



**Configuring the RSA DLP Connector for the
Core Access Assurance Suite Sensitive Data Manager**

Version 9.1

Core Security

1000 Holcomb Woods Parkway

Suite 401

Roswell, GA 30076

Domestic : (678) 304-4500

Fax: (770) 573-3743

Trademarks

Copyright © 2018 by Core Security SDI Corporation. All Rights Reserved. The following are trademarks of Core Security Corporation "Core Impact", "Core Vulnerability Insight", "Core Password", "Core Access", "Core Provisioning", "Core Compliance", "Core Access Insight", "Core Mobile Reset", and "Think Like an Attacker". The following are registered trademarks of Core Security Corporation "WebVerify", "CloudInspect", "Core Insight", and "Core Security". The names of actual companies and products mentioned herein may be the trademarks of their respective owners. The names of additional products may be trademarks or registered trademarks of their respective owners.

Table of Contents

1	OVERVIEW	7
2	PRE-REQUISITES	7
3	ARCHITECTURAL VIEW.....	7
4	BUILDING THE DLP/METRIC ANALYSIS DATABASE.....	8
5	RUNNING THE DLP INTEGRATION TOOL.....	10
5.1	DLP Integration Tool Background.....	10
5.1.1	Configuration XML File– (Sample_RSA_DLP.xml).....	11
5.2	Using the analysis tool to prepare data for the certification review cycle type and metrics	15
5.3	Running the DLP Integration Tool in Encrypt Mode.....	17
6	THE RSA DLP SAMPLE METRICS	17
6.1	Configure the RSA DLP Metric Data Source	18
6.2	At Rest Sensitive Data Violation Trend.....	19
6.2.1	Display Argument Customizations	20
6.2.2	Related Metrics (Drill-down)	21
6.3	At Rest Violation Count by Department.....	21
6.3.1	Display Argument Customizations	22
6.3.2	Related Metrics (Drill-down)	22
6.4	At Rest Violation Count by Role	22
6.4.1	Display Argument Customizations	24
6.4.2	Related Metrics (Drill-down)	24
6.5	At Rest Violation Count by Role and Regulation	24
6.5.1	Display Argument Customizations	25
6.5.2	Related Metrics (Drill-down)	26
6.6	At Rest Unmitigated Sensitive Data Violation Trend.....	26
6.6.1	Display Argument Customizations	27
6.6.2	Related Metrics (Drill-down)	27
6.7	At Rest Unmitigated Sensitive Data Violation by Manager and Regulation.....	28
6.7.1	Display Argument Customizations	29
6.7.2	Related Metrics (Drill-down)	29
6.8	In Motion Sensitive Data Violations by Role	28
6.8.1	Display Argument Customizations	30

1 Overview

This document provides a complete end-to-end guide for configuring the Core Access Assurance Suite Sensitive Data Manager RSA DLP Connector. It provides information on the DLP Integration Tool, RSA DLP Integration Sample workflow and the RSA DLP Metrics.

The document is broken into three sections discussing the manual steps to get each part of the integration process working and describing the relevant schema additions and configuration steps. The actual steps to manually perform are prefaced with a “**PERFORM STEPS:**” header.

1 Pre-requisites

The following pre-requisites must be in place to use any of the RSA DLP integration package:

- Core Access Assurance Suite v8.0 or greater installed
- Access Keys for Sensitive Data
- Up-to-date Transaction Repository deployed

This document assumes that these things are already in place.

1 Architectural View

There are four distinct components:

- The RSA DLP system
- The DLP Analysis tool for converting RSA DLP data into Courion DLP data
- The Certification Review Cycle Type "Sensitive Data Manager" for driving DLP data through ComplianceCourier
- A dashboard application for displaying metrics about identity and DLP violations

The following illustrates the architectural overview of the system:

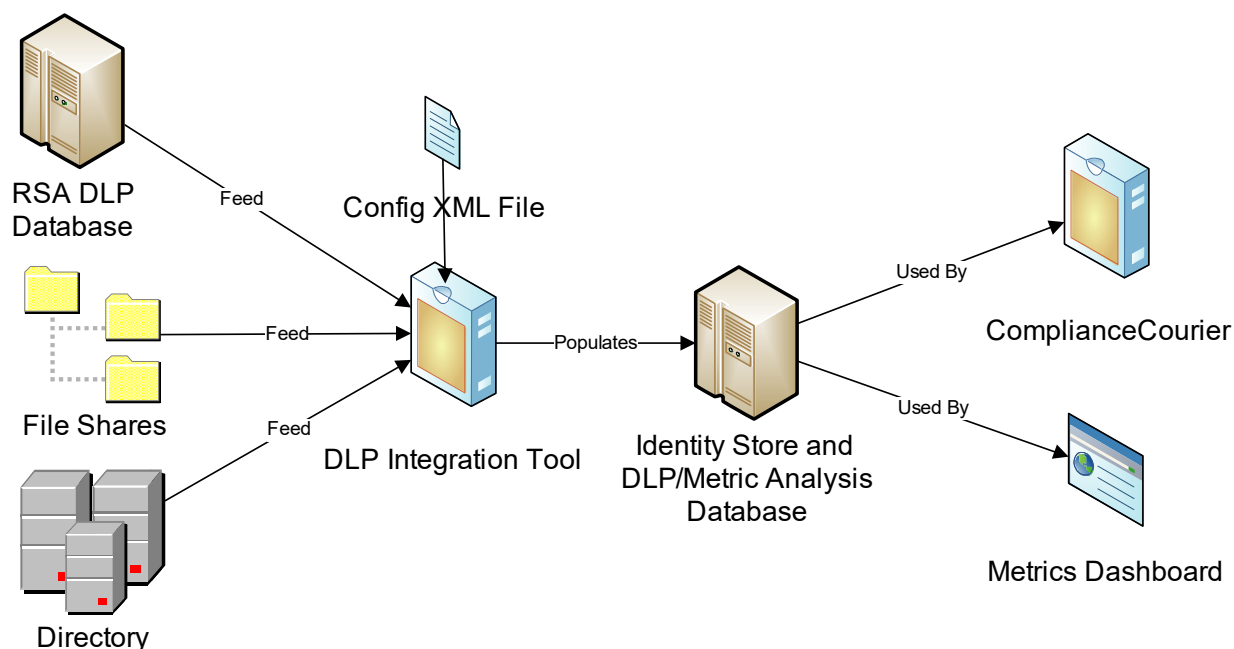


Figure 1: Architectural Overview

1 Building the DLP/Metric Analysis Database

It is assumed for each of the sections below that you have applied an RSA DLP sample schema DDL that is supplied with the suite installation. The document describes the functionality provided by this DDL throughout each of the sections but it should be applied before proceeding with this document to ensure that all of the described procedures run correctly.

