jSONAR
Simplifying Security

SONAR G

Connecting Guardium Systems to SonarG

TECH NOTE

SonarG is a powerful platform for capturing, managing and analyzing TB's of data using its fully integrated high performance analytics engines and highly efficient storage. The underlying data store within SonarG leverages a JSON-native architecture in order to easily and rapidly ingest data from a wide variety of sources and exploit the tremendous flexibility of NoSQL "schema on read" to easily create richer context via data integrations. This capability is especially valuable in the context of DB activity monitoring and security portals, as customers would like to blend data from DAM, VA, classifiers, CMDBs and many other sources in order to stitch together disparate pieces of data into a cohesive, 360° view of the security and compliance profile for all their databases.

Data is fed directly from Guardium collectors into a SonarG repository using a data transfer mechanism developed in collaboration with IBM. The core underlying technology relies on Guardium Datamarts, enhanced to make use of the unique capabilities and resources of SonarG. Specifically, the interface was optimized in order to reduce the processing currently performed on each collector while at the same time enabling increased volumes of raw data to be consolidated into SonarG on an hourly basis, as opposed to daily audit processes.

This paper provides details on how to configure the data exchange mechanism in order to enable data to flow from large numbers of Guardium collectors into a single SonarG node, where it is processed into a high performance repository and made available for reporting, analysis, dashboards, etc.

Steps Covered in this Technote

LIST OF FIGURES

DATA TRANSFER AT A HIGH LEVEL

The SonarG data import process relies on the execution of datamarts on both Guardium collectors and central managers in order to organize and package the appropriate data to be exported on a recurring basis from Guardium into SonarG. Datamarts were introduced in Guardium V9 and are a powerful mechanism for enabling query reports to be generated on an ongoing basis without the need for aggregation processes and in fact without the need for aggregators at all. As it relates to SonarG, datamarts are used to generate a variety of data extracts that are then published as .csv files for transfer to the SonarG system.

These data extract files are typically published on an hourly schedule, although this varies depending on the specific data sets. For example, operational data such as STAP health and key system parameters is published every 5 minutes in order to further reduce the latency of information and improve the ability to respond to issues as they arise. Data such as Buff Usage Monitor or VA results is published on a daily schedule.

At the data transfer level there are a number of advantages to the SonarG approach, including the following:

- *Completely eliminate Aggregators from the Guardium architecture, removing a significant layer of hardware, process complexity and system instability.*

- *Adopt an hourly transfer of data, as opposed to the current daily transfers with aggregation.*

- *Significantly expand the volume of data collected and retained, while at the same time reducing the effort required to manage data "lifecycles" on the various Guardium appliances.*

- *Increased Guardium Collector throughput by shifting various data processing tasks to SonarG where they can be executed more efficiently.*

SonarG simplifies the Guardium architecture and enables clients to substantially improve the value and efficiency of their DB security and compliance investment.

1 CONFIGURING GUARDIUM APPLIANCES

SonarG has developed a script to automatically configure the transfer of data from Guardium appliances into the SonarG environment. This script, called "sonarcli," executes a series of grdapi commands on both the Guardium Central Manager(s) and the Collectors in order to properly configure and enable the appropriate datamarts and also to schedule the recurring data transfers. Following execution of these commands, the script will also validate the output of their execution to confirm successful execution or any exceptions. The execution of the sonarcli.exp script is a one time event performed during the initial integration of SonarG into the Guardium environment and will only need to be re-executed should configuration parameters change. Adding new Guardium appliances, changing the selection of datamarts to be executed, etc., would trigger the need to repeat the script execution and re-executing these scripts will overwrite the previous settings for the affected appliances.

There are two options for configuring the data transfer mechanism between Guardium appliances and SonarG: The "Direct Method" enables data extract files to be sent as csv files via SCP to a specific directory on the SonarG machine and the "Staging Method" relies on an intermediary server that receives data extract files via SCP from the Guardium appliances and these files are subsequently pulled from this directory by SonarG. In most cases the Direct Method will be used and the trade-offs of both will be explored in detail during the implementation planning process. Either option is easily implemented and can be changed at a later date.

For a POC, many customers will choose to use the Staging Method in order to begin collecting data in parallel with securing the machine needed to host the SonarG application during the POC. This enables customers to stockpile large volumes of data and then once the SonarG machine is available the data can be easily ingested and becomes available for reporting, analysis, etc.

1.1 RUNNING SONARCLI

Sonarcli is easy to configure and execute, resulting in the automated configuration of data transfers between Guardium appliances and SonarG. At a high level, sonarcli combines a customer-provided list of Guardium appliances with a set of SonarG/IBM pre-defined datamarts and then communicates with all Guardium appliances to execute and validate the grdapi commands necessary to establish this communication. Script execution takes a matter of minutes and once completed SonarG will begin receiving rich data extracts from all of the Guardium appliances on an ongoing basis.

