Task report for Task Authorization 1 for: Technology demonstration of the Joint Network Defence and Management System (JNDMS) Project

Prepared by:
MacDonald Dettwiler and Associates Ltd.
Suite 60, 1000 Windmill Rd.
Dartmouth, NS B3B 1L7

PWGSC Contract Number:
W7714-040875/001/SV
CDRL/DID SD-005

Contract Scientific Authority:
Marc Gregoire, Project Manager (613-998-2073)

The scientific or technical validity of this Contract Report is entirely the responsibility of the Contractor and the contents do not necessarily have the approval or endorsement of the Department of National Defence of Canada.

Contract Report Number DRDC-RDDC-2014-C93

DN0935: 03 DECEMBER 2008
ISSUE 1/1: 30 JANUARY 2009

TASK REPORT FOR TASK AUTHORIZATION 1

FOR

TECHNOLOGY DEMONSTRATION OF THE JOINT NETWORK DEFENCE AND MANAGEMENT SYSTEM (JNDMS) PROJECT

CONTRACT NO. W7714-040875/001/SV

CDRL/DID SD-005

PREPARED FOR:

DEFENCE R&D CANADA - OTTAWA
3701 CARLING AVENUE
OTTAWA ON K1A 0Z4

PREPARED BY:

MACDONALD DETTWILER AND ASSOCIATES LTD.
SUITE 60, 1000 WINDMILL RD.
DARTMOUTH, NS B3B 1L7

DOCUMENT APPROVAL SHEET

TASK REPORT FOR TASK AUTHORIZATION 1

FOR

TECHNOLOGY DEMONSTRATION OF THE JOINT NETWORK DEFENCE AND MANAGEMENT SYSTEM (JNDMS) PROJECT

CONTRACT NO. W7714-040875/001/SV

CDRL /DID SD-005

MACDONALD DETTWILER AND ASSOCIATES LTD.

Scott MacDonald		
Author	(Signature)	(Date)
Beverly MacNeil		
Quality Assurance	(Signature)	(Date)
Brett Trask		
Project Manager	(Signature)	(Date)

CHANGE RECORD

Rev. #	Pages Affected	Description	Soft Copy Y-N	Date of Issue
1/0	All	First Issue	Y	03 Dec 08
1/1	All	First Issue (Amended)	Y	30 Jan 09

TABLE OF CONTENTS

LIST OF FIGURES

.

LIST OF TABLES

1 Introduction

1.1 Overview

The purpose of this document is to report on the findings and activity for Task Authorization #1 of the Joint Network Defence and Management System (JNDMS) Technology Demonstrator Project. This document specifies DID SD-005 and covers CDRLs 28, 29 and 30.

1.2 Referenced Documents

The following documents are referenced in this report:

1. Project Management Plan for the Technology Demonstration of the Joint Network Defence and Management System (JNDMS) Project, DN0648, Issue 3/0.
2. JNDMS DREnet Deployment Report, DN0916, Issue 1/0.
3. JNDMS Amendment Task Authorization 2 Proposal, Proposal No. 01-5151, Issue 1/1.
4. JNDMS Detailed Design Document DN0678, Issue 3/1.

2 Task Activity and Findings

This section outlines the activity and results of the five tasks within Task Authorization #1.

2.1 Project Management

This task included the planning and oversight for this task authorization. The plan for this task authorization was captured in an update to the Project Management Plan [R1] and monthly reports provided status updates on progress.

The purchase of nCircle IP360 components, including the VnE appliance, the device profiler, the Security Intelligence Hub and associated licenses was originally planned as part of this task authorization. During the execution of this project, however, it was noted that the current DND contract provided a much better procurement vehicle for the licenses as they would be able to procure several thousand licenses where we had only planned on 100 licenses. The IP360 hardware was therefore purchased under this contract while the IP360 licenses for the vulnerability manager and security intelligence hub were to be GFE for any follow on task authorizations. The funds originally allocated for the licenses were reallocated to the priority tasks (see ‘Scalability Study’ below).

The task for the update of the GIS components was reviewed as part of this task authorization and it was noted that alternatives to the current components existed. A brief report was drafted that provided the details of using the GIS task to upgrade the components, instead of concentrating on adding layers to the existing system. The report and the results can be found in Section 2.5 (GIS).

2.2 Scalability Study

The goal of the scalability study task was to assess key performance issues within JNDMS and to determine potential corrective action. As part of this task there were a number of days set aside to specifically address the high priority issues.

This task consisted of updates to the simulated environment, instrumentation and tracking of the system, an assessment of issues and finally effort towards correcting the issues.

2.2.1 Simulated Environment

The simulated environment was updated to support larger data sets. The primary focus was on increasing the number of assets. Much of the processing within JNDMS is asset centric and, as such, the number of assets is likely to cause issues with processing as well as display.

The following table shows some of the primary indicators for some of the preset datasets used. The first data set shows the default data set used for building and small demonstrations. The second data set shows the extended dataset that was built and expanded upon for cycle 3 development. The last dataset shows the primary dataset used for scalability testing. Some individual tests would modify the base dataset to examine potential impacts.

Table 1: Simulated dataset sizes.

Entity	Default	Extended	Scalability
Assets (total)	605	3,000	16,400
Host Assets	88	1,200	5,200
Operations	4	4	5
Zones	15	15	15
Locations	30	30	30
Vulnerability Definitions	10	100	30,000

To support the testing of different configurations and to help build larger and larger datasets a small tool, called the Subnet Creator, was written to help create assets in different subnets (see section 2.2.1.1 below).

2.2.1.1 Subnet Creator Tool

The Subnet creator is a tool written in Java to support the automated creation of simulated subnets. It can be run giving it a subnet, the number of hosts to create, the link from the new subnet to the network and a profile of software that should be installed on new hosts. This tool was used to scale the simulated network and can also be used to create subnets with specific profiles.

Subnet Creator command line:

```
> java –jar SubnetCreator.jar –j [path to client] –e [JSS] –s [template] –oh [hardware]
–os [software] –nh [number] –r [subnet]
```

Description of options:

1. Path to client: This is the path to the JSS client JAR (jss_client.jar).

2. JSS: This is the URL of the JSS endpoint that will be used to submit the generated assets.

3. Template: This is a file that describes what software assets will be created on the newly created hosts. This is an XML file that conforms to the JNDMS XML schema (see the JNDMS Design Document, rev 3.1, section 3.6.2.3.1) to define assets.

   The following is an example of a software template that will create ‘win2k’ (Windows 2000) on all newly created workstations.

```
<?xml version="1.0"?>
<jndms xmlns="http://jndms"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://jndms jndms.xsd">
  <assets>
    <!—create placeholder reference →
    <existingAssets>
      <assetRef>
        <id>workstations</id>
        <ip_range_data/>
      </assetRef>
    </existingAssets>
    <!—list of assets and products to create on workstations →
    <asset create_on_all="true">
      <method>software</method>
      <name>win2k</name>
      <host>
        <id>workstations</id>
      </host>
      <product create="true">
        <name>Windows 2000</name>
        <vendor>Microsoft</vendor>
        <version>Professional SP2</version>
      </product>
    </asset>
  </assets>
</jndms>
```

4. Hardware: This is the output file that will store the list of newly created hardware assets. The file that is created can be submitted directly to JNDMS using the following command:

   > java –jar jss_client.jar com.mdacorporation.jndms.JSS.Client.JSSBatchClient [file]

5. Software: This is the output file that will store the list of newly create software assets. This file can be submitted directly to JNDMS using the following command:

6. > java -jar jss_client.jar op=jndms_xml source=subnetcreator url=file:[file] endpoint=[jss url]

7. Number: This is the number of hosts to create. If this is not given the entire subnet (default of 90.1.1.0/24) will be created.

8. Subnet: This is the subnet mask to use. For example 192.168.3.0/24.

A convenience script is provided in Test/cycle_3/Discovery_scripts called make_subnet.bat that combines the above steps of creation and submission. It can be called in the following manner:

1. Create a given number of assets starting at 90.1.1.0:

   make_subnet.bat -nh <num_assets>

2. Create a full subnet range:

   make_subnet.bat -r <subnet_range(e.g. 90.1.1.0/24)>

3. Create a given number of assets starting at a subnet range:

   make_subnet.bat -nh <num_assets> -r <subnet_range(e.g. 90.1.1.0/24)>

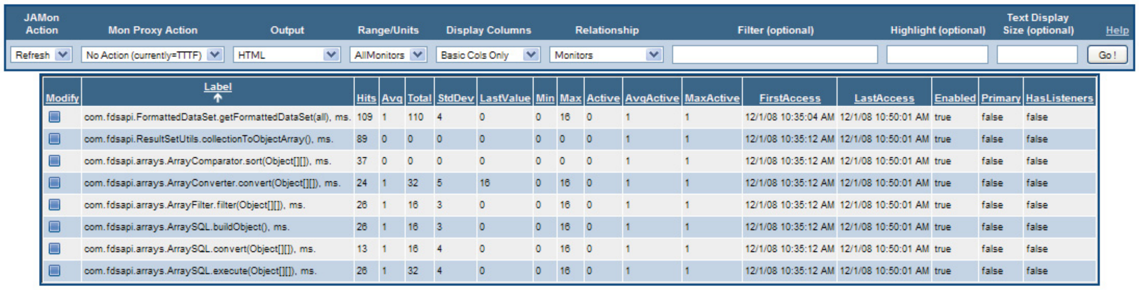
2.2.2 Instrumentation and Tracking

To be able to gain insight into the possible performance bottlenecks, to evaluate changes and to support ongoing tracking and testing, the JNDMS system components were instrumented to collect performance information.