The DDL is named *"DLPTargetDBSchema.ddl"* and is found in the CourionService directory. It must be applied to a version of the Transaction Repository that has been created or upgraded by v8.0 of the Core Access Assurance Suite. It creates tables, views, stored procedures, and user-defined functions. The user that applies the DDL must have appropriate access to create each of these in the target Transaction Repository.

Once the DDL is applied, a table called *"RSA_DLP_701_Integration"* is created that will represent the results of the analysis performed by the DLP Integration tool. Figure 2: *RSA_DLP_701_Integration* Schema describes the schema for this table.


RSA_DLP_701_Integration				
	Column Name	Condensed Type	Nullable	Description
	RecordID	int	No	Unique Key
	FileName	nvarchar(2000)	Yes	The name of the violating file
	[Access Via]	nvarchar(255)	Yes	The group provided access to the file
	Access	nvarchar(255)	Yes	The type of access permitted
	Target	nvarchar(255)	Yes	The target system the file resides on
	Username	nvarchar(255)	Yes	The username of a user with access
	[Special Group]	nvarchar(255)	Yes	"Yes" if it is an unrolled group
	Policy	nvarchar(255)	Yes	The name of the RSA DLP policy
	Severity	nvarchar(255)	Yes	The severity of the RSA DLP violation
	IncidentType	nvarchar(255)	Yes	The incident type of the RSA DLP violation
	[Content Blade]	nvarchar(255)	Yes	The content blade of the RSA DLP violation
	[Match Count]	int	Yes	The match count from RSA DLP
	[Risk Factor]	int	Yes	The risk factor from RSA DLP
	Status	nvarchar(255)	Yes	The status of the RSA DLP violation
	Validity	nvarchar(255)	Yes	The validity of the RSA DLP violation
	Owner	nvarchar(255)	Yes	The owner of the file involved
	Time	datetime	Yes	The time the analysis was performed
	CourionMitigation	nvarchar(511)	Yes	Text describing what, if any, mitigation was performed
	AtRestViolation	bit	Yes	True if it is a data at rest violation

Figure 2: RSA_DLP_701_Integration Schema

The schema is designed to mirror the fields that a user of the RSA DLP system would view on the violation screen. There are two additional fields added to the schema that are not shown in the RSA DLP product. They are "Target" and "Username". The DLP Integration tool populates these fields by taking the ACLs assigned to each "FileName" and cross-referencing the AD permissions with one or more AD domains. The result is domain names being placed into the "Target" field and Active Directory user account names being placed into the "Username" field.

PERFORM STEPS:

- 1) Open SQLServer Manager
- 2) Find and open the database that contains the Transaction Repository
- 3) Locate the "DLPTargetSchema.ddl" file and open it in an editor.
- 4) Open up a SQL query window in SQLServer Manager and copy the contents of the DDL.

- 5) Execute the SQL query window to create the new tables and views in the TR.

1 Running the DLP Integration Tool

To display RSA DLP sensitive data violations in a full access assurance context, Core Security performs analysis that ties individual users and user accounts to the violations using the permissions (ACLs) on the files that contain the sensitive data (as discovered by RSA DLP). To make these connections it is necessary to process the data produced by RSA DLP through an off-line analysis tool. The DLP Integration Tool is a Windows command line utility that requires minimal configuration and can be run either on demand or automated as part of a regular analysis process. It must be run before using any Core Security workflow or analytics that involves RSA DLP data.

PERFORM STEPS:

Before using the DLP integration tool prepare the following:

- 1) Determine which database server the RSA DLP database is hosted on and obtain credentials sufficient to query all of the tables/views
- 2) Obtain access to a machine on an AD domain which has trust relationships with all of the domains that file shares referenced by the RSA DLP tool reside on
- 3) Obtain access to an account which is an administrator of all of the domains that accounts with access to sensitive data will be discovered on. This account will be used to tie the file permissions of files with sensitive data to the users that have that access.

1.1 DLP Integration Tool Background

The DLP integration tool performs two key tasks:

- Transferring and processing the raw data that already exists in the RSA DLP database into a database staging table used by Core Security's products.
- Perform ACL lookups for the files identified by the native DLP tool.

The DLP Integration Tool requires a [configuration XML file](#) <XREF> to be passed as a command line argument to describe where the RSA DLP database lives, credentials for accessing it, where the Core Security Analysis/Database resides, its credentials, and the definition of how the RSA DLP database fields map onto the RSA_DLP_701_Integration schema.

As part of the analysis process in the DLP Integration tool, all ACL assignments attached to each file

(found to be in violation through RSA DLP) are added as individual entries into the “RSA_DLP_701_Integration” table. If an ACL assignment represents a group and not an account, that AD group is unrolled and the resolved list of AD accounts are added individually to the table as well.

NOTE: Because built-in groups like “Everyone” and “Authenticated User” can result in large lists of member users and/or false-positive member lists due to domain ambiguity, those groups are excluded (by default) from getting unrolled. The full list of groups excluded from the “Sample_RSA_DLP.xml” configuration file is as follows:

- Everyone
- Authenticated Users
- Domain Users
- Domain Admins
- Power Users
- Guests
- Domain Computers

1.1.1 Configuration XML File– (Sample_RSA_DLP.xml)

The file “Sample_RSA_DLP.xml” (located in the CourionService directory) is an example of a DLP integration tool configuration file. The following shows the overall structure of the XML file along with a description of the important parts that must be modified in order for the DLP Integration Tool to function properly.

```
<dlptool_config>
```

```
    <config_params>
```

```
        <params>
```

```
            <param>
```

```
                <key/>
```

```
                <value/>
```

```
        </param>
    </params>
</config_params>

<column_mappings>
    <mapping>
        <source_col/>
        <destination_col/>
    </mapping>
</column_mappings>

<groups>
    <group/>
</groups>

</dlptool_config>
```

1.1.1.1 *Configuration Parameters Section (<config_params>)*

The first section declares parameters that are used by the tool to define how it operates. The following parameters are supported by the DLP Integration tool:

- **SelectQuery** – This field represents the actual SELECT SQL query that will be used to extract data from a RSA DLP database and should at a minimum return at least one field representing the UNC path of the file that has been identified as an incident by the RSA DLP tool
- **TargetTable** – An output table that will store the results from the DLP Integration tool analysis. In the “Sample_RSA_DLP.xml” configuration file, the table supplied is the “RSA_DLP_701_Inte-

gration” table that was established earlier in this document.