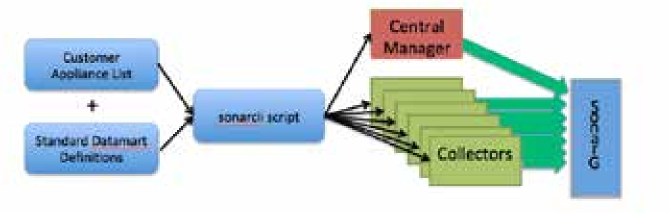
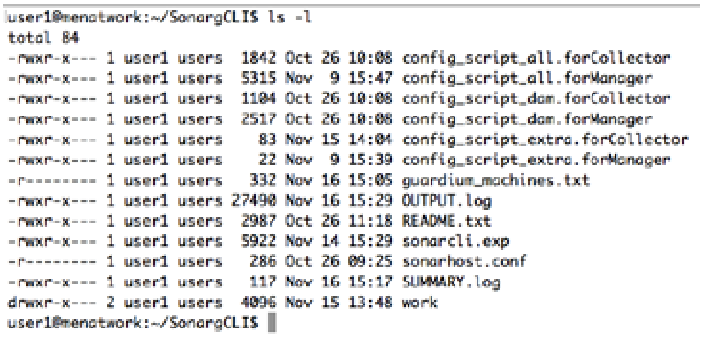


Figure 1 - High-level sonarcli flow

1.2 SCRIPT COMPONENT DETAILS

Brief descriptions of the files needed to successfully execute sonarcli are listed below and they must be contained in the same file directory, as shown in Figure 2.
It should be noted that sonarcli can also be used to execute a variety of tasks on the appliances, above and beyond the SonarG configuration processes described within this document, since it provides a flexible framework for automatically executing any grdapi/cli commands. Also note that several of the files contain password details and thus should have more restrictive permissions to protect this information.



```
user1@menotwork:~/SonargCLI$ ls -l
total 84
-rwxr-x--- 1 user1 users  1842 Oct 26 10:08 config_script_all.forCollector
-rwxr-x--- 1 user1 users  5315 Nov  9 15:47 config_script_all.forManager
-rwxr-x--- 1 user1 users  1104 Oct 26 10:08 config_script_dam.forCollector
-rwxr-x--- 1 user1 users  2517 Oct 26 10:08 config_script_dam.forManager
-rwxr-x--- 1 user1 users    83 Nov 15 14:04 config_script_extra.forCollector
-rwxr-x--- 1 user1 users    22 Nov  9 15:39 config_script_extra.forManager
-r-------- 1 user1 users   332 Nov 16 15:05 guardium_machines.txt
-rwxr-x--- 1 user1 users 27490 Nov 16 15:29 OUTPUT.log
-rwxr-x--- 1 user1 users  2987 Oct 26 11:18 README.txt
-rwxr-x--- 1 user1 users  5922 Nov 14 15:29 sonarcli.exp
-r-------- 1 user1 users   286 Oct 26 09:25 sonarhost.conf
-rwxr-x--- 1 user1 users   117 Nov 16 15:17 SUMMARY.log
drwxr-x--- 2 user1 users  4096 Nov 15 13:48 work
user1@menotwork:~/SonargCLI$ 
```

Figure 2 -sonarcli directory and files

sonarcli.exp – main executable for automatically executing cli/ grdapi commands

config_script_XXX.forManager – grdapi commands to be executed on each central manager communicating with SonarG.

Config_script_XXX.forCollector – grdapi commands to be executed on each collector communicating with SonarG

guardium_machines.txt – listing of Guardium appliances to receive commands

sonarhost.conf – includes details on the SonarG host machine to enable SCP transfers

config_script_extra.forCollector – Additional commands to be executed on collectors outside of the SonarG configuration process

config_script_extra.forManager - Additional commands to be executed on Central Manager outside of the SonarG configuration process

OUTPUT.log – log of the activity for each Guardium appliance cli session generated by sonarcli

SUMMARY.log – Summarized results from the sonarcli execution confirming both successful execution and any exceptions.

README.txt – brief description of sonarcli, its components and execution

1.2.1 SONARCLI.EXP

This is the main executable for configuring the Guardium appliances. Written using the Expect language (https://en.wikipedia.org/wiki/Expect), sonarcli.exp at a high level executes a defined set of grdapi commands against a customer-defined listing of Guardium appliances.

The sonarcli script is easily invoked as expect sonarcli.exp mode, with the mode options outlined below:

Expect sonarcli.exp DAM - Will execute the scripts necessary to configure SonarG for capturing DAM data as defined in config_script_DAM.forManager and config_script_DAM.forCollector. This will also execute any commands listed in config_script_extra.forCollector and config_script_extra.forManager.

Expect sonarcli.exp ALL - Will execute the scripts necessary to configure SonarG for capturing ALL data as defined in config_script_ALL.forManager and config_script_ALL.forCollector. This will also execute any commands listed in config_script_extra.forCollector and config_script_extra.forManager.

Expect sonarcli.exp EXTRA - This will only execute the commands listed in config_script_extra.forCollector and config_script_extra.forManager. This option can be used as a general-purpose framework for executing grdapi commands on any appliances reference in the Guardium_machines.txt file independent of the SonarG configuration.