The instrumentation was performed by adding calls or annotations to the code and using JAMon (http://jamonapi.sourceforge.net/) for support. The instrumentation can be done in two ways to ensure a flexible environment. The first is by adding Java Annotations to methods that should be tracked. An annotation of "@PerfMonitor" will cause the given method to be monitored.

The second method is to add specific start and stop calls to a new PerfMon class. This type of annotation allows any block of code to be tracked, however it is more error prone than the first case.

After the code has been annotated there are two methods to collect the performance related information. The first is through a JAMon applet that shows live counters of all annotated sections (see Figure 1). This can be used to quickly collect information and also to examine how a running or live system is performing.



JAMon Action	Mon Proxy Action	Output	Range/Units	Display Columns	Relationship	Filter (optional)	Highlight (optional)	Text Display Size (optional)	Help
Refresh	No Action (currently=TTTF)	HTML	AllMonitors	Basic Cols Only	Monitors				Go !

Modify	Label ↑	Hits	Avg	Total	StdDev	LastValue	Min	Max	Active	AvgActive	MaxActive	FirstAccess	LastAccess	Enabled	Primary	HasListeners
■	com.fdsapi.FormattedDataSet.getFormattedDataSet(all), ms.	109	1	110	4	0	0	16	0	1	1	12/1/08 10:35:04 AM	12/1/08 10:50:01 AM	true	false	false
■	com.fdsapi.ResultSetUtils.collectionToObjectArray(), ms.	89	0	0	0	0	0	0	0	1	1	12/1/08 10:35:12 AM	12/1/08 10:50:01 AM	true	false	false
■	com.fdsapi.arrays.ArrayComparator.sort(Object[][]), ms.	37	0	0	0	0	0	0	0	1	1	12/1/08 10:35:12 AM	12/1/08 10:50:01 AM	true	false	false
■	com.fdsapi.arrays.ArrayConverter.convert(Object[][]), ms.	24	1	32	5	16	0	16	0	1	1	12/1/08 10:35:12 AM	12/1/08 10:50:01 AM	true	false	false
■	com.fdsapi.arrays.ArrayFilter.filter(Object[][]), ms.	26	1	16	3	0	0	16	0	1	1	12/1/08 10:35:12 AM	12/1/08 10:50:01 AM	true	false	false
■	com.fdsapi.arrays.ArraySQL.buildObject(), ms.	26	1	16	3	0	0	16	0	1	1	12/1/08 10:35:12 AM	12/1/08 10:50:01 AM	true	false	false
■	com.fdsapi.arrays.ArraySQL.convert(Object[][]), ms.	13	1	16	4	0	0	16	0	1	1	12/1/08 10:35:12 AM	12/1/08 10:50:01 AM	true	false	false
■	com.fdsapi.arrays.ArraySQL.execute(Object[][]), ms.	26	1	32	4	0	0	16	0	1	1	12/1/08 10:35:12 AM	12/1/08 10:50:01 AM	true	false	false

Figure 1: Live Performance Reporting

Each of the annotation methods will cause performance information to be logged (if performance logging is enabled for the log4j) through Tomcat and log4j. The resulting log files can be analyzed and the information collected into a database. This collected information can be used to examine the various calls.

2.2.3 Automated Build and Test Case Support

The ability to quickly identify when tests fail or if significant performance changes occur is essential to having confidence in the deployed system. To support this effort an automated build tool called Hudson was implemented and the test cases were reviewed and updated. The test cases were set up so that they can be automatically run.

The tests, ideally, should ensure that complete end to end functionality is tested and to support this goal a web test suite called Selenium (http://selenium.seleniumhq.org/) was used. This allows for scripting of interaction through the portal so that portal use can be part of the test cases.

The use of Hudson as a build tool to support continuous builds allows a number of views of the status of the builds, as well as some initial performance information. The first view will list the various builds that have been configured and show its current status. The status will be blue for a complete success (both build and all test cases), yellow for unstable (build was successful however test cases failed) and red for failed (build did not even complete). Hudson can provide a summary of a given profile and show the status of the builds over time as well as how long they took (see Figure 2).

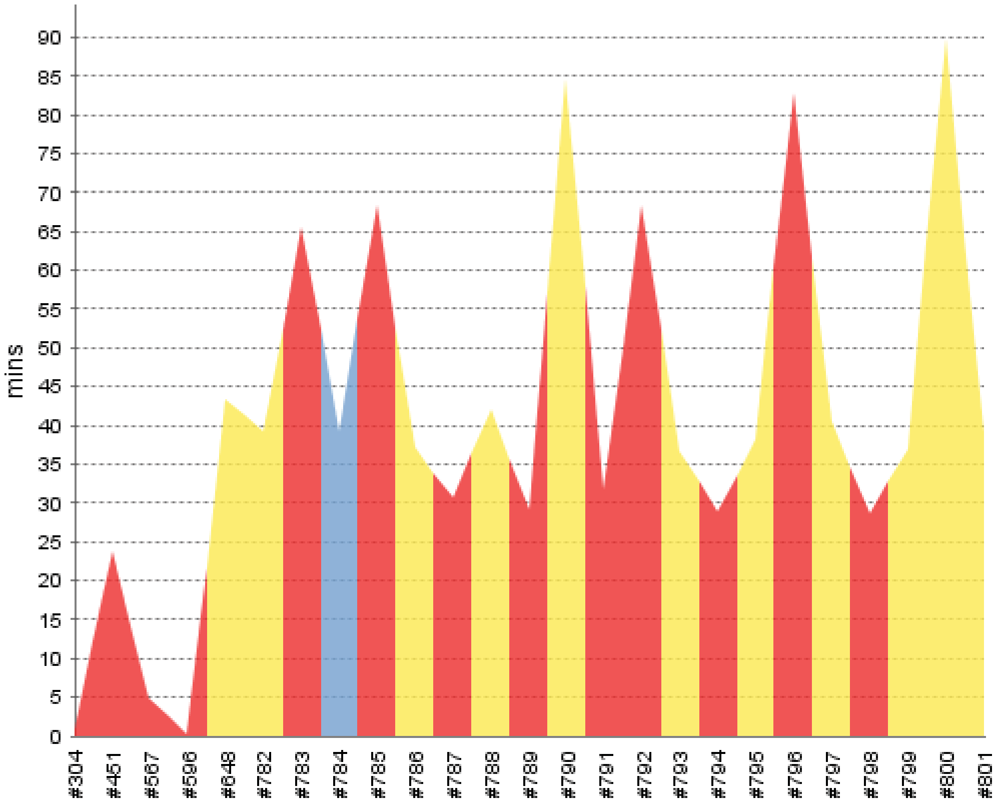


Figure 2: Hudson Build Summary

Hudson can also give a report on the history of the test cases that shows the number of test cases run against each build as well as their status (pass or fail). Figure 3 shows an example of this view.

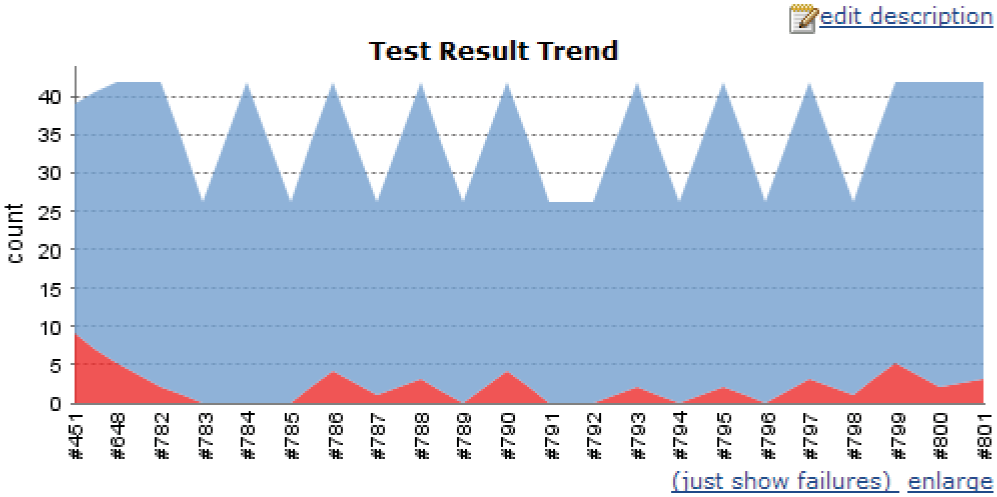
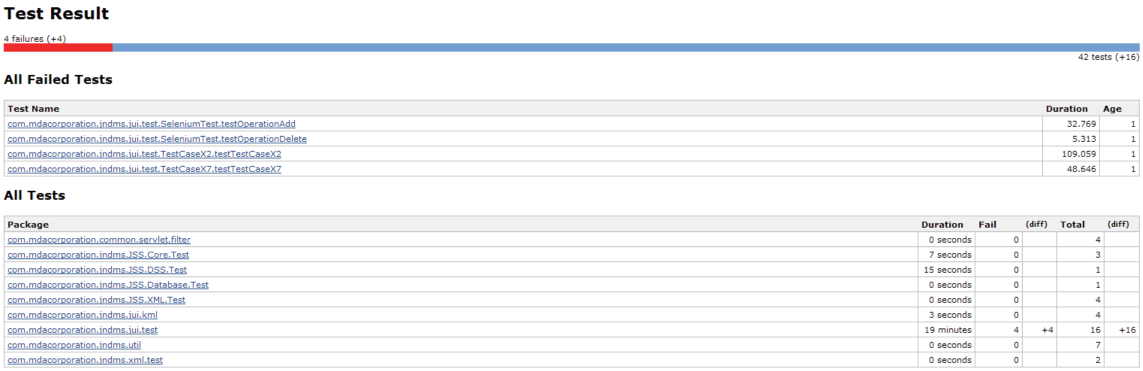


Figure 3: Hudson Test Case History.