- **InputConnectionString** - This is an ADO connection string that defines access to the source RSA DLP database on which the query defined in the “SelectQuery” parameter will be executed.
- **OutputConnectionString** - This is an ADO connection string that defines access to the Transaction Repository database in which the “RSA_DLP_701_Integration” table was created using the sample DDL above.
- **ClearTargetTable** - This is a property that will tell the DLP Integration tool if it should delete the contents of the target table (“RSA_DLP_701_Integration”) before adding new data. In the absence of this parameter, the default behavior is set to “true” which means the data will be cleared. Setting the value to “false” will force the DLP Integration tool to append the analysis onto the end of the table. There can be instances where the DLP Integration tool cannot talk to all of the domains from a single machine. In this scenario, it is advised that this parameter is set to “false” so that the DLP Integration tool can be run from multiple machines located in multiple domains so that the analysis results are accurate and complete.

1.1.1.1 *Field Mapping Section (<column_mappings>)*

The second section defines a mapping between the returned fields of the “SelectQuery” to the fields in the table defined by the “TargetTable” parameter above. Each mapping in this section must define a mapping from either a constant value or a valid field returned by the SQL query to a valid field defined in the table represented with the “TargetTable” parameter.

An example of this mapping can be shown below:

```
<config_params>
  <params>
    <param>
      <name>SelectQuery</name>
      <value>SELECT ViolationFile, TimeOfViolation FROM DLPViolation-
Source</value>
    <param>
    <param>
      <name>TargetTable</name>
      <value> RSA_DLP_701_Integration </value>
    <param>
  </params>
</config_params>
<column_mappings>
```

```

    <mapping isFilename="yes">
        <source_col>ViolationFile</source_col>
        <destination_col>FileName</destination_col>
    </mapping>
    <mapping>
        <source_col>TimeOfViolation</source_col>
        <destination_col>Time</destination_col>
    </mapping>
</column_mappings>

```

In the example above, the two fields “ViolationFile” and “TimeOfViolation” returned from the SQL Select statement are mapped to two fields defined in the target table “RSA_DLP_701_Integration”. The following outlines some rules around the behavior of these mappings:

- If two columns are defined then the value from the source column will be copied directly into the target column.

```

<mapping>
    <source_col>ViolationFile</source_col>
    <destination_col>FileName</destination_col>
</mapping>

```

- If the source has a 'useAsLiteral' attribute set to true, then the value provided will be taken literally and placed into the target column for every row regardless of content.

```

<mapping>
    <source_col useAsLiteral='true'>High</source_col>
    <destination_col>Severity</destination_col>
</mapping>

```

- The configuration file must, at a minimum, contain a mapping that includes the “FileName” field

defined in the `<destination_col/>` tag. Alternatively, the attribute “isFilename” can be leveraged to signify that the given mapping represents filenames if the field “FileName” cannot explicitly be used for that purpose. If this attribute is set to “yes” for more than one mapping, then this is considered a misconfiguration and the tool will immediately exit when detected.

- Any number of columns can be mapped between the source and target tables. Ensure that the target table has the columns specified as destination columns.

1.1.1.1 *Group exclusion Section (<groups>)*

The last section specifies a list of groups, which if encountered while performing ACL lookups, will not be unrolled. The “Sample_RSA_DLP.xml” configuration file already defines that certain Built-in groups be excluded from the rollout process. Any group that you wish not to be un-rolled can be added to this list.

CAUTION: The groups defined in this list represent a set of well-known containers that can result in very large collections of users (in the order of 1000’s per violation found) and/or ambiguity in proper domain resolution. Removing a group entry from this list constitutes that you understand the ramifications on both performance of the analysis database as well as the potential for establishing false-positives in the user populations retrieved on behalf of groups that resolve differently in an AD forest containing more than one domain.

1.1 Using the analysis tool to prepare data for the certification review cycle type and metrics

A sample RSA DLP configuration file has been supplied in the CourionService directory called Sample_RSA_DLP.xml. To prepare for the sample workflow and sample metrics, this configuration file requires the following manual modifications:

PERFORM STEPS:

- 1) Make a copy of the sample configuration file and modify the following parameters in the first section of the configuration file as follows:
 - a. **InputConnectionString** parameter – Supply the connection string that represents the RSA DLP output database. The example provided assumes integrated authentication for credentials using the “Trusted_Connection=True” parameter. The easiest way to build and test a connection string is to create a “*.udl” file and double-click on it to bring up

Microsoft's ADO connection string configuration GUI.

- b. **OutputConnectionString** parameter – Supply the connection string that represents the database where the Sample_StagingTable_SensitiveData_DLP staging table resides. The example provided assumes integrated authentication for credentials using the “Trust-ed_Connection=True” parameter. The easiest way to build and test a connection string is to create a “*.udl” file and double-click on it to bring up Microsoft's ADO connection string configuration GUI..
 - c. **ClearTargetTable** parameter – Defaults to “true” to ensure that only the most recent data is stored and to minimize database size. For valid trending in the sample analytics metrics this should be changed to “false”. However, if analytics will not be used then this should be left as “true”.
- 2) Copy the following files from the CourionService directory onto a machine with appropriate access to the RSA DLP database, Transaction Repository, along with ACL lookup capabilities on the domains that contain the file violations:
- a. DLPIntegrationTool.exe
 - b. DLPIntegrationTool.Resources.dll
 - c. Copy of Sample_RSA_DLP.xml (modified version from step 1 above)
 - d. Common.dll
 - e. Common.Resources.dll
 - f. CmdlineArgs.dll
 - g. Microsoft.Practices.EnterpriseLibrary.Common.dll
 - h. Microsoft.Practices.EnterpriseLibrary.ExceptionHandling.dll
 - i. Microsoft.Practices.EnterpriseLibrary.ExceptionHandling.Logging.dll
 - j. Microsoft.Practices.EnterpriseLibrary.Logging.dll
 - k. Microsoft.Practices.EnterpriseLibrary.ObjectBuilder.dll
 - l. CourProfiler.exe (Install using the “*CourProfiler.exe –service*” command followed by setting the appropriate DCOM privileges on the exposed COM object)

- 3) Ensure that the logged in user executing the DLP Integration tool has administrative access to

each of the domains that were flagged for having Sensitive data violations.

