:localDetectedTime>2011-07-26T14:17:57+10:00</
evt:localDetectedTime>
            <evt:installVersion>7.6</evt:installVersion>
            <evt:resourceType>EXCHANGE</evt:resourceType>
            <evt:totalSize>36827</evt:totalSize>
          </evt:incidentInfo>
          <evt:rules>
            <evt:rule id="170998" type="1" policyID="170893">
              <evt:severity>2</evt:severity>
              <evt:actionSettings id="172003"/>
              <evt:numOfMatches>1</evt:numOfMatches>
              <evt:classifierMatches>
                <evt:classifierMatch id="171094">
                  <evt:numberOfMatches>1</evt:numberOfMatches>
                  <evt:isTruncated>false</evt:isTruncated>
                  <evt:breachContent>
                    <evt:contentInfo>
                      <evt:pathPartInfo order="0">
                       <evt:path>ismith/Deleted Items/DSS Incident
[ID:12564].EML</evt:path>
                        <evt:partType>1</evt:partType>
                        <evt:fileType>233</evt:fileType>
                      </evt:pathPartInfo>
                      <evt:pathPartInfo order="1">
                        <evt:path>Transaction Body.txt</evt:path>
                        <evt:partType>1</evt:partType>
                        <evt:fileType>236</evt:fileType>
                      </evt:pathPartInfo>
                    </evt:contentInfo>
                    <evt:detectedValues>
                      <evt:detectedValue>
                        <evt:unMasked>WebsenseTestKeyword</
evt:unMasked>
                      </evt:detectedValue>
                    </evt:detectedValues>
                    <evt:numberOfMatches>1</evt:numberOfMatches>
                  </evt:breachContent>
                  <evt:breachContent>
                    <evt:contentInfo>
                      <evt:pathPartInfo order="0">
```

```
                        <evt:path>ismith/Deleted Items/DSS Incident
[ID:12564].EML</evt:path>
                         <evt:partType>1</evt:partType>
                         <evt:fileType>233</evt:fileType>
                       </evt:pathPartInfo>
                       <evt:pathPartInfo order="1">
                         <evt:path>Original_Message_Incident_12564</
evt:path>
                         <evt:partType>2</evt:partType>
                         <evt:fileType>233</evt:fileType>
                       </evt:pathPartInfo>
                       <evt:pathPartInfo order="2">
                         <evt:path>Transaction Body.txt</evt:path>
                         <evt:partType>2</evt:partType>
                         <evt:fileType>2</evt:fileType>
                       </evt:pathPartInfo>
                     </evt:contentInfo>
                     <evt:detectedValues>
                       <evt:detectedValue>
                         <evt:unMasked>WebsenseTestKeyword</
evt:unMasked>
                       </evt:detectedValue>
                     </evt:detectedValues>
                     <evt:numberOfMatches>1</evt:numberOfMatches>
                   </evt:breachContent>
                 </evt:classifierMatch>
               </evt:classifierMatches>
             </evt:rule>
           </evt:rules>
           <evt:actionTaken type="2097152">
           </evt:actionTaken>
           <evt:properties>
             <evt:property>
               <evt:name>checksum</evt:name>
               <evt:value>60104d41558c2d6aba1ad287813155ea</
evt:value>
             </evt:property>
             <evt:property>
               <evt:name>exchange-from</evt:name>
               <evt:value>&amp;quot;DSS@nolosscorp.com&amp;quot;
&amp;lt;DSS@nolosscorp.com&amp;gt;</evt:value>
             </evt:property>
             <evt:property>
               <evt:name>exchange-subject</evt:name>
               <evt:value>DSS Incident [ID:12564]</evt:value>
             </evt:property>
             <evt:property>
               <evt:name>exchange-to</evt:name>
               <evt:value>&amp;quot;ismith@nolosscorp.com&amp;quot;
&amp;lt;ismith@nolosscorp.com&amp;gt;</evt:value>
             </evt:property>
             <evt:property>
               <evt:name>fileOwner</evt:name>
               <evt:value>ismith</evt:value>
             </evt:property>
             <evt:property>
               <evt:name>folderOwner</evt:name>
               <evt:value>N/A</evt:value>
             </evt:property>
```