Hudson can also identify the details of what test cases were run and how long each of the test cases took to run. This view shows when there are build problems but also gives an initial indication of performance issues.



Test Result

4 failures (+4)

42 tests (+16)

All Failed Tests

Test Name	Duration	Age
com.mdacorporation.jndms.jui.test.SeleniumTest.testOperationAdd	32.769	1
com.mdacorporation.jndms.jui.test.SeleniumTest.testOperationDelete	5.313	1
com.mdacorporation.jndms.jui.test.TestCaseX2.testTestCaseX2	109.059	1
com.mdacorporation.jndms.jui.test.TestCaseX7.testTestCaseX7	48.646	1

All Tests

Package	Duration	Fail	(diff)	Total	(diff)
com.mdacorporation.common.servlet.filter	0 seconds	0		4	
com.mdacorporation.jndms.JSS.Core.Test	7 seconds	0		3	
com.mdacorporation.jndms.JSS.DSS.Test	15 seconds	0		1	
com.mdacorporation.jndms.JSS.Database.Test	0 seconds	0		1	
com.mdacorporation.jndms.JSS.XML.Test	0 seconds	0		4	
com.mdacorporation.jndms.jui.kml	3 seconds	0		4	
com.mdacorporation.jndms.jui.test	19 minutes	4	+4	16	+16
com.mdacorporation.jndms.util	0 seconds	0		7	
com.mdacorporation.jndms.xml.test	0 seconds	0		2	

Figure 4. Hudson Test Result Details

The following table (Table 2) shows the default tests that are run against every build. Hudson will run and report (as shown above) on each of the configured tests.

Table 2: Automated build tests

Package	**Test**	**Note**
Servlet	Filter Test	Test filtering of servlets (used for performance monitoring)
JSS	Asset Report	Submission of software asset report.
	Symantec Vulnerability Report	Example of vulnerability scan
	Database check	Check connections and status of data base.
	Point XML Test	Test handling of geographic points.
	Jaxb test	General XML handling.
DSS Core	Core unit tests	Internal sanity checks.
JUI	KML	Test KML creation.
	Products	Test viewing of products.
	Operation	Test add, delete and edit of operations.
	Test Case 2	Automated version of cycle 3 test case 2
	Extended test cases 1-2,4-8,10	Automated version of cycle 3 extended test cases.
	CVE Import	Vulnerability definition import
	Nessus Import	Vulnerability scan import
	Topology Test	Test the loading and interaction with the topology applet.
Utils	Internals	Additional internal sanity checks.

2.2.4 Scalability Priority Tasks

The initial findings, using all of the above tools, were reviewed and then a set of priority tasks was defined. These tasks identified the initial performance issues that would have to be under taken before deployment. These tasks were reviewed with DRDC and identified as the objective for the sixty engineering days.

Table 3: Scalability Priority Tasks

Pri	Component	Task	Description	Notes	Status
1	DSS	Correlation optimization	The scalability study has shown that one of the bottlenecks in the DSS is the correlation calculations. An initial review has identified the database handling as a key issue (addressed in the 'database optimization task') but also noted that additional effort directly on the correlation calculation can be done. The effort allocated for the 'modularization' task will also likely impact the performance of the correlation. As part of this task the ability to defer the calculation will be looked at. This may involve waiting until a request is made or setting up a timed event. It is, however, expected, that the database and modularization efforts combined with the algorithm review will be sufficient.	See 2.2.6 below	Complete

Pri	Component	Task	Description	Notes	Status
			This task would include the effort to use JGraphT to create an in memory model of the incidents and events. This would allow not only for decreased database access but also for faster processing. The current investigations into this task have noted that our current processing is in the order of O(n^2) with respect to the number of incidents so as the number of incidents grow the problem gets exponentially worse. The algorithm may be reviewed to assess the possibility, for example, of only correlating the 'aggregated' events instead of all events. This would be more beneficial in conjunction with the 'correlation updates' task and the 'ISM optimization' task.		
1	DSS	Modularization	The DSS currently runs in a single thread separate from all other processing. The DSS itself performs several functions, many of which are independent. For example the correlation and risk calculations do not directly depend on each other.	See 2.2.6 below	Core of modularization was completed, additional issues remain (see 2.2.6) Additional tasks part of TA2 (see section 2.4.2)

Pri	Component	Task	Description	Notes	Status
			This task would be to modularize the DSS such that database access, risk calculations, correlation, impact assessment, etc, can all be done in a modular fashion and many of the tasks in parallel. This task would depend on the database optimization task being done so that the in memory models can be leveraged to ease the modularization. Another advantage of this task would be in future development and in risk mitigation. A more modular DSS would ease integrating additional processing or alternate algorithms. For example an alternate impact assessment would be easier to implement on a more modular DSS.		
1	Mapping	GIS Upgrade	This is a separate task under the TA1 GIS task. The mapping tasks assume that this task has been completed.	See Section 2.5	Core upgrade complete, but additional functionality required (See Section 2.5)
1	Portal	Data lists	This task would include: • Change drop downs to ajax updates to improve page load responsiveness • Update list contents via ajax to further improve page load responsiveness.	See Section 2.2.7	Complete

Pri	Component	Task	Description	Notes	Status
			The current scalability study has shown that this is more than just a performance issue; with larger data sets preventing pages from loading all together. Using the filter to limit the amount of data has allowed pages to load.		
1	SIM	ISM optimization	This task would be to leverage more of the Intellitactics ability to process large numbers of events, then to report on the correlated events. This task would include the following: • Expand the event types escalated to JNDMS including aggregated event buckets and updates to these events • Implement user configurable 'priority cutoff' filter • Provide information on vulnerability instances and asset valuations from JNDMS into ISM to help with ISM correlation.	See Section 2.2.5 Section 2.2.5.4.1 discusses information flow into ISM from JNDMS	Initial alert handling complete, see Section 2.2.5 Deployment tasks identified for TA2 (see section 2.4.2)
1	System	XML memory management	The method used to parse XML files can be optimized to improve memory consumption. At the moment this is not seen as a major issue, however it would be a risk mitigation effort to balance additional memory load put on by the DSS activities.	Updates to Jaxb (See Section 2.2.7)	Complete

Pri	Component	Task	Description	Notes	Status
2	DSS	Database optimization	It has been identified that database interaction is causing several issues with the DSS. These issues include long query times and access of the database at inappropriate times such as in the middle of processing. This task would involve the following: • Review and optimization of key queries • Ensuring connection pooling is properly done in all instances • Maintain more information in memory so that queries do not have to be done as often. Database queries and updates should be confined to initialization and clean up routines. • Effort to ensure that all in-memory information is consistent with the database. Risk: • Moving more information into in memory data structures may cause memory issues, however initial investigations show that this is manageable.	See Section 2.2.6	In memory models separated, see Section 2.2.6 Deployment tasks identified for TA2 (see section 2.4.2)

Pri	Component	Task	Description	Notes	Status
2	DSS	Vulnerability Definition optimization	The scalability effort so far has identified issues with the handling of large numbers of definitions and instances. This task would be to eliminate the current bottlenecks.	See Section 2.2.6	Large ingestion of vulnerability definitions can now be run, however performance may still be an issue TA2 work identified in section 2.4.2.
2	DSS	Algorithm Optimization	There are a number of areas that can be optimized by reviewing processing loops. This task provides some effort to evaluate bottlenecks identified by the scalability study. Small changes done during scalability investigations for this task have already provided about 50% performance improvement in some areas.	See Section 2.2.6	Several optimizations complete, some new issues identified (see Section 2.2.6) . Additional work identified for deployment.
2	Portal	Portal System Upgrades	There is a significant update to Tomcat (v6.0) available as well as a new update to the Liferay portal. It has been identified that the newer Tomcat may provide performance improvements during initialization and during heavy loads. The new Tomcat can also be easier to maintain. An optional part of this task (time permitting) would be to migrate the data store of Liferay into the Oracle database.	See Section 2.2.7	Tomcat, Liferay and many components updated Some verification scheduled for TA2.