- a. It may not be possible to find a single such user. In this case, ensure that the “ClearTargetTable” parameter in the configuration XML file is set to “false”. Then, run the analysis from a set of machines.
- 4) Run the DLP integration tool with the following command line:
 - a. `DLPIntegrationTool.exe –configxml=“Sample_RSA_DLP.xml”`
- 5) Allow analysis to complete.
- 6) Check the “DLPIntegrationTool.log” (located in the same directory that the tool was executed from) for any problems that might have occurred during the analysis.
- 7) Verify that the analysis completed successfully by opening up the target table “RSA_DLP_701_Integration” through SQLServer Manager and examining the contents of the data.

1.1 Running the DLP Integration Tool in Encrypt Mode

The DLP Integration tool has another mode of operation – Encrypt Mode. If you do not wish to store user name, passwords in the configuration file, you may store them encrypted. You can use the same tool to perform the encryption.

`DLPIntegrationTool.exe –configxml=“Sample_RSA_DLP.xml” – Encrypt`

Run the tool with the “-Encrypt” switch. The tool will then prompt you for the fields that can be encrypted. The following fields can be encrypted

- 1) SQL Server UserID for InputConnectionString
- 2) SQLServer Password for InputConnectionString
- 3) SQL Server UserID for OutputConnectionString
- 4) SQLServer Password for OutputConnectionString

Note: If you run the tool in normal mode (without encrypt) and if any of the fields are encrypted, the tool will automatically decrypt those fields and continue execution

1 The RSA DLP Sample Metrics

In addition to the sample workflow, a set of sample metrics have been provided for the Core Security

Advanced Analytics framework which was introduced in v8.0. The supplied metrics provide an overview on some of the performance indicators that can be created using the combination of RSA DLP data and the Core Access Assurance Suite. Some of these metrics require workflow/business process configuration outside of what has already been described above. Also, some of the metrics require relatively long running analysis operations (on the order of minutes or more). Sample analysis queries have been included in the RSA DLP Schema as stored procedures so that they can be included as part of scheduled batch processes using SSIS or any other engine that provides ADO integration.

In the remainder of this section each metric will be described and any operations necessary to get the metric running will be detailed. For more information on modifying the sample metrics look to the Advanced Analytics Metric Design Guide that will also be supplied with the v8.0 Release. Also, any display argument customizations that are available will be detailed. All display arguments can be modified by accessing the “MetricDisplayArguments” table that was laid down with the Transaction Repository.

Some of the sample metrics refer to “in-motion” sensitive data violations. The default analysis XML configuration file “sample_RSA_DLP.xml” will **not** supply any of these kinds of violations. The configuration file consumed by the DLP Integration tool will only return “at rest” RSA DLP violations and will always set the “AtRestViolation” field of the “RSA_DLP_701_Integration” table to 1 (representing TRUE). To use the “In-motion” metrics, it will be necessary to determine an appropriate SQL query into the RSA DLP database and set up a configuration XML file that defines this query as well as sets the “AtRestViolation” to 0 (representing FALSE).

NOTE: In all cases, each metric requires its associated stored procedure to be run before the metric data will become available for viewing.

Each of the following sections outlines:

- Brief description of the metric and its purpose
- Steps to perform for establishing the metric
- A list of “Display Arguments” that can be adjusted through the “MetricDisplayArguments” table for how the metric is calculated and/or displayed to the user
- A list of other metrics that the current metric can drill down into when the mouse right-clicks on a part of the displayed graph/chart

1.1 Configure the RSA DLP Metric Data Source

The RSA DLP Sample Metrics depend on a default data source which uses the connection string “RsaDlp-DataSource”. This data source should point to the Transaction Repository where you have applied the

RSA DLP sample DDL. More details about configuring Metric DataSource can be found in the [*Advanced Analytics Metric Design Guide*](#).

PERFORM STEPS:

- 1) Determine a valid SQL connection string to the database where the sample DDL has been deployed. One way to do this is to generate a *.udl file for the database and retrieve the connection string generated.
- 2) Open the web.config file under the www\Analytics folder
- 3) Find the section “connectionStrings” and uncomment the <add/> tag that has the name attribute set to “MetricRepositoryDefault”.
- 4) Replace the contents of the connectionString attribute with the string generated from step 1 above. This <add/> tag will represent the ADO connection that represents all of the configuration required to properly show metrics in the “Access Assurance Suite Advanced Analytics” product.
- 5) Add another <add> tag as follows: `<add name="RsaDlpDataSource" connectionString="" providerName="System.Data.SqlClient"/>`
- 6) In the connectionString attribute, add the connection string constructed from Step 1 above. This <add/> tag will represent the ADO connection that references all of the data that feeds metric charts and graphs .

1.1 At Rest Sensitive Data Violation Trend

This metric shows the trend of accounts with sensitive data access over time (with the granularity of one day). This differs from the type of metrics that RSA DLP can display because we have tied the violations to particular accounts with access by unrolling the file permissions (ACLs) and Active Directory groups. So, if a single violation is found to affect a large user population, that condition would become immediately obvious using this trending metric.

Three statistical trends are provided for comparison purposes: 90 day minimum, 90 day rolling average, and 90 day maximum. The length of the history should be decided at analysis time and then recorded in the appropriate Display Argument.

The “RSADLPMetrics_AtRestDataViolationTrend” table captures the results of executing the stored procedure and is used to store the metric data prepared for display. The format of the table can be found in Figure 8.

RSADLPMetrics_AtRestDataViolationTrend				
	Column Name	Condensed Type	Nullable	Description
	Date	datetime	No	Date of analysis
	AtRestDataViolations	int	No	Count of accounts in violation on that day
	AtRestDataViolationsMin	int	No	Minimum accounts in violation in past N days
	AtRestDataViolationsAvg	float	No	Average accounts in violation over past N days
	AtRestDataViolationsMax	int	No	Maximum accounts in violation in past N days

Figure 3: RSADLPMetrics_AtRestDataViolationTrend Table

PERFORM STEPS:

- 1) Execute the DLP Integration tool using the "sample_RSA_DLP.xml" configuration file to populate the "RSA_DLP_701_Integration" table if this hasn't been done already.
- 2) Open up SQLServer Manager and execute the "RSADLP_PopulateAtRestTrends" stored procedure.
- 3) When asked for a value for "HistoryLength", enter in the number of days to consider in the rolling statistics window that the trend will cover. This value looks at the date of each violation occurrence relative to current target date and time to determine the range of violations to consider. The default length is 90 days. If a value other than 90 is entered here the Display Argument "HistoryLength" should be modified to match this value to ensure that the baselines are appropriately labeled.
- 4) Examine the "RSADLPMetrics_AtRestDataViolationTrend" table to see that it is populated with data.