The sonarcli script requires the "expect" and "tcl" packages to be installed on the Linux machine that will communicate with the appliances. If they are not installed they are readily available as follows:

RedHat:

```
Yum install expect
Yum install tcl
```

Ubuntu:

```
apt-get install expect
apt-get install tcl
```

1.2.2 CONFIG_SCRIPT_XXX.FORMANAGER

This file contains the grdapi commands to be executed on each of the central managers in the environment. There are two types of grdapi commands in this file: commands to configure datamarts and commands to schedule their execution. Examples of the two command types are shown below. All of the fields within each of these commands will be populated for you automatically by sonarcli.

```
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:"Group Members" transferMethod="SCP"
```

Figure 3 - grdapi for datamart SCP configuration

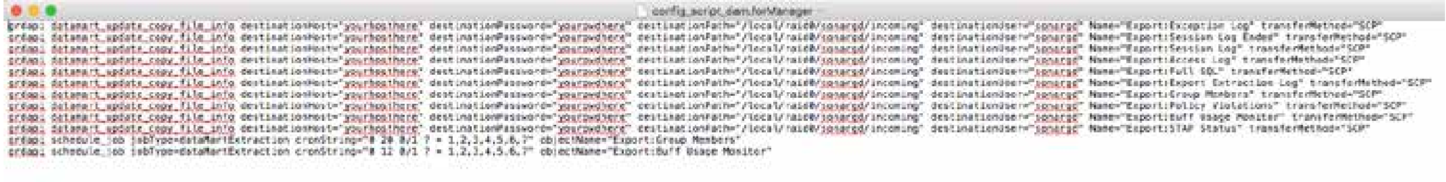
```
grdapi schedule_job jobType=dataMartExtraction cronString="0 20 0/1 ? * 1,2,3,4,5,6,7" objectName=" Group Members "
```

Figure 4 - grdapi for datamart scheduling

Note that there are two different versions for this file type as follows:

config_script_dam.forManager – the standard version used to extract DAM data, shown in Figure 5.

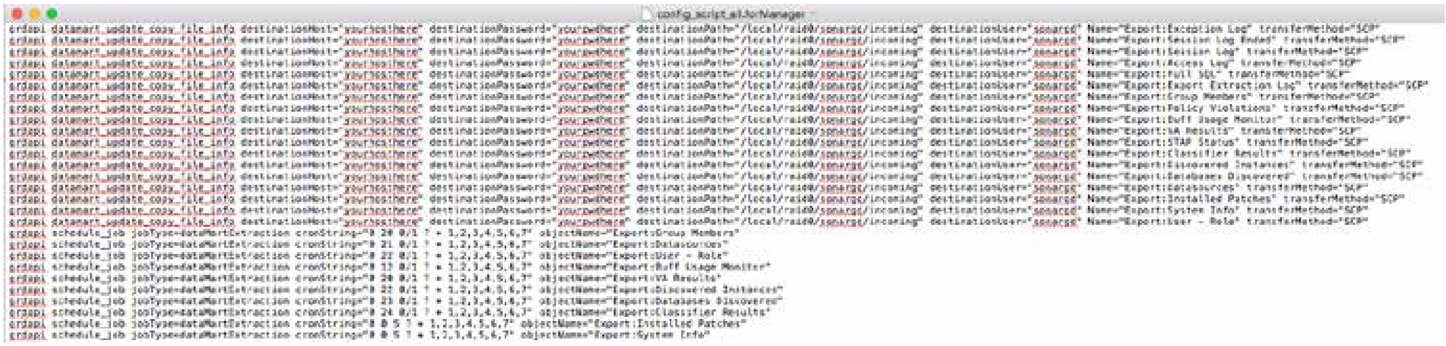
config_script_all.forManager – an expanded version used to extract DAM data along with VA, Discovery and Classifier results, shown in Figure 6.



config_script_dam.forManager

```
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Exception Log" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Session Log Ended" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Session Log" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Access Log" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Full SQL" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Export Extraction Log" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Group Members" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Policy Violations" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Buff Usage Monitor" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:STAP Status" transferMethod="SCP"
grdapi schedule_job jobType=dataMartExtraction cronString="0 20 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Group Members"
grdapi schedule_job jobType=dataMartExtraction cronString="0 12 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Buff Usage Monitor"
```

Figure 5 – config_script_dam.forManager commands



config_script_all.forManager

```
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Exception Log" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Session Log Ended" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Session Log" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Access Log" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Full SQL" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Export Extraction Log" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Group Members" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Policy Violations" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Buff Usage Monitor" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:VA Results" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:STAP Status" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Classifier Results" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Discovered Instances" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Databases Discovered" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Datasources" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:Installed Patches" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:System Info" transferMethod="SCP"
grdapi datamart_update_copy_file_info destinationHost="yourhosthere" destinationPassword="yourpwdhere" destinationPath="/local/raid0/sonargd/incoming" destinationUser="sonargd" Name="Export:User - Role" transferMethod="SCP"
grdapi schedule_job jobType=dataMartExtraction cronString="0 20 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Group Members"
grdapi schedule_job jobType=dataMartExtraction cronString="0 21 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Datasources"
grdapi schedule_job jobType=dataMartExtraction cronString="0 22 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:User - Role"
grdapi schedule_job jobType=dataMartExtraction cronString="0 12 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Buff Usage Monitor"
grdapi schedule_job jobType=dataMartExtraction cronString="0 20 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:VA Results"
grdapi schedule_job jobType=dataMartExtraction cronString="0 22 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Discovered Instances"
grdapi schedule_job jobType=dataMartExtraction cronString="0 23 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Databases Discovered"
grdapi schedule_job jobType=dataMartExtraction cronString="0 24 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Classifier Results"
grdapi schedule_job jobType=dataMartExtraction cronString="0 0 5 ? * 1,2,3,4,5,6,7" objectName="Export:Installed Patches"
grdapi schedule_job jobType=dataMartExtraction cronString="0 0 5 ? * 1,2,3,4,5,6,7" objectName="Export:System Info"
```