Pri	Component	Task	Description	Notes	Status
			The current Liferay installation uses HSQL and the use of a single data store for JNDMS would ease maintenance and possibly perform better under high loads.		
2	System	Location inference	The location information in JNDMS is generally entered as part of the operation data, however in most instances a new IP address will exist in the same location as the rest of the subnet. The network subnets are generally strongly related to the physical locations. To ease the assignment of location information for the much larger networks this task would add the ability to infer the location of the newly reported asset based on the location of other assets in the same subnet or zone. For example we could see if any assets in the same subnet have a different location, and if they don't it should be reasonable to assume that the location of the new asset is the same. This ability should be able to be turned off if required.	-	Pending. This is scheduled for TA2 (see section 2.4.2).

Pri	Component	Task	Description	Notes	Status
2	System	Batch processing of client requests	The data loads can take a very long time which may impact the ability of the system to process events at high loads. This task is to allow batch events to be read in which will eliminate the overheads related to initializing the JVM and key libraries such as support for web services.		Complete
2	Topology	Initial data load	Dynamic updates within the applet are already implemented, however this task will help the applet during initialization.	See Section 2.2.8	Complete
2	Topology	Cancel data loads		See Section 2.2.8	Complete
3	DSS	Vulnerabilit y scan optimization	One of the issues identified would be in dealing with patches and what vulnerabilities are covered. Currently identified data sources do not readily provide this information and tool vendors such as nCircle or Symantec consider this information proprietary. Each of these companies will maintain their own mapping of patches/fixes to vulnerabilities to try and provide a competitive edge for their respective tools. To address this issue during deployment a number of things should be done:	See Section 2.2.6	Pending. This has been scheduled for deployment efforts.

Pri	Component	Task	Description	Notes	Status
			• Update vendor and product information to conform to the CPE standard. This is not the closest thing the industry has as a standard. • Draft common procedures to manually updating patch/fix information. This manual process should allow for common or high priority fixes to be addressed by the operators. • The credibility of the source should be considered when integrating results from multiple scanners. In this case we can identify which sources are preferred when assessing what of the current list of vulnerability instances are active. This will allow tools that have access to detailed mapping to override internal calculations or identification by other tools that may not have the same level of insight.		
3	DSS	Correlation updates	The current implementation has two forms of correlation. The first represents cause and effect and is identified as a parent/child relationship in the portal.	-	Pending. This has been scheduled for deployment efforts.

Pri	Component	Task	Description	Notes	Status
			The other type of correlation (coincident) is more general and is noted as a percentage of correlation. It has been noted that, especially as the number of events grow, these two views are related. This provided two possible changes: • Allow the user to 'fuse' the events to create the parent/child relationship • If the coincident correlation is strong enough the events should be fused automatically. This task should also integrate the time of events in the correlation scheme.		
3	Mapping	Map Query Optimization	This task will evaluate and update database queries for the mapping code. This will also include evaluating connection pooling. Assumption: • This task assumes that the GIS task of TA1 has completed and migrated the mapping component to a more modular approach.	See Section 2.5	Core queries are complete, as new queries are required they will be examined. Additional effort has been scheduled for deployment efforts.
3	Mapping	Thread safety	There have been some issues noted with thread safety within the mapping core that may cause issues as the processing load scales.	-	Complete

Pri	Component	Task	Description	Notes	Status
			Assumptions: • This task assumes that the GIS task of TA1 has modified the mapping core which will make the code more manageable.		
3	Portal	General Navigation or view updates	There are several areas in which updates to the display would be beneficial. This would include: • Updates to the event views to hide child incidents. • Tweaks to the navigation based on usability. This would include a few updates to views to facilitate the creation or editing of relationships from additional details pages. This task will provide additional time to prioritize and implement key changes.		Incident view is updated, additional relationships may still be required Effort for navigation or view updates has been scheduled for deployment efforts.
3	Portal	SQL Query Optimization	This task would include a few days to identify top queries, then several hours per query to optimize.	Several optimizations have been done. The most significant was an update to display zones in asset lists.	Initial review done, likely more to be done Some effort has been scheduled for deployment efforts.

Pri	Component	Task	Description	Notes	Status
3	Portal	Database connection optimization	The use and maintenance of connections to the database can be improved by the use of connection pooling software similar to the way the JSS manages connections.	-	Pending This has been scheduled for deployment efforts.
3	Portal	Navigation List Optimizatio n	Some of the lists contain data in tables that is itemized by location. These could be slow and may be improved with Ajax updates.	-	Pending This has been scheduled for deployment efforts.
3	Topology	Layout algorithm improveme nts	The layout of very large datasets can be an issue. This task can address some issues; however this is not a major overhaul to the layout. Some tasks to consider include: • Ensuring the dynamic loading doesn't have too many assets on the screen at one time. • Allowing the operator the ability to provide hints or clues to the layout, such as selecting an area. • Improve grouping of icons. Assumptions: • This will be investigated during TA1 to determine what can be done within the allocated scope.	See Section 2.2.8	Investigations complete, core changes pending The additional effrot has been scheduled for deployment efforts.

Pri	Component	Task	Description	Notes	Status
4	Topology	Memory management	This is to address the memory management within the applet when using large data sets.	See Section 2.2.8	Initial work done, may depend on browser configuration This has been scheduled for deployment efforts.
4	Topology	Updates to visual cues	This task is to provide additional cues to help navigate larger data sets.	-	Pending This has been scheduled for deployment efforts.
4	Topology	Additional views	The core of the visualization applet can be expanded to provide additional views to help with larger data sets.	-	Pending This has been scheduled for deployment efforts.

2.2.5 Intellitactics Security Manager (ISM) Task

Some modifications are needed to both the DRDC SIM and the NIOSOC SIM installations in order to integrate with JNDMS. The diagram below shows how Snort IDS alerts flow from the DREnet, through the DRDC SIM and into the NIOSOC SIM via the Data Fork. The NIOSOC SIM then processes the individual IDS alerts and tracks all of the correlated alerts in memory. The correlated alerts table is written to a file at a set interval. A Java application polls the directory where the correlated alerts table gets dumped and looks for the latest file. It then parses the file and sends the relevant data fields from each correlated alert to the JSS/DSS. The DSS calculates risk for the affected assets based on the types of correlated alerts and the vulnerabilities that exist on those assets.