1.1.1 Display Argument Customizations

The following Display Arguments can be customized for this metric by modifying the values in the Metric-DisplayArguments table which are mapped to the metric named "RSADLP-AtRestTrend".

The HistoryLength value should match the number of days entered while running the analysis stored procedure.

Display Argument	Description	Default Value
DisplayLength	Number of days to show in chart	90

HistoryLength	Number of days used during analysis (shown for stats)	90
---------------	---	----

Table 1: Display Arguments for the metric “At Rest Sensitive Data Violation Trend”

1.1.1 Related Metrics (Drill-down)

This can drill down to two metrics: “At Rest Violation Count by Department” and “At Rest Violation Count by Role”.

1.1 At Rest Violation Count by Department

This metric displays the 5 most popular departments that have access to sensitive data (as judged by unique accounts with access to sensitive data). This metric is reported for a certain “TargetDate” which is either supplied as a Display Argument or through drill-down from another metric (such as the “At Rest Sensitive Data Violation Trend”). A baseline is provided representing the previous 7 day average.

The “RSADLPMetrics_AtRestDepartmentViolationCount” table captures the results of executing the stored procedure and is used to store the metric data prepared for display. The table format can be found in Figure 4: RSADLPMetrics_AtRestDepartmentViolationCount Table.

RSADLPMetrics_AtRestDepartmentViolationCount				
	Column Name	Condensed Type	Nullable	Description
	Date	datetime	No	Date of analysis
	Department	nvarchar(255)	No	Department name
	UsersInViolation	int	No	The number of accounts in violation on the given day

Figure 4: RSADLPMetrics_AtRestDepartmentViolationCount Table

PERFORM STEPS:

- 1) Execute the DLP Integration tool using the “sample_RSA_DLP.xml” configuration file to populate the “RSA_DLP_701_Integration” table if this hasn’t been done already.
- 2) Validate that the IdentityMap is properly populated. See section “Populating the IdentityMap” for more instructions.
- 3) Validate that the “SampleProfile” table in the Transaction Repository contains profile entries that properly map to the IdentityMap (“ProfileUID” field in the IdentityMap is the same AD acct that is also populated in the “EmployeeID” field of the “SampleProfile” table). This table can be prop-

erly populated using ProfileCourier sample workflows that ship with the product (such as “Self-Service Profile Management”).

- 4) Open up SQLServer Manager and execute the “RSADLP_PopulateAtRestDeptCount” stored procedure.
- 5) When asked for a value for “HistoryLength”, enter in the number of days to consider in the rolling statistics window that the trend will cover. This value looks at the date of each violation occurrence relative to the target date and time to determine the range of violations to consider.
- 6) Examine the “RSADLPMetrics_AtRestDepartmentViolationCount” table to see that it is populated with data.

1.1.1 Display Argument Customizations

The following Display Arguments can be customized for this metric by modifying the values in the Metric-DisplayArguments table which are mapped to the metric named “RSADLP-AtRestTopDepartments”.

Notice that the default values use macros. These macros are part of the Analytics framework and are covered in the Advanced Analytics Metric Design Guide.

Display Argument	Description	Default Value
HeaderText	Label to show context of chart	Top %DisplayArgument.TopN% Departments on %DisplayArgument.Target-Date%
HistoryLength	Number of days of history to use for baseline	7
TargetDate	The default date to report on	12/17/2009
TopN	The number of departments to include	5

Table 2: Display Arguments for the At Rest Violation Count by Department metric

1.1.1 Related Metrics (Drill-down)

None.

1.1 At Rest Violation Count by Role

This metric displays the 5 most popular roles that have access to sensitive data (as judged by unique

accounts with access to sensitive data). This metric is reported for a certain “Target Date” which is either supplied as a Display Argument or by drill-down from another metric. A baseline is provided representing the previous 7 day average.

The “RSADLPMetrics_AtRestDepartmentViolationCount” table captures the results of executing the stored procedure and is used to store the metric data prepared for display. The table format can be found in Figure 5: RSADLPMetrics_AtRestRoleViolationCount Table.

RSADLPMetrics_AtRestRoleViolationCount				
	Column Name	Condensed ...	N...	Description
	Date	datetime	No	Date of analysis
	Role	nvarchar(255)	No	Role name
	UsersInViolation	int	No	The number of accounts in violation on the given day

Figure 5: RSADLPMetrics_AtRestRoleViolationCount Table

PERFORM STEPS:

- 1) Execute the DLP Integration tool using the “sample_RSA_DLP.xml” configuration file to populate the “RSA_DLP_701_Integration” table if this hasn’t been done already.
- 2) Validate that the IdentityMap is properly populated. See section “Populating the IdentityMap” for more instructions.
- 3) Populate the RSADLP_RoleMap table (see Figure 6: RSADLP_RoleMap table). This table contains a mapping between users represented by the ProfileID and the business-friendly role name. This table can either be populated as part of a role creation/maintenance workflow or can be replaced by a view into the existing role map that exists for an implementation. Though this data may already exist, the easiest way to configure the metrics is to create a view into the data so that the existing queries and stored procedures can work as written.
- 4) Open up SQLServer Manager and execute the “RSADLP_PopulateAtRestRoleCount” stored procedure.
- 5) Examine the RSADLPMetrics_AtRestRoleViolationCount table to ensure that it has been populated.

RSADLP_RoleMap				
	Column Name	Condensed Type	Nullable	Description
	RSARoleMapID	int	No	Primary Key
	ProfileID	nvarchar(255)	Yes	Profile Unique ID
	[Role Name]	nvarchar(255)	Yes	Descriptive role name

Figure 6: RSADLP_RoleMap table

1.1.1 Display Argument Customizations

The following Display Arguments can be customized for this metric by modifying the values in the Metric-DisplayArguments table which are mapped to the metric named “RSADLP-AtRestTopRoles”. Notice that the default values use macros. These macros are part of the Analytics framework and are covered in the Advanced Analytics Metric Design Guide.