```
            <evt:property>
              <evt:name>jobID</evt:name>
              <evt:value>172106</evt:value>
            </evt:property>
            <evt:property>
              <evt:name>jobName</evt:name>
              <evt:value>Test discovery</evt:value>
            </evt:property>
            <evt:property>
              <evt:name>resourceSubType</evt:name>
              <evt:value>PRIVATE FOLDER</evt:value>
            </evt:property>
          </evt:properties>
          <evt:file>
            <evt:filepath>cifs://ismith/Deleted Items/DSS Incident
[ID:12564].EML</evt:filepath>
            <evt:filesize>19672</evt:filesize>
            <evt:filetype>233</evt:filetype>
            <evt:encodeType>N/A</evt:encodeType>
            <evt:hostname>ismith@nolosscorp.com</evt:hostname>
            <evt:dateAccessed>2010-10-21T03:10:51.505</
evt:dateAccessed>
            <evt:dateCreated>2010-10-21T03:10:51.505</
evt:dateCreated>
            <evt:dateModified>2010-10-21T03:10:51.505</
evt:dateModified>
            <evt:owner>
              <evt:incidentUser>
                <evt:detail type="5" value="ismith"
isLookedUp="false"/>
              </evt:incidentUser>
            </evt:owner>
            <evt:folderOwner>
              <evt:incidentUser>
               <evt:detail type="5" value="N/A" isLookedUp="false"/
>
              </evt:incidentUser>
            </evt:folderOwner>
          </evt:file>
          <evt:jobId>172106</evt:jobId>
          <evt:jobName></evt:jobName>
          <evt:scanStartTime>2011-07-26T14:16:49</
evt:scanStartTime>
          <evt:discoveryEndpointInfo>
            <evt:endpointType>Unknown</evt:endpointType>
          </evt:discoveryEndpointInfo>
        </evt:dataAtRest>
      </evt:incident>
    </ns1:params>
  </ns1:request>
</ns1:pa-xml-rpc>
```

Please note the main differences:

- The `evt:parameters` sections contain more Exchange centric information (such as email fields).
- The pathname in the `evt:file` section is invalid as a path name, but is valid as a URL suffix in OWA instead.

- The `evt:resourceType` is "`EXCHANGE`", which differentiates this incident.

You will have to write your own parsing code to get the information in this incident. The provided module will not be able to extract any meaningful information from it.

DLP incident structure

Creating Remediation Scripts | Data Protection | Version 8.3.x

```
<?xml version="1.0" encoding="UTF-8"?>
<ns1:pa-xml-rpc xmlns:ns1="http://www.portauthoritytech.com/
schmea/xml-rpc/1.0" xmlns:evt="http://www.portauthoritytech.com/
schmea/incident/1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <ns1:request>
    <ns1:service-name>insertEventService</ns1:service-name>
    <ns1:params>
      <evt:incident>
        <evt:dataInMotion>
          <evt:incidentInfo>
            <evt:incidentId>5352285115603247792</evt:incidentId>
            <evt:serviceId isSecured="false">486169846</
evt:serviceId>
            <evt:analyzedBy>nlcv10k-c-esg.nolosscorp.com</
evt:analyzedBy>
            <evt:subject>test inbound 3</evt:subject>
            <evt:localDetectedTime>2011-07-21T12:33:35+10:00</
evt:localDetectedTime>
            <evt:installVersion>7.6</evt:installVersion>
            <evt:resourceType>NETWORK</evt:resourceType>
            <evt:totalSize>1740</evt:totalSize>
          </evt:incidentInfo>
          <evt:rules>
            <evt:rule id="171601" type="1" policyID="170899">
              <evt:severity>2</evt:severity>
              <evt:actionSettings id="172004"/>
              <evt:numOfMatches>1</evt:numOfMatches>
              <evt:classifierMatches>
                <evt:classifierMatch id="171094">
                  <evt:numberOfMatches>1</evt:numberOfMatches>
                  <evt:isTruncated>false</evt:isTruncated>
                  <evt:breachContent>
                    <evt:contentInfo>
                      <evt:pathPartInfo order="0">
                        <evt:path>/var/spool/postfix/tmp//
887C7850695.eml</evt:path>
                        <evt:partType>1</evt:partType>
                        <evt:fileType>233</evt:fileType>
                      </evt:pathPartInfo>
                      <evt:pathPartInfo order="1">
                        <evt:path>Transaction Body.txt</evt:path>
                        <evt:partType>1</evt:partType>
```