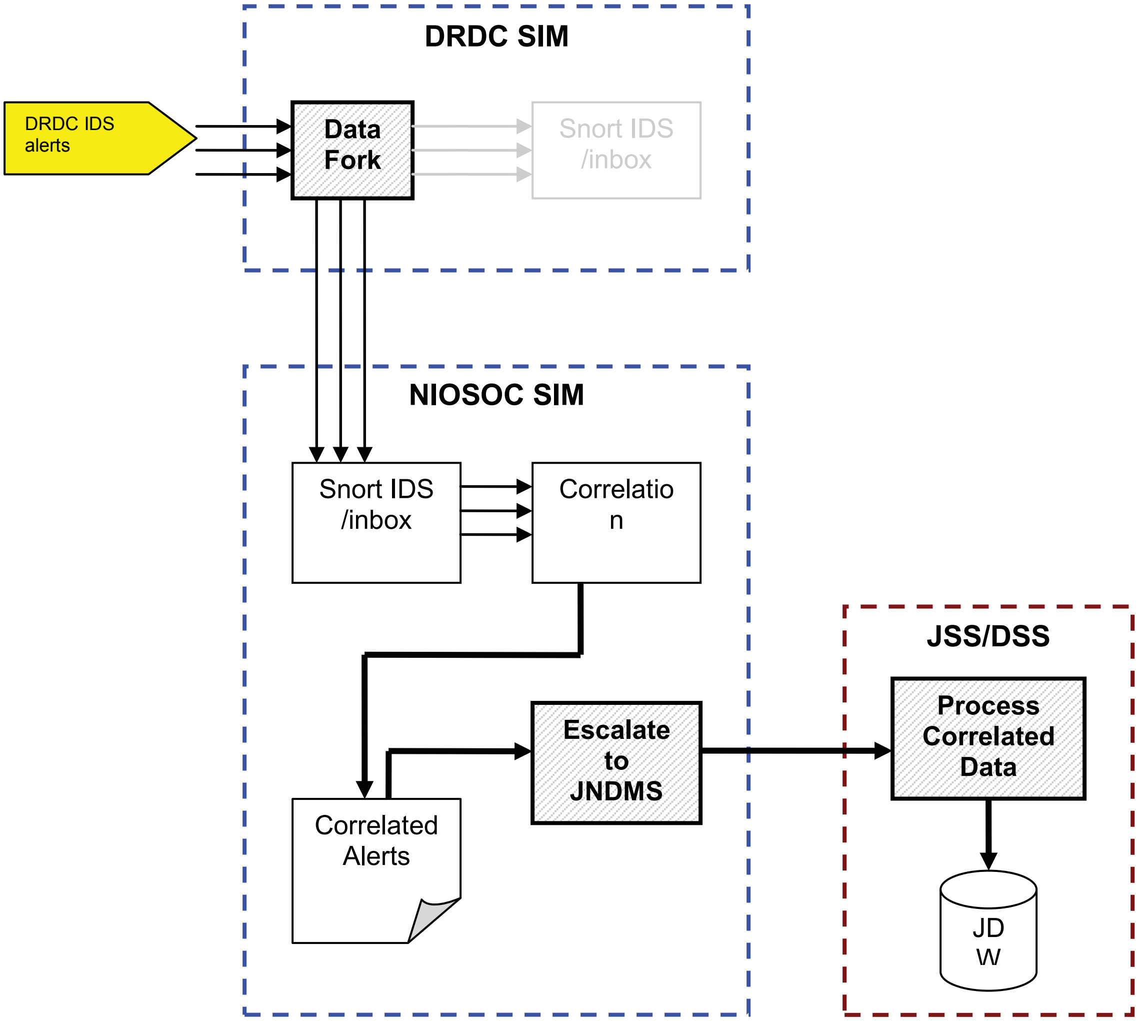


Figure 5: ISM into JNDMS Overview

2.2.5.1 Data Fork

The purpose of implementing the Data Fork is to send a copy of every IDS alert that the DRDC SIM receives to the NIOSOC SIM. Snort IDS alerts are written to /var/nsm/inbox/nids/snort.org/snort/2.x/db and /var/nsm/inbox/nids/snort.org/snort/2.x/db_2 on the DRDC SIM. The DRDC SIM collects the alerts from /var/nsm/inbox/nids/snort.org/snort/2.x/db and processes them. A Perl script runs on the DRDC SIM that transfers the alerts using scp from /var/nsm/inbox/nids/snort.org/snort/2.x/db_2 to the Snort IDS inbox located at /var/nsm/inbox/nids/snort.org/snort/2.x/db on the NIOSOC SIM. The NIOSOC SIM then collects the alerts from the Snort IDS inbox and processes them. If the NIOSOC SIM is unavailable for an extended period of time, the backlog of Snort IDS alerts on the DRDC SIM will be purged based on a defined time period. This is a safeguard to prevent the file system on the DRDC SIM from filling up.

2.2.5.2 Correlated Alerts

The NIOSOC SIM stores the correlated alerts graph in memory. An Intellitactics operation has been defined that dumps the entire contents of the correlated alerts graph into a log file in the /tmp/correlated_alerts/ directory at a user-defined interval. These log files are parsed by a separate application and the correlated alerts data is transferred to the DSS for processing by JNDMS.

2.2.5.3 Correlated Alert Escalation to JNDMS

The "Send Correlated Alerts to JNDMS" Java application will scan the directory where the correlated alerts files are stored and retrieve the latest file. It will parse the file and extract the data fields for each correlated alert that are required by the DSS. Additional information will be mined from the Intellitactics Security Data Warehouse (SDW) MySQL database to gather greater detail relating to the individual events within the correlated alert. Each correlated alert and event information will then be sent to the DSS via the JSSClient where risk analysis will be done for the affected assets.

The following information describes a single correlated alert. It can be retrieved from the data files that the NIOSOC SIM dumps periodically from the correlated alerts graph in memory:

id	1970325977501740
alert_id	111cea30-9f84-11dd-bb50-001b78be1cba
insert_type	Update
alert_count	25
type	Abuse/Misuse by Critical Asset: 10.142.2.23 (DREnet.ABC.XXYYZZ.Intranet)

The following information can be retrieved by querying the alert_events table on the NIOSDW by using the alert_id value as the key. Each entry describes an event that contributed to the creation and maintenance of a correlated alert.

count	nsm_type	source(s)	target(s)	target port(s)
4	ids.detect.compromise.web	10.142.2.23	67.195.37.171	80
6	ids.detect.compromise.web	10.142.2.23	66.249.84.67	80
15	ids.detect.insecure.web	10.142.2.23	67.195.37.171	8080

By combining the information from the correlated alerts file and the alert_events database table on the NIOSDW, the DSS will have enough information to calculate and propagate risk throughout the JNDMS environment. The Java application will communicate a set of correlated alerts to the DSS and each correlated alert will reference many alert_events.

The correlated alerts file is parsed at a set interval. Only the correlated alerts that have been updated since the last time the file was processed will be retrieved and processed. The Java application will locate new and recently updated correlated alert records based on the record's “last_seen” attribute. For each of these updated correlated alerts, only the most recent event details will be retrieved from the alert_events table of the NIOSDW database. Each alert_events table record contains a timestamp that identifies when the alert_events record was created. The Java application will locate new alert_events records based on this timestamp.

In order to further reduce the information sent from the NIOSOC SIM to the DSS, the Java application will aggregate (count) similar alert_events. A counter will identify the number of events that contain identical nsm_type, source_ip, target_ip and target_port values.

2.2.5.4 Process Correlated Data (JNDMS)

Below is an algorithm for tracking the correlated alerts in JNDMS as they are received from the NIOSOC SIM. This information will be stored in two new tables in the JDW: the correlated_alerts and alert_events tables. The correlated_alerts and alert_events tables will be created and/or updated with each correlated alerts update from the NIOSOC SIM.

```
COMPARE the new data with the data currently stored in the JDW

IF the id does not exist

   ADD the new correlated alert to the JDW correlated_alerts table

   ADD the associated alert_event records to the JDW alert_events table

   CREATE a new INCIDENT

ELSE (the id exists)

   ADD or UPDATE the associated alert_event records to the JDW
   alert_events table

   UPDATE this INCIDENT

ENDIF

IF an existing correlated alert has not been updated with new
information after a certain time period

   DELETE the associated alert_event records from the JDW alert_events
   table

   DELETE the correlated alert from the JDW correlated_alerts table

   RESOLVE the corresponding INCIDENT

ENDIF
```

NOTE: We only considered the seven types of correlated alerts that are reported by the DREnet ISM based on IDS alerts (see section 2.2.5.4.2). There may be others that we need to analyze.

NOTE: The amount of time that the DSS will wait before automatically resolving an incident will be configurable. This configurable value and the appropriate resolution mechanism will be evaluated as part of DREnet scenarios.

2.2.5.4.1 ISM Critical Assets and Vulnerabilties

The ISM considers an asset critical if it is assigned an operational risk greater than zero. It uses a scale of 1-5 to rate the importance of an asset and will use this rating to determine when and if to escalate the events to JNDMS.

The current configuration depends on creating a configuration file that identifies the critical assets. This currently will list the primary servers responsible for services deemed important.

The ISM task has examined the potential of synchronizing both the asset importance and the vulnerability data between JNDMS and ISM. This synchronization would allow a more meaningful escalation of alerts for JNDMS from ISM. This will be expanded in the follow on tasks, however, the expected configuration of the JNDMS ISM instance during deployment will include Intellitactics native integration with vulnerability scanners as well as an import of assets with defined importance from JNDMS. This would mean that although not all vulnerabilities would be visible to ISM, the vast majority would be, and this level of integration is seen as configuration of ISM only. This would also mean that the importance placed on assets within ISM are derived from the analysis done by JNDMS.

2.2.5.4.2 Correlated Alarms

This section identifies correlated alarms that will be broadcast from ISM.

IDS Matched Vulnerability

- A managed asset has been the target of an attack.
- The attack type matches a known vulnerability on the target asset.
- The attack is most likely successful – the vulnerability has likely been exploited.

Abuse/Misuse by Critical Asset

- This alert triggers when a critical asset is seen as the source of abuse/misuse activity.
- This rule requires that the operational risk of the managed asset is greater than 2, but the threshold is configurable.
- The ISM will also create correlated alerts of type “IDS Matched Vulnerability” if the activity targeted known vulnerabilities within managed assets.

Restricted Target

- This alert is triggered when network activity is observed involving a restricted target host.
- The restricted targets are populated in the lookup list, "Restricted Targets" with the zone names, host names and IP addresses that are restricted.
- The "Restricted Targets" lookup list will typically not refer to managed assets.

- If a managed asset is the source of this activity, it could imply that the managed asset has been compromised.

Restricted Target Port

- This alert is triggered when network activity is observed involving a restricted target port.
- The restricted ports are populated in the lookup list, "Restricted Target Ports" with the numbers of the ports which are restricted.
- If a managed asset is the source of this activity, it could imply that the managed asset has been compromised.

Restricted Source Port

- This alert is triggered when network activity is observed involving a restricted source port.
- The restricted ports are populated in the lookup list, "Restricted Source Ports" with the numbers of the ports which are restricted.
- If a managed asset is the source of this activity, it could imply that the managed asset has been compromised.

Recon Followed by Attack

- This alert is triggered when a host is seen performing reconnaissance activity, followed by an attempt to compromise a host.
- The source host and/or the target host(s) may be managed assets.
- If a managed asset is the source of this activity, it could imply that the managed asset has been compromised.
- The ISM will also create correlated alerts of type “IDS Matched Vulnerability” if the activity targeted known vulnerabilities within managed assets.

IDS Many Alerts Targeting Critical Asset

- This correlation produces an alert when a critical asset has been recorded as a target by many IDS/IPS alerts.
- The ISM will also create correlated alerts of type “IDS Matched Vulnerability” if the activity targeted known vulnerabilities within managed assets.