Display Argument	Description	Default Value
HeaderText	Label to show context of chart	Top %DisplayArgument.TopN% Roles on %DisplayArgument.TargetDate%
HistoryLength	Number of days of history to use for baseline	7
TargetDate	The default date to report on	12/17/2009
TopN	The number of roles to include	5

Table 3: Display Argument for At Rest Violation Count by Role

1.1.1 Related Metrics (Drill-down)

Drills down to [At Rest Violation Count by Role and Regulation<XREF>](#)

1.1 At Rest Violation Count by Role and Regulation

This metric shows the number of accounts that had access to particular types of sensitive data on a target date. The types of sensitive data are categorized by regulatory categories establishes as part of the sample metric and sample workflow configuration. The target date is determined either by a default Display Argument or through drill-down from the [At Rest Violation Count by Role <XREF>](#) metric.

The “RSADLPMetrics_AtRestRoleRegViolationCount” table captures the results of executing the stored procedure and is used to store the metric data prepared for display. The format of the table can be seen in Figure 7: RSADLPMetrics_AtRestRoleRegViolationCount Table.

RSADLPMetrics_AtRestRoleRegulationCount				
	Column Name	Condensed Type	Nullable	Description
	Date	datetime	No	Date of analysis
	Regulation	nvarchar(255)	No	Name of regulation violated
	RoleName	nvarchar(255)	No	Name of role accounts are associated with
	UserCount	int	No	Count of accounts violating the given reg. in the given role

Figure 7: RSADLPMetrics_AtRestRoleRegViolationCount Table

PERFORM STEPS:

- 1) Execute the DLP Integration tool using the “sample_RSA_DLP.xml” configuration file to populate the “RSA_DLP_701_Integration” table if this hasn’t been done already.
- 2) Validate that the IdentityMap is properly populated. See section “Populating the IdentityMap” for more instructions.
- 3) Populate the RSADLP_RoleMap table. See the “Perform Steps” section of the sample metric [At Rest Violation Count by Role](#) <XREF>for more detail.
- 4) Ensure that the RSADLP_ContentToRegulationMap table is populated. See Section for more information on configuring this table.
- 5) Open up SQLServer Manager and execute the “RSADLP_PopulateAtRestRoleRegCount” stored procedure.
- 6) Examine the RSADLPMetrics_AtRestRoleRegulationCount table to ensure that it has been populated.

1.1.1 Display Argument Customizations

The following Display Arguments can be customized for this metric by modifying the values in the Metric-DisplayArguments table which are mapped to the metric named “RSADLP-AtRestRoleRegulations”. Notice that the default values use macros. These macros are part of the Analytics framework and are covered in the Advanced Analytics Metric Design Guide.

Display Argument	Description	Default Value
------------------	-------------	---------------

HeaderText	Label to show context of chart	Regulations Violated By %DisplayArgument.TargetRole% on %DisplayArgument.TargetDate%
TargetRole	The default role to report on	CxO
TargetDate	The default date to report on	12/17/2009

Table 4: Display Arguments for At Rest Violation Count by Role and Regulation

1.1.1 Related Metrics (Drill-down)

None.

1.1 At Rest Unmitigated Sensitive Data Violation Trend

This metric shows the trend of accounts with **unmitigated** sensitive data access over time (with the granularity of one day). Sensitive data access is considered unmitigated if the **CourionMitigation** column in the RSA_DLP_701_Integration table is left with its default NULL value.

Three statistical trends are provided for comparison purposes: 90 day minimum, 90 day rolling average, and 90 day maximum. The length of the history should be decided at analysis time and then recorded in the appropriate Display Argument.

The “RSADLPMetrics_AtRestUnmitigatedViolationTrend” table captures the results of executing the stored procedure and is used to store the metric data prepared for display. The format of the table can be seen in Figure 8: RSADLPMetrics_AtRestUnmitigatedViolationTrend Table.

RSADLPMetrics_AtRestUnmitigatedViolationTrend				
	Column Name	Condensed Type	Nullable	Description
	Date	datetime	No	Date of analysis
	AtRestDataViolations	int	No	Count of unmitigated accounts in violation
	AtRestDataViolationsMin	int	No	Minimum unmitigated accounts in violation in past N days
	AtRestDataViolationsAvg	float	No	Average unmitigated accounts in violation over past N days
	AtRestDataViolationsMax	int	No	Maximum unmitigated accounts in violation in past N days

Figure 8: RSADLPMetrics_AtRestUnmitigatedViolationTrend Table

PERFORM STEPS:

- 1) Execute the DLP Integration tool using the “sample_RSA_DLP.xml” configuration file to populate the “RSA_DLP_701_Integration” table if this hasn’t been done already.

- 2) Validate that the IdentityMap is properly populated. See section “Populating the IdentityMap” for more instructions.
- 3) Modify the existing sample workflow, or existing business process, to populate the CourionMitigation column of the RSA_DLP_701_Integration table to contain non-null values once attestation or remediation has occurred.
- 4) Open up SQLServer Manager and execute the “RSADLP_PopulateAtRestUnmitigatedTrends” stored procedure.
- 5) When asked for a value for “HistoryLength”, enter in the number of days to consider in the rolling statistics window that the trend will cover. This value looks at the date of each violation occurrence relative to the target date and time to determine the range of violations to consider.
- 6) Examine the RSADLPMetrics_AtRestUnmitigatedDataViolationTrend table to ensure that it has been populated.

1.1.1 Display Argument Customizations

The following Display Arguments can be customized for this metric by modifying the values in the Metric-DisplayArguments table which are mapped to the metric named “RSADLP-AtRestUnmitigatedTrend”. The HistoryLength Display Argument value should match the value entered when running the stored procedure to ensure that the baseline trends are appropriately labeled.

Display Argument	Description	Default Value
DisplayLength	Number of days to show in chart	90
HistoryLength	Number of days used during analysis (shown for stats)	90

Table 5: Display Arguments for the At Rest Unmitigated Sensitive Data Violation Trend metric

1.1.1 Related Metrics (Drill-down)

This metric can drill down to the [Unmitigated Sensitive Data Access by Manager and Regulation](#) <XREF>metric.

1.1 At Rest Unmitigated Sensitive Data Violation by Manager and Regulation

This metric shows the top 5 most popular managers who have employees with unmitigated sensitive data access for 4 target regulations on a target date. The possible regulation categories are defined in the RSADLP_ContentToRegulationMap which was previously configured.