Figure 6 - config_script_all.forManager commands

1.2.3 CONFIG_SCRIPT_XXX.FORCOLLECTOR

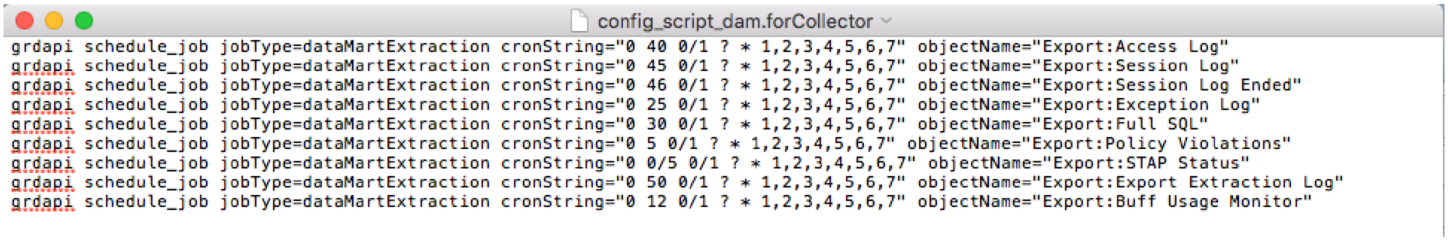
This file contains the grdapi commands to be executed on each of the collectors in the environment in order to schedule the recurring extraction as defined in the cronString. This command structure is identical to the one in Figure 4 - grdapi for datamart scheduling, only modified to schedule the delivery of the Access Log instead of Group Members.

```
grdapi schedule_job jobType=dataMartExtraction cronString="0 40 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Access Log"
```

Note that there are two different versions for this file type as follows:

config_script_dam.forCollector – the standard version used to extract DAM data, shown in Figure 7.

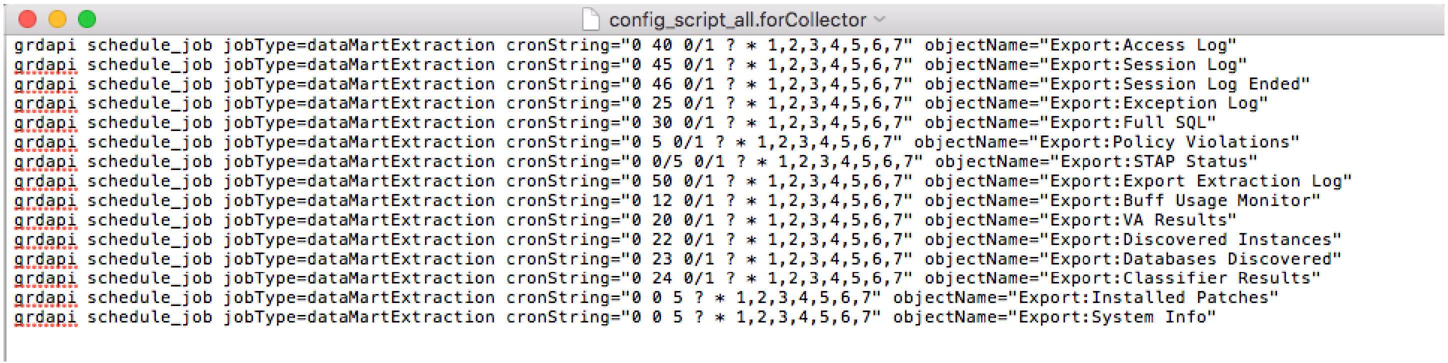
config_script_all.forCollector – an expanded version used to extract DAM data along with VA, Discovery and Classifier results, shown in Figure 8.



config_script_dam.forCollector

```
grdapi schedule_job jobType=dataMartExtraction cronString="0 40 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Access Log"
grdapi schedule_job jobType=dataMartExtraction cronString="0 45 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Session Log"
grdapi schedule_job jobType=dataMartExtraction cronString="0 46 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Session Log Ended"
grdapi schedule_job jobType=dataMartExtraction cronString="0 25 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Exception Log"
grdapi schedule_job jobType=dataMartExtraction cronString="0 30 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Full SQL"
grdapi schedule_job jobType=dataMartExtraction cronString="0 5 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Policy Violations"
grdapi schedule_job jobType=dataMartExtraction cronString="0 0/5 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:STAP Status"
grdapi schedule_job jobType=dataMartExtraction cronString="0 50 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Export Extraction Log"
grdapi schedule_job jobType=dataMartExtraction cronString="0 12 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Buff Usage Monitor"
```