Table 4. DSS Action Summary

ALERT	DSS ACTION
IDS Matched Vulnerability	Large increase in risk on managed asset. Raise risk depending on the vulnerability type. Propagate risk throughout zones / neighbours. Assume this asset has been compromised.
Abuse/Misuse by Critical Asset	Small increase in risk on managed asset. If the target is also a managed asset, small increase in risk on target asset. If subsequent investigation determines that the managed source asset has been compromised, large increase in risk on managed source asset.
Restricted Target	If the source is a managed asset, small increase in risk on source asset. If subsequent investigation determines that the managed source asset has been compromised, large increase in risk on managed source asset.
Restricted Target Port	If the source is a managed asset, small increase in risk on source asset. If subsequent investigation determines that the managed source asset has been compromised, large increase in risk on managed source asset.
Restricted Source Port	If the source is a managed asset, small increase in risk on source asset. If subsequent investigation determines that the managed source asset has been compromised, large increase in risk on managed source asset.
Recon Followed by Attack	If the source is a managed asset, small increase in risk on source asset. If the target is also a managed asset, small increase in risk on target asset. If subsequent investigation determines that the managed source asset has been compromised, large increase in risk on managed source asset.
IDS Many Alerts Targeting Critical Asset	Small increase in risk on target asset.

2.2.6 DSS Updates

The effort to update the DSS examined a number of potential bottlenecks within the system and identified potential tasks to further define or correct any performance barriers. It was identified that several key areas required attention, including modularization, correlation, vulnerability processing, database optimization, and examining the algorithms.

It was identified that two key processes, the event correlation and the risk analysis, were responsible for much of the processing done by the DSS. It was further recognized that quick access to the required information and repeated queries of the database both contributed to the issue. The first tasks were to tackle the database queries and to modularize the internals of the DSS.

The database optimization concentrated on ensuring that in memory models could store the bulk of the information required for the analysis of events and that database queries should, ideally, be kept to updating these models, then writing the analysis results back into the data warehouse.

The modularization task concentrated on the two key processes, the event correlation and the risk analysis. These two processes represented performance barriers for many different event types.

Figure 6 shows some of the main components within the DSS after the initial modularization.

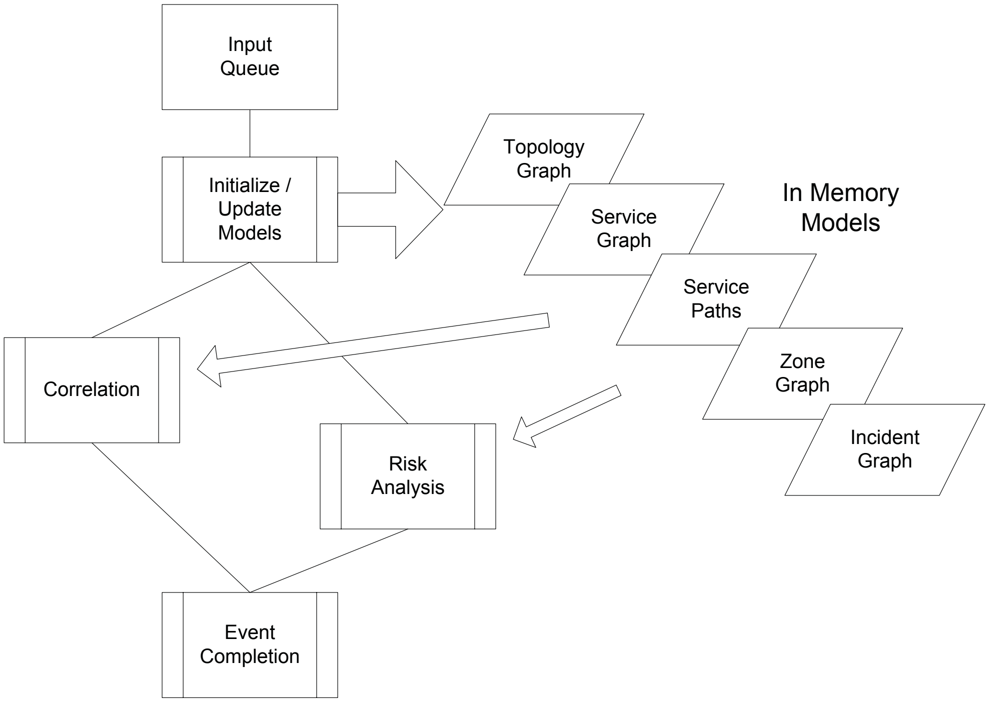


Figure 6: DSS Overview

It had been identified early in cycle 3 that the risk analysis did not depend on the order of events processed and that it could be run at any point. This led to the implementation of the risk timer so that even processing would not be held up by the risk analysis. On a periodic basis the risk analysis would be done and would ensure that the impact of all complete events would be taken into account. This led to a conceptual disconnect with the views in the portal because it wasn't apparent which events were included in the current risk analysis and which ones were not. This also led to a noticeable lag between the events being seen and the eventual update to the risk indicators.

The alternative of having risk being processed on every event is not realistic either. This would lead to longer processing times for events than necessary and if events enter the system faster than the processing can be done, then the system will get further and further behind. The scalability task has to plan for more events.

The results of these investigations and the initial modularization work led to a new solution. The risk analysis, now that it has been modularized, can be run in its own thread. This ensures that other processing, such as the event correlation, can continue at its own pace. The updated risk analysis processor will implement a queue at the start of the processing that stores all events that must processed. The risk analysis, instead of processing these one at a time, will extract all available events at once and perform the risk analysis. It will then signal to the database that these events have been processed.

This processing technique is seen as allowing significant event flows, even if the internals of the analysis takes a significant amount of time. This also minimizes, as much as possible, the lag between the event entering the system and the analysis because it will process events immediately if an existing risk analysis is not in progress. This method tends to be self correcting such that if the events are entering the system faster than the risk analysis can be performed, then the risk analysis will process more events at once (all those available on the queue).

The current modularization and database optimization has separated out the event correlation, the risk analysis and the in memory models. The full implementation of the input queue for the risk analysis is recommended before deployment.

Another component examined was the vulnerability processing, both for the definitions and the scans. The initial work was to ensure that these would use the in memory models as much as possible and to fix a number of issues relating to memory management and how large numbers of vulnerability definitions can be processed. The current implementation, however, can still take a significant amount of time. To process the full 30,000+ vulnerability definitions on about 7,500 assets, creating in excess of 11,000 vulnerability instances it will take more than 12 hours. This is an extreme case of processing all known vulnerabilities at once, however it is still a target for further optimization. These components can be separated into their own threads and can undergo further optimization within their respective routines.

Another area examined was the algorithms within the DSS. Various optimizations and clean ups of the existing algorithms have provided a number of performance improvements in the area of 50% - 100% gains. These changes in combination with the database optimizations have led to gains over 600% in some areas.

As well as the pure performance of the DSS being examined the results of the risk analysis were examined as the data sets grew. It was noted that providing the link between the risks of individual assets and the overall operation becomes increasingly difficult. It also becomes difficult to provide a stable risk analysis in our current implementation. To address these concerns we evaluated the risk assessment and found that some changes would be required. It is recommended that this effort continue and provide an updated risk assessment before deployment.

2.2.7 Portal and System Upgrades

There were a number of updates to system components to address issues. The following list identifies some of the major updates:

- Tomcat updated. This was done for two reasons. First it was done to ensure that we are running on a reasonable new, supported platform. This was also done because one of the known advantages of upgrading from v5 to v6 is the ability to handle more events. It is not likely that Tomcat would be the primary bottleneck in event processing, however if an issue did arise we would unlikely be able to address after deployment.

- Liferay updated. This was done primarily because of the number of other updates that were being addressed. There were many improvements to the portal and we wanted to ensure we remained on a supported platform. This was a major upgrade from v4 to v5.

- Switch from Apache Axis to Apache CXF for web services. There were a number of issues relating to the version of Axis that we were using. This was causing significant issues in memory management and performance. It was particularly noted that Axis 1.2 could not handle large SOAP payloads, which we sometimes used. We examined updating to a new version of Axis (1.4) but initial prototyping show similar issues. An alternative called Apache CXF was chosen. This API also follows the Java JAX-WS standard.

- Jaxb updates. We had a number of issues with memory management and performance related to the processing of large XML files. Much of our parsing of the XML is done by Jaxb so this was updated.

- Dojo updated. The investigations showed that the rendering of Dojo components was causing a significant performance impact on all pages. This was especially noticeable on lists that used the drop down filters. The replacement of these filters with HTML drop down boxes drastically reduced the functionality and still had a significant performance impact when large lists were used.

- A replacement for Dojo, YUI (http://developer.yahoo.com/yui/) was prototyped, however many layout and display issues were found. The updated version of Dojo was also examined and it was found that the methods used to dynamically update the visual components (widgets) was dramatically improved.

- In addition to just the Dojo update the way that the filters were used and loaded were updated. The new filters perform much better and offer more functionality.
- Updates to data views. In addition to the Dojo updates (which were significant) additional modifications were made to various page views to improve performance. This would include query optimization and updates to HTML or Ajax.