The metric displays violations categorized by four target regulations. The regulations targeted can be customized by changing the Display Arguments for this metric.

The “RSADLPMetrics_AtRestUnmitigatedMgrRegCount” table captures the results of executing the stored procedure and is used to store the metric data prepared for display. The format of the table can be seen in Figure 9: The RSADLPMetrics_AtRestUnmitigatedMgrRegCount Table.

RSADLPMetrics_AtRestUnmitigatedMgrRegCount				
	Column Name	Condensed Type	Nullable	Description
	Date	datetime	No	Date of analysis
	Manager	nvarchar(255)	No	Manager employee ID
	Regulation	nvarchar(255)	No	Name of regulation
	UserCount	int	No	Unmitigated accounts in violation for manager and reg.

Figure 9: The RSADLPMetrics_AtRestUnmitigatedMgrRegCount Table

PERFORM STEPS:

- 1) Execute the DLP Integration tool using the “sample_RSA_DLP.xml” configuration file to populate the “RSA_DLP_701_Integration” table if this hasn’t been done already.
- 2) Validate that the IdentityMap is properly populated. See section “Populating the IdentityMap” for more instructions.
- 3) Validate that the “SampleProfile” table in the Transaction Repository contains profile entries that properly map to the IdentityMap (“ProfileUID” field in the IdentityMap is the same AD acct that is also populated in the “EmployeeID” field of the “SampleProfile” table). This table can be properly populated using ProfileCourier sample workflows that ship with the product (such as “Self-Service Profile Management”).
- 4) Validate that each entry in the “SampleProfile” table has a value in the “ManagerEmployeeID” field. This field is what represents the ‘Manager’ in this metric.

- 5) Modify the existing sample workflow, or existing business process, to populate the CourionMitigation column of the RSA_DLP_701_Integration table to contain non-null values once attestation or remediation has occurred.
- 6) Ensure that the RSADLP_ContentToRegulationMap table is populated. See Section for more information on configuring this table.
- 7) Open up SQLServer Manager and execute the “RSADLP_PopulateAtRestUnmitigatedMgrRegCount” stored procedure.
- 8) Examine the RSADLPMetrics_AtRestUnmitigatedMgrRegCount table to ensure that it has been populated.

1.1.1 Display Argument Customizations

The following Display Arguments can be customized for this metric by modifying the values in the MetricDisplayArguments table which are mapped to the metric named “RSADLP-AtRestUnmitigatedByRegAnd-Manager”. The Display Arguments Reg1, Reg2, Reg3, and Reg4 can be customized to determine which regulation categories should be considered.

Display Argument	Description	Default Value
TargetDate	Date to report on	12/17/2009
TopN	Number of managers to show	5
Reg1	First regulation to show	HIPAA
Reg2	Second regulation to show	PCI
Reg3	Third regulation to show	Corporate Policy
Reg4	Fourth regulation to show	SOX

Table 6: Display Arguments for the At Rest Unmitigated Sensitive Data Violations by Regulation and Manager metric

1.1.1 Related Metrics (Drill-down)

None

1.1 In Motion Sensitive Data Violations by Role

Note: this will not work without a customized analysis configuration (see the introduction of this section).

This metric displays the top 5 most popular roles as judged by the number of accounts assigned to a given role who commit in-motion sensitive data violations on a target day.

The “RSADLPMetrics_InMotionRoleViolationCount” table captures the results of executing the stored procedure and is used to store the metric data prepared for display. The format of the table can be seen in Figure 10: RSADLPMetrics_InMotionRoleViolationCount Table.

RSADLPMetrics_InMotionRoleViolationCount				
	Column Name	Condensed Type	Nullable	Description
	Date	datetime	No	Date of analysis
	Role	nvarchar(255)	No	The role name
	UsersInViolation	int	No	Number of accounts with violations for the role

Figure 10: RSADLPMetrics_InMotionRoleViolationCount Table

PERFORM STEPS:

- 1) Create a new DLP integration tool configuration XML file that captures in-motion sensitive data violations. Use this file to populate the RSA_DLP_701_Integration table with in-motion violations in addition to the at-rest violations that are found by default.
- 2) Validate that the IdentityMap is properly populated. See section “Populating the IdentityMap” for more instructions.
- 3) Populate the RSADLP_RoleMap table. See the “Perform Steps” section of the sample metric [At Rest Violation Count by Role](#) <XREF>for more detail.
- 4) Open up SQLServer Manager and execute the “RSADLP_PopulateInMotionRoleCount” stored procedure.
- 5) Examine the RSADLPMetrics_InMotionRoleViolationCount table to ensure that it has been populated.

1.1.1 Display Argument Customizations

The following Display Arguments can be customized for this metric by modifying the values in the Metric-DisplayArguments table which are mapped to the metric named “RSADLP-InMotionTopRoles”. Notice that the default values use macros. These macros are part of the Analytics framework and are covered in the Advanced Analytics Metric Design Guide.

Display Argument	Description	Default Value
------------------	-------------	---------------

HeaderText	The label to show context	Top %DisplayArgument.TopN% Roles on %DisplayArgument.TargetDate%
TopN	Number of roles to show	5
HistoryLength	Number of days to use in baseline average	7
TargetDate	The date to report on	12/17/2009

Table 7: Display Arguments for the In Motion Sensitive Data Violations by Role metric

1.1.1 Related Metrics (Drill-down)

This drills down to the [In Motion Sensitive Violation Trend](#) <XREF>metric.

1.1 In Motion Sensitive Violation Trend

Note: this will not work without a customized analysis configuration (see the introduction of this section).

This metric shows the number of accounts committing in-motion sensitive data violations over time for accounts associated with a specific role. The target role is defined either as a default role as a Display Argument or through drill-down from the [In Motion Sensitive Data Violations by Role](#) <XREF>metric.

Three statistic trends are provided for comparison purposes: 90 day minimum, 90 day rolling average, and 90 day maximum. The length of the history should be decided at analysis time and then recorded in the appropriate Display Argument.

The “RSADLPMetrics_InMotionDataViolationTrend” table captures the results of executing the stored procedure and is used to store the metric data prepared for display. The format of the table can be seen in Figure 11: RSADLPMetrics_InMotionDataViolationTrend Table.