```
                        <evt:fileType>2</evt:fileType>
                      </evt:pathPartInfo>
                    </evt:contentInfo>
                    <evt:detectedValues>
                      <evt:detectedValue>
                        <evt:unMasked>WebsenseTestKeyword</
evt:unMasked>
                      </evt:detectedValue>
                    </evt:detectedValues>
                    <evt:numberOfMatches>1</evt:numberOfMatches>
                  </evt:breachContent>
                </evt:classifierMatch>
              </evt:classifierMatches>
            </evt:rule>
          </evt:rules>
          <evt:actionTaken type="2097152">
          </evt:actionTaken>
          <evt:source>
            <evt:incidentUser>
              <evt:detail type="2" value="test@arik.baratz.org"
isLookedUp="false"/>
            </evt:incidentUser>
          </evt:source>
          <evt:destinations>
            <evt:destination>
              <evt:incidentUser>
                <evt:detail type="2"
value="administrator@nolosscorp.com" isLookedUp="false"/>
              </evt:incidentUser>
              <evt:destinationType>TO</evt:destinationType>
              <evt:actionTaken type="2097152">
              </evt:actionTaken>
              <evt:direction>1</evt:direction>
            </evt:destination>
            <evt:destination>
              <evt:incidentUser>
                <evt:detail type="2" value="ragg@nolosscorp.com"
isLookedUp="false"/>
              </evt:incidentUser>
              <evt:destinationType>TO</evt:destinationType>
              <evt:actionTaken type="2097152">
              </evt:actionTaken>
              <evt:direction>1</evt:direction>
            </evt:destination>
            <evt:destination>
              <evt:incidentUser>
                <evt:detail type="2" value="ismith@nolosscorp.com"
isLookedUp="false"/>
              </evt:incidentUser>
              <evt:destinationType>TO</evt:destinationType>
              <evt:actionTaken type="2097152">
              </evt:actionTaken>
              <evt:direction>1</evt:direction>
            </evt:destination>
```