2.2.8 Visualization Applet Updates

The initial investigations showed several issues with the visualization applet. On larger data sets it would take a significant amount of time to load and when the number of assets grew to about 3,500 it simply would not load. It was also noted that the ability to navigate and layout the larger graphs was problematic.

A number of tasks were identified to address these issues. The first one was to be able to cancel the data loads. It was found that during large data loads the portal would be unresponsive and this would at least allow the user to see partial data or navigate elsewhere. The user can now cancel the initial data loads.

The next task was to improve the performance of building of the graphs and displaying the results. The previous version would send assets to the applet one at a time to integrate into the graph, this caused delays in sending the data but also put more of a strain on the layout engines. The applet was updated to send batches of nodes (assets, events, etc) to the applet so that the user still sees periodic updates as the data loads, but that the applet can process nodes in bulk. This has resulted in a view of about 3,500 assets in less than 20 seconds. The performance of the layout algorithms is now also quite responsive for the larger data sets (~2s for ~3,500 assets).

Another issue related to the visualization applet was its memory management. There were significant issues that were fixed and the memory required for each node was reduced to a minimum. The results is being able to view up to at least 3,500 assets on a graph at a time. The limit that is reached is because of the default memory allocated to the Java Virtual Machine within the browser. The browser can be configured to allow for more memory to be allocated to the JVM, which would allow more assets to be viewed at once.

There were a number of visual cues and views examined and an indicator was added to show when data loading is in progress. It is recommended that additional views be examined before deployment.

2.2.9 Scalability findings

The following table shows some of the key areas examined before and after various updates related to scalability were implemented.

Table 5: Scalability Summary

Task	Data	Pre Scalability	Post Scalability	Notes
Asset Load	Initial data load before software load (~780 events)	18m 02s	01m 23s	This was primarily the reporting of new hosts and network connections on a new database.
Event Processing	Per event, taken from build event processing	500 ms – 2300 ms (avg 728 ms)	15ms – 375ms (avg 90ms)	This mostly measured the overhead incurred by the JVM on the client side and web service processing.
Topology View	~3,500 assets	Did not complete	20s	Initial data load of assets.
Vulnerability Load	~7,500 asset, ~30,000 vulnerabilities	Did not complete	12 hours	This represents a complete load of all known vulnerabilities from NVD.
Analysis (risk and correlation)	Default data load, 15 events	38m	6m	This identified a sequence of events that would escalate risk.
View Product List in Relationship	~65,000 products	7m 30s	6m	The product relationship could be updated to use a 'paged' view (see view product list below). The new relationship has the filters active.
View Product List	~65,000 products	2s	2s	This uses paging to reduce the amount of information sent to the client.
View all Assets	~7,500 assets	Did not complete within 10m	9s	This was a result of database optimization and page updates.

2.3 Analysis and prototyping of interface between JNDMS and nCircle IP360

The analysis and prototyping of an interface between nCircle IP360 and JNDMS initially performed a feasibility study to determine how appropriate IP360 would be in the follow on deployment demonstration and to see if there are any issues relating to the integration of IP360 into JNDMS.

2.3.1 Feasibility

The Department of National Defence is in the process of deploying the nCircle's IP360 vulnerability management system on the DWAN. Currently JNDMS only supports Tenable's Nessus vulnerability scanner. The goal of this document is to explore the potential avenues of integration of IP360 vulnerability data into JNDMS along with the level of effort and expected complexity of each approach.

It is noteworthy that the analysis of IP360 for this document was done without the benefit of having access to the actual product, nor key documentation such as install, configuration, administration, or usage guides. Additionally there was extremely limited support in terms of access to personnel with relevant IP360 expertise. For these reasons, there may still be gaps in the conclusions or recommendations contained herein. Some initial consultation took place with IPSS engineers, however detailed information was not available until after the IP360 system was procured.

2.3.1.1 Role of IP360 in JNDMS

First and foremost nCircle IP360 is a Vulnerability Scanner. In the JNDMS context, this means that IP360 produces "Vulnerability Instance" data that is used by the JNDMS system in a consistent fashion regardless of the source. The question then is; what is the most efficient means of getting that data into the existing JNDMS Vulnerability Instance interface, and what changes, if any, are required to the existing interface.

The IP360 rollout on the DWAN is a major initiative for the JNDMS "target client" (i.e. DND) so ensuring that JNDMS harmonizes well with IP360 could be an important benefit from a political and technical perspective.

Showing the applicability of JNDMS technology to the DND customer, relevance to their environment, and the flexibility of the JNDMS technology to integrate with their tools should generate positive interest in JNDMS. Additionally, since the goal of JNDMS is to provide Situational Awareness (SA), having access to a complete and regularly updated source of vulnerability data is an important part of the requisite Computer Network Defense (CND) data to provide that SA.

2.3.1.2 IP360 Overview

2.3.1.2.1 Architecture and Terminology

According to nCircle's documentation, IP360 is a scalable, enterprise-class vulnerability and risk management system that proactively delivers a comprehensive view of network risk and enables cost-effective risk reduction. It discovers detailed intelligence about IP-enabled devices on the network, and utilizes best-in-class reporting and analytics to prioritize a comprehensive view of network risk.

The IP360 platform is delivered via hardened, non-Windows appliances and designed for scalability, rapid deployment, and ease of management. It can identify over 1400 operating systems, 3800 applications, and 3500 vulnerabilities and the list grows daily. It provides comprehensive, agentless network discovery and profiling of all network assets, with vulnerability and security risk assessment.

Given proper credentials, it has the ability to gather more detailed information about each monitored device, including specific host configurations such as password requirements and file permissions. As well it has the ability to test for vulnerabilities using credentials via SSH, SNMP, and Windows (SMB) along with remote unauthenticated testing.

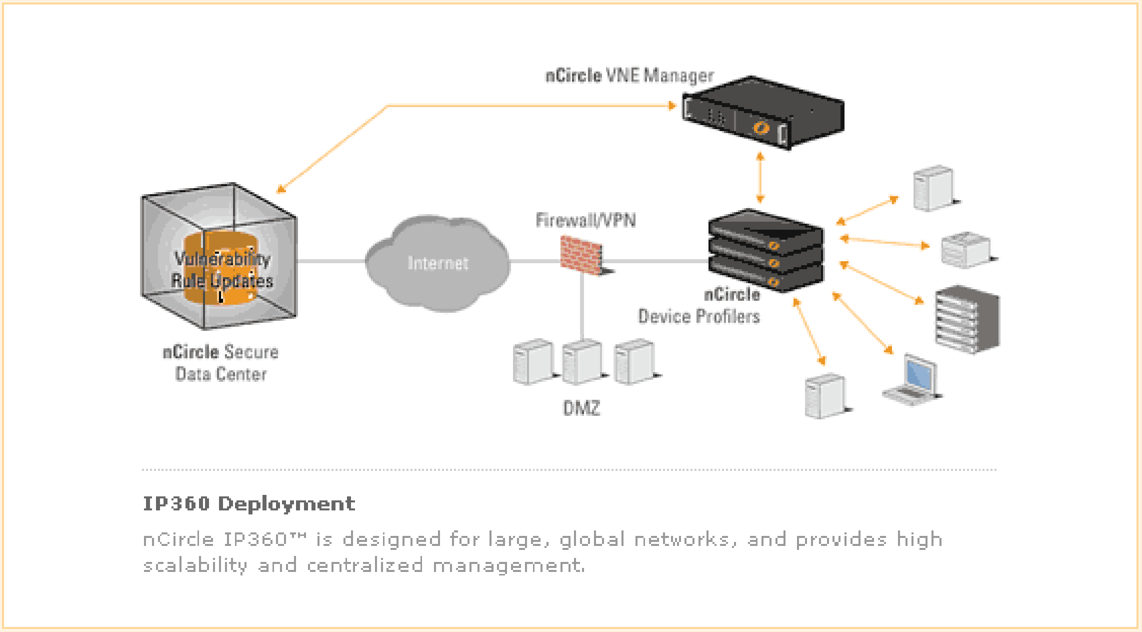
It is noteworthy that the use of the word “risk” within nCircle differs substantially from the concept of risk within the JNDMS context, and is principally a measure of the severity of the vulnerability using scoring similar to CVSS.

While IP360 does include a “discovery” capability, this is limited to generating lists of active IP addresses discovered within subnets specified by the user. This does not appear to include any type of topology information or any ability to discover IP addresses that the tool is not specifically configured to enumerate.

2.3.1.2.2 Components

The VnE Manager (VnE) is used to coordinate scan options, scheduling, bandwidth limits. The Device Profiler (DP) takes commands from the VnE, performs the actual scanning, and sends the results back to the VnE.

The Security Intelligence Hub (SiH) is an optional component of the architecture (but part of the DND IP360 deployment) that links to the VnE as well as to other information and security systems such as asset management, IDS/IPS, and Security Information Management (SIM). The SiH generates various reports including base lining and trend analysis. It also allows for long term data retention of scan results beyond what the VnE is designed to retain.



IP360 Deployment

nCircle IP360™ is designed for large, global networks, and provides high scalability and centralized management.