Figure 7 - config_script_dam.forCollector commands



```
grdapi schedule_job jobType=dataMartExtraction cronString="0 40 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Access Log"
grdapi schedule_job jobType=dataMartExtraction cronString="0 45 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Session Log"
grdapi schedule_job jobType=dataMartExtraction cronString="0 46 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Session Log Ended"
grdapi schedule_job jobType=dataMartExtraction cronString="0 25 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Exception Log"
grdapi schedule_job jobType=dataMartExtraction cronString="0 30 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Full SQL"
grdapi schedule_job jobType=dataMartExtraction cronString="0 5 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Policy Violations"
grdapi schedule_job jobType=dataMartExtraction cronString="0 0/5 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:STAP Status"
grdapi schedule_job jobType=dataMartExtraction cronString="0 50 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Export Extraction Log"
grdapi schedule_job jobType=dataMartExtraction cronString="0 12 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Buff Usage Monitor"
grdapi schedule_job jobType=dataMartExtraction cronString="0 20 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:VA Results"
grdapi schedule_job jobType=dataMartExtraction cronString="0 22 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Discovered Instances"
grdapi schedule_job jobType=dataMartExtraction cronString="0 23 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Databases Discovered"
grdapi schedule_job jobType=dataMartExtraction cronString="0 24 0/1 ? * 1,2,3,4,5,6,7" objectName="Export:Classifier Results"
grdapi schedule_job jobType=dataMartExtraction cronString="0 0 5 ? * 1,2,3,4,5,6,7" objectName="Export:Installed Patches"
grdapi schedule_job jobType=dataMartExtraction cronString="0 0 5 ? * 1,2,3,4,5,6,7" objectName="Export:System Info"
```

Figure 8 - config_script_all.forCollector commands

1.2.4 GUARDIUM_MACHINES.TXT

This file provides a simple listing of the Guardium appliances within your environment that will receive commands from the sonarcli script as shown in Figure 9. Although the format allows for communicating with aggregators, in most SonarG deployments there will not be any aggregators. The template can use either IP address or hostname to identify the appliance.

```
# List of Guardium machines:
#
#  Format:  <type>:<Guardium IP/hostname>:<cli password>
#  where type is:  C-Collector, A-Aggregator, M-Manager
#
#  Comment out guardium hosts you want to skip using #
#  #C:192.168.0.123:Gu@rd1um
#
M:192.168.0.110:Gu@rd1um
C:192.168.0.115:Gu@rd1um
C:192.168.0.116:Gu@rd1um
#C:collector2.corp.com:Gu@rd1um
```

Figure 9 - guardium_machines.txt format

1.2.5 SONARHOST.CONF

This file specifies the details for the SonarG machine that will receive the SCP transfers from the Guardium appliances, as shown in Figure 10. These parameters will be automatically inserted into the grdapi commands that sonarcli executes on the appliances. The user "sonargd" is a specific user defined within the SonarG system to receive incoming data.

Note that this file should be managed with stringent permissions since it contains password details.

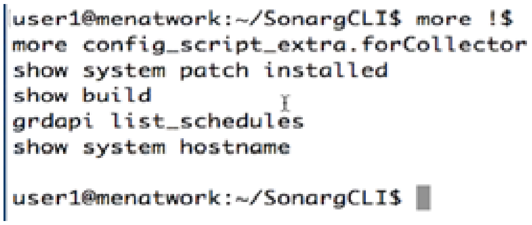
```
# Configure the values for the SonarG machine
#
#  SONARG_HOST  - IP/Hostname for the SonarG server
#  SONARG_PWD   - Password for the sonargd user
#  SONARG_PATH  - Target path for the Datamart dumps
#
SONARG_HOST=192.168.0.90
SONARG_PWD=password
SONARG_PATH=/var/lib/sonargd/incoming
```

Figure 10 - sonarhost.conf

1.2.6 CONFIG_SCRIPT_EXTRA.FORXX

Sonarcli can be used as a general-purpose mechanism for executing any grdapi commands on any of the appliances specified in the Guardium_machines.txt file. From a central point of control you can easily direct a variety of commands at the various appliances in an efficient and automated manner.

There are two files used to specific the commands to be executed: config_script_extra.forCollector and config_script_extra.forManager and the format for each of these files is very simple, as shown in Figure 11, which shows a sample file for a Collector.



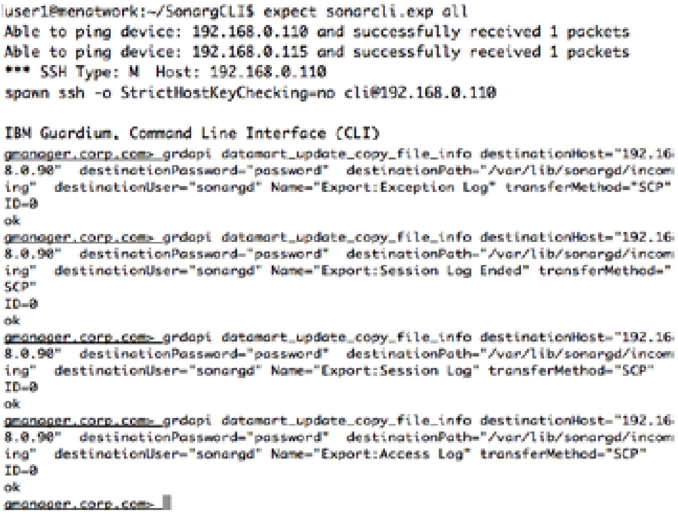
```
user1@menatwork:~/SonargCLI$ more !$
more config_script_extra.forCollector
show system patch installed
show build
grdapi list_schedules
show system hostname

user1@menatwork:~/SonargCLI$ 
```