```
          </evt:destinations>
          <evt:eventEndpointInfo>
            <evt:endpointType>Unknown</evt:endpointType>
            <evt:endpointSourceAppName>N/A</
evt:endpointSourceAppName>
            <evt:endpointDestAppName>N/A</evt:endpointDestAppName>
            <evt:endpointDestDeviceName>N/A</
evt:endpointDestDeviceName>
            <evt:endpointDestDeviceType>N/A</
evt:endpointDestDeviceType>
            <evt:endpointOperationType>N/A</
evt:endpointOperationType>
            <evt:endpointPolicyVersion>0</
evt:endpointPolicyVersion>
            <evt:confirmationId>0</evt:confirmationId>
            <evt:confirmationString></evt:confirmationString>
            <evt:endpointSourceAppID>N/A</evt:endpointSourceAppID>
            <evt:endpointDestAppID>N/A</evt:endpointDestAppID>
          </evt:eventEndpointInfo>
          <evt:hasForensics>true</evt:hasForensics>
        </evt:dataInMotion>
      </evt:incident>
    </ns1:params>
  </ns1:request>
</ns1:pa-xml-rpc>
```

Some interesting nodes in this file:

`/ns1:pa-xml-rpc/ns1:request/ns1:params/evt:incident/evt:dataInMotion/evt:incidentInfo/evt:resourceType`	Contains NETWORK or ENDPOINT – depending on the type of discovery incident. Can enable the creation of scripts that work on both types.
`/ns1:pa-xml-rpc/ns1:request/ns1:params/evt:incident/evt:dataInMotion/evt:incidentInfo/evt:subject`	Channel dependent, in case of email would be the email subject, in other channels will contain other types of information (URL for HTTP for example).
`/ns1:pa-xml-rpc/ns1:request/ns1:params/evt:incident/evt:dataInMotion/evt:rules/`	A list of the violated rules.

…evt:rules/ evt:rule/ evt:classifierMatches /ns1:pa-xml-rpc/ ns1:request/ ns1:params/ evt:incident/	A list of the matched classifier in a specific rule.
evt: dataInMotion / evt:source/ evt:incidentUser/ evt:detail /ns1:pa-xml-rpc/ ns1:request/ ns1:params/ evt:incident	The source of the incident in the “value” attribute. Resource type depends on what the source is – 2 is email.
/ evt:DataInMotion/ evt:destinations/ evt:destination/ evt:incidentUser/ evt:detail	The destination(s) of the incident. In this example these are email addresses.

Existing discovery incident data functionality

Creating Remediation Scripts | Data Protection | Version 8.3.x

To make it easier to write some code for common use cases, Forcepoint provides a helper Python module that performs some common tasks with the incident data XML file. This section describes the module in detail.

The module name is DiscoveryIncidentProcessing, and this module can be easily imported into your Python code.

It is not mandatory to use DiscoveryIncidentProcessing; it is perfectly valid to write your own XML parsing routines.

Note

Due to a current limitation with the XML parser, the module is not supported on Windows Endpoints that run on a 64-bit operating system.

Below are some provided routines:

Def: **GetFilePathFromXML(IncidentXml)**

Description: Reads and analyses the incident details from the provided XML file.

Params:	`IncidentXml`	Unicode	The path to the XML file containing the incident details.
Returns: tuple	0	Unicode	"NETWORK" or "ENDPOINT", depends on the discovery location.
	1	Unicode	The absolute file path. On the endpoint it will have the UNC form \\hostname\C\path\to\file.
	2	Unicode	True.

Example:

```
>>> import DiscoveryIncidentProcessing
>>>
DiscoveryIncidentProcessing.GetFilePathFromXML(r'C:\Temp\537110
6770671816417.xml')
(u'NETWORK',
u'\\\\10.4.228.150\\DiscoveryTarget\\TestFile.txt', True)
```

```
>>>
```

Def: **RunCommand(Command, FilePath)**

Description: Runs a command, ignores the output and logs any errors.

Params:	`Command`	Unicode	A command to execute. The string should contain the string "$filepath$" which would be replaced with the FilePath parameter.
	`FilePath`	Unicode	A path given as a parameter to the command. If no path is given, it must be an empty string.
Returns:	`None`		

Example:

```
>>> import DiscoveryIncidentProcessing
>>> DiscoveryIncidentProcessing.RunCommand(u'calc.exe',u'')
2011-07-19 18:14:59,888 root  Debug  Command:calc.exe
2011-07-19 18:15:02,663 root  Debug  RunCommand Successful
>>>
```

Def: **ProcessDicoveryIncident(IncidentXml, Command)**

Description: Runs a command, providing the incident file name as a parameter. This is quite useful to run commands that expect the original file as one of its parameters.

NOTE: The typo in the function name will be fixed in future versions.

Params:	`IncidentXML`	Unicode	The path of the incident XML file.
	`Command`	Unicode	A command to execute. The string should contain the string "$filepath$" which would be replaced with the actual filename in the incident XML.
Returns:	`None`		

Example:

```
>>>
DiscoveryIncidentProcessing.ProcessDicoveryIncident(r'C:\Temp\5
371106770671816417.xml',
u'notepad.exe   filepath  ')
2011-07-19 18:32:45,312 root   Debug    Processing
C:\Temp\5371106770671816417.xml Encryption
2011-07-19 18:32:45,496 root   Debug    Processing
\\10.4.228.150\DiscoveryTarget\TestFile.txt
2011-07-19 18:32:45,500 root   Debug    Command:notepad.exe
\\10.4.228.150\DiscoveryTarget\TestFile.txt
```