RSADLPMetrics_InMotionDataViolationTrend				
	Column Name	Condensed Type	Nullable	Description
	Date	datetime	No	Date of analysis
	DataViolations	int	No	Number of in motion violations
	DataViolationsMin	int	No	Min violations in the last N days
	DataViolationsAvg	float	No	Average violations over the last N days
	DataViolationsMax	int	No	Max violations in the last N days
	Role	nvarchar(255)	Yes	Role of accounts

Figure 11: RSADLPMetrics_InMotionDataViolationTrend Table

PERFORM STEPS:

- 1) Create a new DLP integration tool configuration XML file that captures in-motion sensitive data violations. Use this file to populate the RSA_DLP_701_Integration table with in-motion violations in addition to the at-rest violations that are found by default.
- 2) Validate that the IdentityMap is properly populated. See section “Populating the IdentityMap” for more instructions.
- 3) Populate the RSADLP_RoleMap table. See the “Perform Steps” section of the sample metric [At Rest Violation Count by Role](#) <XREF>for more detail.
- 4) Open up SQLServer Manager and execute the “RSADLP_PopulateInMotionTrends” stored procedure.
- 5) When asked for a value for “HistoryLength”, enter in the number of days to consider in the rolling statistics window that the trend will cover. This value looks at the date of each violation occurrence relative to the target date and time to determine the range of violations to consider.
- 6) Examine the RSADLPMetrics_InMotionDataViolationTrend table to ensure that it has been populated.

1.1.1 Display Argument Customizations

The following Display Arguments can be customized for this metric by modifying the values in the Metric-DisplayArguments table which are mapped to the metric named “RSADLP-InMotionTrend”. Notice that the default values use macros. These macros are part of the Analytics framework and are covered in the Advanced Analytics Metric Design Guide.

Display Argument	Description	Default Value
DisplayLength	Number of days to show in chart	90
HistoryLength	Number of days used in the history calculations	90
RoleName	The name of the role being filtered on	CxO
HeaderText	The label to show context	Trend for %DisplayArgument.RoleName%

Table 8: Display Arguments for the In Motion Sensitive Data Violation Trend metric

1.1.1 Related Metrics (Drill-down)

None.

1.1 In Motion Unmitigated Sensitive Data Violations by Manager and Action

Note: this will not work without a customized analysis configuration (see the introduction of this section).

This metric displays the top 5 most popular managers ranked by the number of direct reports who have unmitigated in-motion sensitive-data violations on a given day. The direct reports are categorized by the action that was occurring when the sensitive data violation was detected. Four action categories are considered with the actions determined by Display Arguments.

The “RSADLPMetrics_InMotionUnmitigatedMgrActCount” table captures the results of executing the stored procedure and is used to store the metric data prepared for display. The format of the table can be seen in Figure 12: RSADLPMetrics_InMotionUnmitigatedMgrActCount table.

RSADLPMetrics_InMotionUnmitigatedMgrActCount				
	Column Name	Condensed Type	Nullable	Description
	Date	datetime	No	Date of analysis
	Manager	nvarchar(255)	No	Manager Employee ID
	Action	nvarchar(255)	No	Name of the violating action
	UserCount	int	No	Count of violating accounts

Figure 12: RSADLPMetrics_InMotionUnmitigatedMgrActCount table

PERFORM STEPS:

- 1) Create a new DLP integration tool configuration XML file that captures in-motion sensitive data violations. Use this file to populate the RSA_DLP_701_Integration table with in-motion violations in addition to the at-rest violations that are found by default. The “access” column of the RSA_DLP_701_Integration should contain the action name in this case.
- 2) Validate that the IdentityMap is properly populated. See section “Populating the IdentityMap” for more instructions.
- 3) Validate that the “SampleProfile” table in the Transaction Repository contains profile entries that

properly map to the IdentityMap (“ProfileUID” field in the IdentityMap is the same AD acct that is also populated in the “EmployeeID” field of the “SampleProfile” table). This table can be properly populated using ProfileCourier sample workflows that ship with the product (such as “Self-Service Profile Management”).

- 4) Validate that each entry in the “SampleProfile” table has a value in the “ManagerEmployeeID” field. This field is what represents the ‘Manager’ in this metric.
- 5) Open up SQLServer Manager and execute the “RSADLP_PopulateInMotionUnmitigatedMgrActCount” stored procedure.
- 6) Examine the RSADLPMetrics_InMotionUnmitigatedMgrActCount table to ensure that it has been populated.

1.1.1.1 Display Argument Customizations

The following Display Arguments can be customized for this metric by modifying the values in the Metric-DisplayArguments table which are mapped to the metric named “RSADLP-InMotionUnmitigatedByActAndManager”. Notice that the default values use macros. These macros are part of the Analytics framework and are covered in the Advanced Analytics Metric Design Guide.

Display Argument	Description	Default Value
TargetDate	Date to report on	12/17/2009
TopN	Number of managers to show	5
Act1	First action to show	E-mail
Act2	Second action to show	Instant Message
Act3	Third action to show	Web Post
Act4	Fourth action to show	File Transfer
HeaderText	Label to show context	On %DisplayArgument.Targetdate%

Table 9:

1.1.1.1 Related Metrics (Drill-down)

None.

1 Starting up the Advanced Analytics Dashboard

Once all of the metrics have been configured and populated, it is time to show the results of the analysis.

The “Access Assurance Suite Advanced Analytics” dashboard is located in the “Analytics” directory underneath the “Courion” virtual directory.

PERFORM STEPS:

- 1) Define a URL that points to the Advanced Analytics dashboard. This is located in the Analytics directory underneath “Courion” virtual directory and the starting page is called “default.aspx”. For example: <http://myserver/courion/analytics/default.aspx>.
- 2) Enter the URL into a browser and supply AD credentials if they are asked.
- 3) Once the dashboard displays, open up the “Metrics” panel on the left and drag a metric over to the viewing area on the right. Repeat as necessary for each metric that you are interested in examining.

If an error window pops up stating that the metric repository could not be contacted, it’s most likely a problem with the connection strings defined in the web.config file within the Analytics directory. Refer to “Configure the RSA DLP Metric Data Source” section for more information on how to properly set up the ADO connections required for the dashboard application.

NOTE: After dragging a metric to the right viewing area, it is possible that a message will show stating that access to a particular *.png image file is denied. It is problem centered around file permissions. Refer to the section “Correcting File Permission Issues” in the “Advanced Analytics Deployment Guide” for more details on how to remedy the situation.