Figure 11 - Sample config_script_extra.forCollector file

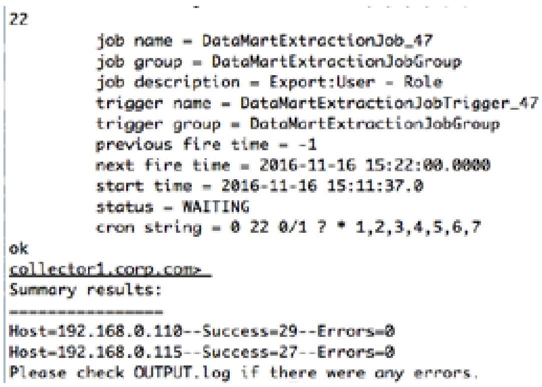
1.2.7 OUTPUT.LOG

Output of the cli activity associated with the sonarcli execution. An example series of extracts is shown in Figure 12. For each grdapi command the ID=0 and OK messages confirm success.



```
user1@menatwork:~/SonargCLI$ expect sonarcli.exp all
Able to ping device: 192.168.0.110 and successfully received 1 packets
Able to ping device: 192.168.0.115 and successfully received 1 packets
*** SSH Type: M  Host: 192.168.0.110
spawn ssh -o StrictHostKeyChecking=no cli@192.168.0.110

IBM Guardium, Command Line Interface (CLI)
gmanager.corp.com> grdapi datamart_update_copy_file_info destinationHost="192.16
8.0.90"  destinationPassword="password"  destinationPath="/var/lib/sonargd/incom
ing"  destinationUser="sonargd" Name="Export:Exception Log" transferMethod="SCP"
ID=0
ok
gmanager.corp.com> grdapi datamart_update_copy_file_info destinationHost="192.16
8.0.90"  destinationPassword="password"  destinationPath="/var/lib/sonargd/incom
ing"  destinationUser="sonargd" Name="Export:Session Log Ended" transferMethod="
SCP"
ID=0
ok
gmanager.corp.com> grdapi datamart_update_copy_file_info destinationHost="192.16
8.0.90"  destinationPassword="password"  destinationPath="/var/lib/sonargd/incom
ing"  destinationUser="sonargd" Name="Export:Session Log" transferMethod="SCP"
ID=0
ok
gmanager.corp.com> grdapi datamart_update_copy_file_info destinationHost="192.16
8.0.90"  destinationPassword="password"  destinationPath="/var/lib/sonargd/incom
ing"  destinationUser="sonargd" Name="Export:Access Log" transferMethod="SCP"
ID=0
ok
gmanager.corp.com> 
```

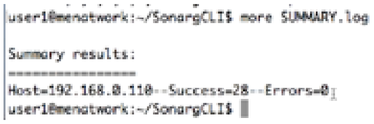


```
22
        job name = DataMartExtractionJob_47
        job group = DataMartExtractionJobGroup
        job description = Export:User - Role
        trigger name = DataMartExtractionJobTrigger_47
        trigger group = DataMartExtractionJobGroup
        previous fire time = -1
        next fire time = 2016-11-16 15:22:00.0000
        start time = 2016-11-16 15:11:37.0
        status = WAITING
        cron string = 0 22 0/1 ? * 1,2,3,4,5,6,7
ok
collector1.corp.com>
Summary results:
================
Host=192.168.0.110--Success=29--Errors=0
Host=192.168.0.115--Success=27--Errors=0
Please check OUTPUT.log if there were any errors.
```

Figure 12 - OUTPUT.log Example

1.2.8 SUMMARY.LOG

Details from each appliance on the successful completion of grdapi commands and also any exceptions.



```
user1@menatwork:~/SonargCLI$ more SUMMARY.log

Summary results:
================
Host=192.168.0.110--Success=28--Errors=0
user1@menatwork:~/SonargCLI$
```

Figure 13 - SUMMARY.log Example

1.2.9 README.TXT

A condensed version of this more detailed explanation.

1.3 DIRECT VS. STAGING METHOD FOR EXTRACT DELIVERY

The standard configuration of SonarG will have Guardium appliances "pushing" csv files via SCP directly to a defined directory on the SonarG machine. During the POC process the customer will often wish to enable data extracts in advance of the installation of the SonarG application in order to validate this process and to create a richer data set for the POC. This is easily accomplished by modifying the sonarhost.conf file to point to a STAGING server where the data extract files being generated can be stored prior to delivery to the SonarG host.

```
# Configure the values for the STAGING server
#
#  SONARG_HOST  - IP/Hostname for the STAGING server
#  SONARG_PWD   - Password for the user on the STAGING server
#  SONARG_PATH  - Target path for the Datamart dumps on the STAGING server
#
SONARG_HOST=192.168.0.90
SONARG_PWD=password
SONARG_PATH=/any path
```

Note that the STAGING server must be able to execute SCP transfers.