```
2011-07-19 18:32:50,898 root   Debug
\\10.4.228.150\DiscoveryTarget\TestFile.txt RunCommand
Successful
>>>
```

Def: **MoveDiscoveryIncident (IncidentXml, Location, RemoveFile, DaysKeepActiveFiles, QuarentineMsg)**

Description: Move the file pointed to by the incident into a folder. The file will be moved by copying it to the destination folder and then overwriting the original with a text message. Alternatively the file can be copied. The file is checked for access before it is copied / moved and will not be moved if it is accessed recently.

Params:	`IncidentXML`	Unicode	The path of the incident XML file.
	`Location`	Unicode	A command to execute. The string should contain the string "$filepath$" which would be replaced with the actual filename in the incident XML.
	`RemoveFile`	bool	If True, the original file is moved. If False – it is only copied.
	`DaysKeepActiveFiles`	Int	Don't move the file if it was accessed within this number of days.
	`QuarantineMsg`	str	A string which will replace the original file. Should be formatted according to the way the file will look. The file will always have a .txt extension so make sure it can be opened in Notepad – for example if you want the message to be in Unicode you must encode it as UTF-8 or UTF-16 with BOM.
Returns:	`None`		

Example:

```
>>>
DiscoveryIncidentProcessing.MoveDiscoveryIncident(r'C:\Temp\537
1106770671816417.xml',r'C:\Temp',False,0,'')
2011-07-21 16:03:16,365 root      Debug    Processing
C:\Temp\5371106770671816417.xml move file 0
2011-07-21 16:03:16,742 root      Debug    Moving
\\10.4.228.150\DiscoveryTarget\TestFile.txt to C:\Temp
2011-07-21 16:03:16,786 root      Debug    Creating
C:\Temp\10.4.228.150\DiscoveryTarget
>>>
```

Code samples

This section provides some code samples; some samples use DiscoveryIncidentProcessing, while some don't.

```
#Set the destination folder for sensitive files
Location = r'\\127.0.0.1\quarantine'
DaysKeepActiveFiles = 0
#-----------------------------------------------------------------
#DO NOT MODIFYSCRIPT PAST THIS LINE
import sys
from DiscoveryIncidentProcessing import MoveDiscoveryIncident

MoveDiscoveryIncident(sys.argv[1],Location,False,DaysKeepActiveF
iles,'')
```

This example is one of the 3 scripts that come with TRITON AP-DATA and reside in the %dss_home%/RunCommands folder. It is an example of using the `DiscoveryIncidentProcessing` module and is self-explanatory.

The following example is a little more involved and does the XML parsing without using the helper module - the module is useful only for discovery incidents:

```
"Send an email to the recipients of a Data in Motion incident"

# some XML search path constants for easier XML handling
NS1=u".//{http://www.portauthoritytech.com/schmea/xml-rpc/1.0}"
EVT=u".//{http://www.portauthoritytech.com/schmea/incident/
1.0}"
EVTSOURCE=EVT+u"source"
EVTDETAIL=EVT+u"detail"
EVTDESTINATIONS=EVT+u"destinations"
EVTDESTINATION=EVT+u"destination"
EVTSUBJECT=EVT+u"subject"

## the email message that would be sent to the user(s)
EMAILMESSAGE="""From: Your friendly TRITON AP-DATA
<dss@example.com>\r
To: %(deststring)s\r
Subject: Re: %(subject)s\r
\r
Dear Sir or Madam,\r
\r
A message sent to you from %(source)s with the subject
"%(subject)s" has violated the PCI regulation - and therefore
has been blocked by TRITON AP-DATA. Please contact the sender
and request that they redact all cardholder information (such as
name, credit card number, expiration date, CVV) from their
message and resend it.\r
\r
Regards,\r
\r
```