Once the SonarG machine is installed and the application is running, the data extract files will be moved into the /var/lib/sonargd/incoming and automatically processed into the SonarG data store. The sonarhost.conf file will then be modified to define a direct connection between the Guardium appliances and SonarG and the sonarcli.exp script re-executed in order to reconfigure the transfers.

2 DATAMARTS

The SonarG team works closely with IBM to develop and maintain a pre-defined library of datamarts that enable the transfer of commonly requested data between the two systems. This includes DAM, policy violations, sessions, STAPs and many other data types as described in detail below. IBM has embedded the code necessary to activate these datamarts directly into v10.1 and for v9.5 the complete set of datamarts are available for patch GPU700 or later. A subset of the pre-defined datamart library is also available for versions 8.2 and 9.5 with the appropriate patches installed.

Note that the SonarG system is capable of receiving data from all 3 different major Guardium versions concurrently and will automatically reconcile these into a single, unified view within the SonarG system. This includes spanning multiple CM environments, thus yielding a true enterprise view of the entire DB Security landscape.

2.1 PRE-DEFINED DATAMARTS

Listed below are the pre-defined datamarts jointly developed by IBM and SonarG, along with information regarding their content and delivery frequency. This library will continue to evolve based on customer feedback and as noted in section 2.2 there is also the option for custom datamarts to be added to address unique customer requirements.

The reference to DAM or ALL modes is referring to the depth of data that SonarG will receive. DAM mode is tuned for those customers strictly using Guardium for DAM activity whereas ALL mode collects all data from the Guardium system including VA, Discovery and Classifier data from the pre-defined datamarts.

2.1.1 EXCEPTION LOG

Details from the Guardium Exception domain are delivered on an hourly basis.
Appears in both DAM and ALL modes.

2.1.2 SESSION LOG ENDED

End of Session details are delivered on an hourly basis.
Appears in both DAM and ALL modes.

2.1.3 SESSION LOG

Details from the Sessions Log are delivered on an hourly basis.
Appears in both DAM and ALL modes.

2.1.4 ACCESS LOG

Details regarding the database queries are delivered on an hourly basis.
Appears in both DAM and ALL modes.

2.1.5 ACCESS LOG – DETAILED

This is an optional version of the Access Log datamart expanded to include Application Event ID information. This is also delivered on an hourly basis.
Appears in both DAM and ALL modes.

2.1.6 FULL SQL

Details from the Guardium Full SQL domain are transferred on an hourly basis.
Appears in both DAM and ALL modes.

2.1.7 EXPORT EXTRACTION LOG

Details from the Datamart Extraction Logs on the Guardium appliances are delivered on an hourly basis.
Appears in both DAM and ALL modes.

2.1.8 GROUP MEMBERS

SonarG remains synchronized with the Guardium group member information and this information is fully accessible for reports and queries. This information is delivered to SonarG hourly.
Appears in both DAM and ALL modes.

2.1.9 POLICY VIOLATIONS

Information regarding Policy Violations is delivered on an hourly basis.
Appears in both DAM and ALL modes.

2.1.10 BUFF USAGE MONITOR

Operational information regarding the health and capacity utilization of the collectors is passed on a daily basis to SonarG. This includes items such as Snif restart rates, Guessed rates, Average Daily loads, etc.
Appears in both DAM and ALL modes.

2.1.11 VA RESULTS

The results of a Vulnerability Assessment scan will be delivered to SonarG on a daily basis for any new results.
Appears in ALL mode only.

2.1.12 STAP STATUS

STAP status details are delivered every 5 minutes in order to improve the ability to identify and respond to STAP outages.
Appears in both DAM and ALL modes.

2.1.13 CLASSIFIER RESULTS

The results of a Classifier scan will be delivered to SonarG on a daily basis for any new results.
Appears in ALL mode only.

2.1.14 DISCOVERED INSTANCES

The specific instances discovered during an STAP-based Discovery scan will be delivered to SonarG on a daily basis for any new results.
Appears in ALL mode only.

2.1.15 DATABASES DISCOVERED

The results of a Database Discovery scan will be delivered to SonarG on a daily basis for any new results.
Appears in ALL mode only.

2.1.16 DATASOURCES

Information regarding datasources defined and managed within Guardium is shared with SonarG in order to facilitate additional automation such as CMDB reconciliation, enrichment with application owner data, password management, etc. This information is delivered on a daily basis.

There is the option of using SonarG to automatically populate the datasource definitions for Guardium, but details of that are beyond the scope of this document.
Appears in ALL mode only.

2.1.17 INSTALLED PATCHES

Details on the currently installed patches for all appliances linked to SonarG are delivered on a daily basis.
Appears in ALL mode only.

2.1.18 SYSTEM INFO

A rich set of system information is delivered to SonarG on a daily basis. This includes details on all Guardium appliances installed, unit type, purge schedule, etc.
Appears in ALL mode only.

2.2 CUSTOM DATAMARTS

In the case where you need to collect data that is available within Guardium but not captured in the standard datamarts, there is the option to incorporate custom datamarts for expanded data access. Any report that is executed on the Guardium collector or central manager can be converted into a datamart and its results piped directly into SonarG using the standard data transfer process. Shown below in Figure 14 you can see the option for converting a query into a datamart from within the query builder.