```
23. Your friendly TRITON AP-DATA\r
24. """
25.
26. # email gateway
27. SMTPGATEWAY='10.4.228.240:25'
28.
29. import sys
30. import xml.etree.ElementTree as ET
31. import smtplib
32.
33. # parse the XML file
34. oXMLTree=ET.parse(sys.argv[1])
35.
36. ## search for a few key pieces of data
37. dIncidentDetails={}
38.
39. # source
40. dIncidentDetails['source']=oXMLTree.find(EVTSOURCE).find(EVTDET
    AIL).get('value')
41.
42. # destinations
43. lDests=[
44.       elem.find(EVTDETAIL).get('value')
45.       for elem
46.       in oXMLTree.find(EVTDESTINATIONS)
47.    ]
48. dIncidentDetails['deststring']=', '.join(lDests)
49.
50. # extract the subject
51. dIncidentDetails['subject']=oXMLTree.find(EVTSUBJECT).text
52.
53. ## send an email message
54.
55. oSMTP=smtplib.SMTP(SMTPGATEWAY)
56. oSMTP.sendmail(dIncidentDetails['source'],lDests,EMAILMESSAGE %
    dIncidentDetails)
57. oSMTP=None
```

Please note several important aspects of this example:

- The search paths constructed on lines 4-10 do not contain the xmlns shortcuts for the namespaces but rather the full URL. This is a requirement of the particular XML library used (xml.etree) and is also more resilient to future changes in the XML format.
- Lines 13 to 24 contain an IMF message (RFC5322) ready to be sent by SMTP. Note the Pythonic variable substitutions inside the script makes for an easy ad-hock template language.
- Lines 30 and 34 are all that is required to parse the XML file. Then, extracting information from the XML file is done in a series of searches on lines 40, 44, 46 and 51. It doesn't get easier than this.

- Lines 43-47 are a Python construct called "list comprehension" - a powerful and useful Python constructs for transforming lists. Read up on it in http://www.python.org/dev/peps/pep-0202/ if you're not familiar with it.
- Lines 55-57 sends an SMTP message. Please note that the gateway must accept SMTP connections for this to work.
- There is a bug in this code - if an incoming mail has destinations outside of the organization as well as inside, then all of the destinations will receive the notification message. This is usually not the desired action. Fixing it is easy and left as an exercise to the reader.

The following example is short but very useful:

```
1.  "copy the context xml file into a backup folder"
2.  import sys,os
3.
4.  fileName=sys.argv[1]
5.
6.  # check which platform we're on
7.  if sys.platform[:5]=='linux':
8.    tempFolder='/tmp'
9.  elif sys.platform[:3]=='win':
10.   tempFolder=r'c:\temp' # assumes the folder is there!!!
11. else:
12.    # different platform?
13.    sys.exit(1)
14.
15. newName=os.path.join(tempFolder,os.path.split(fileName)[1])
16.
17. # copy!
18. newFile=open(newName,'wb')
19. oldFile=open(fileName,'rb')
20. newFile.write(oldFile.read())
21. newFile.close()
```

This short piece of code simply copies the provided XML file to a temporary location - usually for debugging or further examination.

Note

Lines 7-13 demonstrate how to write a remediation script that is valid on both Windows and Linux. It uses the `sys.platform` facility to determine the file system structure. Note that the complete string in `sys.platform` may be `linux2`, `win32`, etc - but the script only looks at the first few characters to make a determination.

Additional note: If you want to write multi-platform code in Python, it's best to avoid making assumptions about the environment if possible. Python helps you quite a bit in that regard by providing facilities that performs certain actions

regardless of the platform you are on. A very common example is concatenating filenames and folders.

The following code is not cross-platform because it will fail on a Linux machine.

```
if not folderName[-1:]=='\\':
   folderName+=r'\\'
pathName=folderName+fileName
```

Python provides a comprehensive library of path handling routines - os.path - that works regardless of the operating system the script runs on:

```
pathName = os.path.join(folderName,fileName)
```

The same applies to other operations such as splitting a path. For example, the following script:

```
pathParts = pathName.rsplit('/',1)
if len(pathParts)==2:
   folderName,fileName=pathParts
```

would work just as well using:

```
folderName,fileName = os.path.split(pathName)
```

These scripts would work on both operating systems, and also cover corner cases you may not have considered (for example the input is just a folder and not a filename).