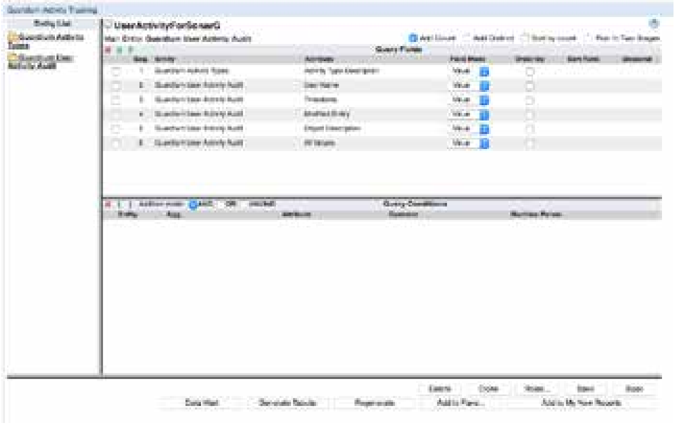
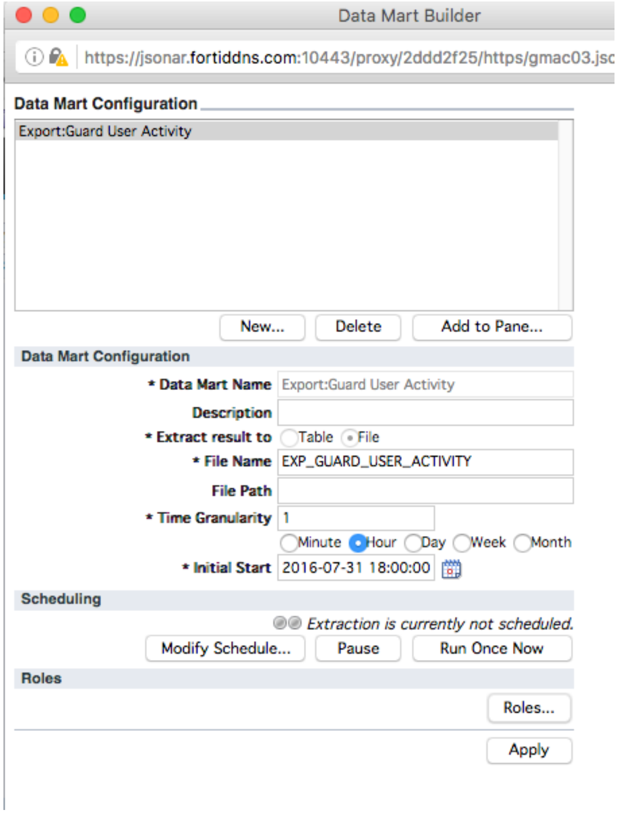


Figure 14 - Converting Query to Datamart

Figure 15 shows the scheduling of the datamart internal to the Guardium appliance, in this case defining a filename with the prefix EXP that is created on an hourly basis. The EXP prefix informs the appliance that this datamart is being created for delivery to the SonarG application. The datamart name must begin with EXPORT and the EXP prefix must appear at the start of the filename in order for the transfer to SonarG complete successfully.



Data Mart Builder

https://jsonar.fortiddns.com:10443/proxy/2ddd2f25/https/gmac03.jsc

Data Mart Configuration

Export:Guard User Activity

Data Mart Configuration

* Data Mart Name: Export:Guard User Activity

Description:

* Extract result to: ○ Table ● File

* File Name: EXP_GUARD_USER_ACTIVITY

File Path:

* Time Granularity: 1

○ Minute ● Hour ○ Day ○ Week ○ Month

* Initial Start: 2016-07-31 18:00:00

Scheduling

Extraction is currently not scheduled.

Roles

Figure 15 - Schedule datamart delivery within Guardium

As with the standard datamarts, grdapi commands must be executed in order to configure the SCP transfer of the file to SonarG and also to schedule this transfer on an hourly basis.

A grdapi command must be executed in order to instruct Guardium to enable headers to be created along with the datamart output, as shown in Figure 16.

```
grdapi datamart_include_file_header
Name="Export:Databases Discovered"
includeFileHeader="Yes"
```

Figure 16 - grdapi to enable Headers

The standard grdapi command for the SCP transfer configuration is shown in Figure 17 and the standard grdapi command for scheduling the hourly transfer from the Guardium appliance to SonarG is shown in Figure 18. This process is providing the Guardium appliance with the target hostname, username and password for the machine that will receive the SCP transfer, along with path details on where to put the file. The second step schedules a CRON job for the recurring delivery of the datamart extract.

```
grdapi datamart_update_copy_file_
info destinationHost="yourhosthere"
destinationPassword="yourpwdhere" destinationPath="/
local/raid0/sonargd/incoming" destinationUser="sonargd"
Name="Export:GUARD_USER_ACTIVITY"
transferMethod="SCP"
```

Figure 17 - grdapi for datamart SCP configuration

```
grdapi schedule_job jobType=dataMartExtraction
cronString="0 40 0/1 ? * 1,2,3,4,5,6,7"
objectName="Export:USER_GUARD_ACTIVITY"
```

Figure 18 - grdapi for datamart transfer schedule

Note that all of these steps are a one-time process executed during the configuration of the datamart extract and only need to be re-executed should any of the configuration parameters change.