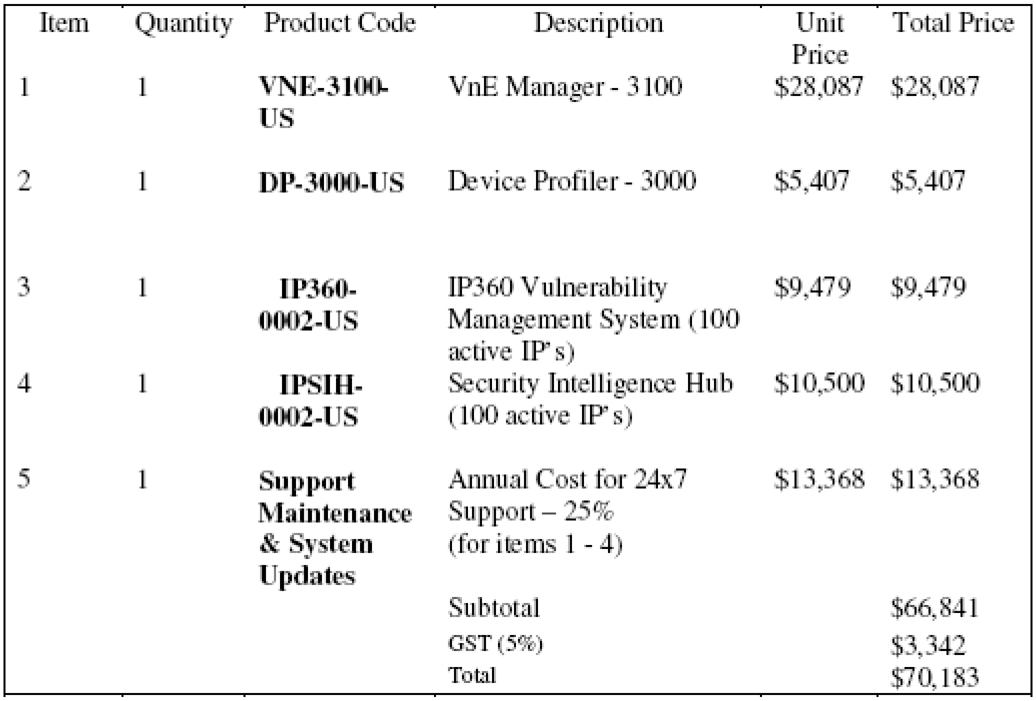
Figure 7: A Sample IP360 Deployment

2.3.1.2.3 Licencing

The tools, equipment and licenses for the nCircle products selected for a limited DREnet deployment are as follows:

- 1 VnE 3100 manager appliance
- 1 Device profiler
- IP scanning licenses for 100 IP addresses
- Security Intelligence Hub reporting for 100 IP addresses.

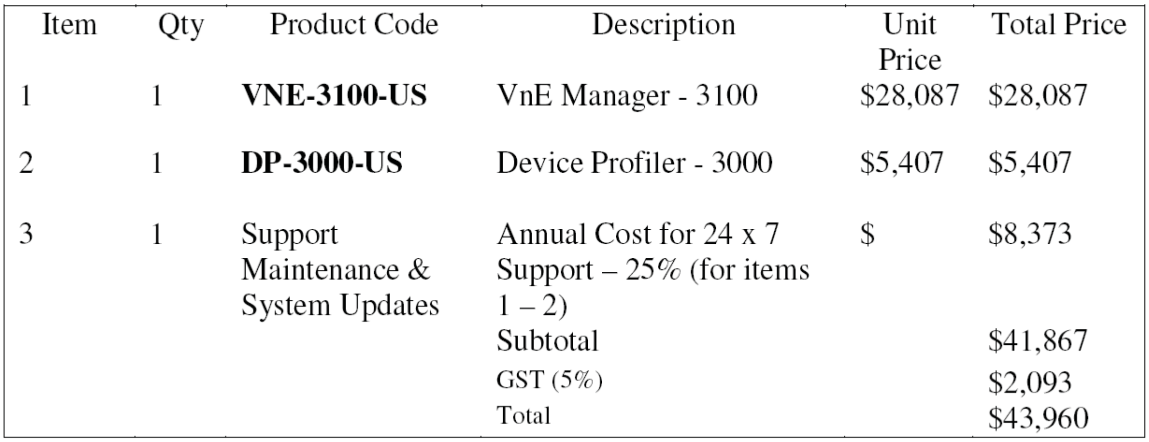
Table 6 includes all components and their cost breakdown which would be required for a limited deployment of JNDMS within the DREnet. This is taken from a quote from the local (exclusive) nCircle vendor, InfoPeople Security Solutions (IPSS). This quote can be extended to 3000 IPs (a full DREnet deployment) at a per IP cost of $13-$14 for the VnE licenses and $3-$4 for the SIH licenses.



Item	Quantity	Product Code	Description	Unit Price	Total Price
1	1	**VNE-3100-US**	VnE Manager - 3100	$28,087	$28,087
2	1	**DP-3000-US**	Device Profiler - 3000	$5,407	$5,407
3	1	**IP360-0002-US**	IP360 Vulnerability Management System (100 active IP's)	$9,479	$9,479
4	1	**IPSIH-0002-US**	Security Intelligence Hub (100 active IP's)	$10,500	$10,500
5	1	**Support Maintenance & System Updates**	Annual Cost for 24x7 Support – 25% (for items 1 - 4)	$13,368	$13,368
			Subtotal		$66,841
			GST (5%)		$3,342
			Total		$70,183

Table 6: Quoted licence cost for complete IP360

An alternate funding mechanism had been identified by leveraging the existing IPSS contract with DND. In this scenario the JNDMS deployment efforts would provide only the IP360 hardware (see Table 7). The additional cost of the per IP and SIH licenses would be procured directly by DND. Information from IPSS identified the cost of 3000 licenses (both IP and SIH) would be $5.04 each for a total of $15,120 (IPSS quote GJ071708-nC-DRDC1). This would represent a significant reduction in the overall cost if the additional (more than 100) licenses would be required.



Item	Qty	Product Code	Description	Unit Price	Total Price
1	1	**VNE-3100-US**	VnE Manager - 3100	$28,087	$28,087
2	1	**DP-3000-US**	Device Profiler - 3000	$5,407	$5,407
3	1	Support Maintenance & System Updates	Annual Cost for 24 x 7 Support – 25% (for items 1 – 2)	$	$8,373
			Subtotal		$41,867
			GST (5%)		$2,093
			Total		$43,960

Table 7: Quoted cost for IP360 Hardware.

2.3.1.2.3.1 Licencing under existing DND contract

An alternative way to procure these components is through the existing DND IP360 procurement contract. This cost may be as low as $6 per monitored IP address, but would require procuring an allotment of 32,000 IP addresses. Under this method, there would be no hardware costs but it is unclear if the 32,000 IP address licenses would include the required VnE and SiH or simply a number of DPs. Furthermore, the DND IP360 contract contains no provisions to procure additional IP360 hardware components as required for the DREnet deployment. Additional details on this option are currently under investigation.

2.3.1.2.4 IP360 CONOPS

The JNDMS project team has access to a draft IP360 DND CONOPS document (Generic Report - Colour Logo) describing at a high level how IP360 would be deployed, administered, and used within DND. It does not contribute significantly to understanding how IP360 could integrate with JNDMS, however it does identify that IP360 can integrate with Security Information Management tools, specifically naming Intellitactics.

2.3.1.2.5 DREnet Deployment

Previously on the DREnet, vulnerability scanning was done during off-hours using Nessus scanners deployed on the main and site firewalls. This architecture eliminates the need for special “scanner traffic” firewall rules, since the scans come from the firewalls themselves. This also reduces the amount of traffic which must pass over the WAN links, since scan traffic for the 9 geographically dispersed sites originates at each of those sites. A further efficiency is achieved by only scanning servers which are exposed to the Internet or to the WAN by firewall policies, which limits the number of systems scanned to less than 100.

By contrast an IP360 deployment would involve scanning over the WAN from a single IP360 DP scanner. This architecture requires a very permissive set of firewall rules for the DP, and sends the scan traffic over the WAN. If instead of a single DP, the IP360 deployment involved 9 (geographically distributed) DP scanners, the issue of firewall rules would be mitigated because the DPs could reside within the “private” part of the site networks and be interconnected by secure IPsec tunnels.

The Nessus licenses for DREnet have lapsed and the current preferred replacement for vulnerability scanning is IP360. This direction by DREnet personnel makes it even more appropriate for JNDMS to be integrated with IP360 for the deployment demonstrations. It is expected that IP360 will scan approximately the same number of hosts that the previous Nessus configuration targeted, however the use of IP360 may change over time.

2.3.1.3 Integration Methods

Initial analysis and prototyping of possible interfaces between JNDMS and nCircle IP360 was undertaken to determine which integration options are most feasible and what advantages and disadvantages they offer.

2.3.1.3.1 Integration using SIM

One promising avenue of integration is to leverage the data acquisition capabilities of the Security Information Manager (SIM) to harvest, tokenize, and normalize the data before sending it to the JNDMS "core". Currently the SIM in question is Intellitactics Security Manager (ISM). Using this approach, the existing interface between the SIM and the JNDMS System Server (JSS) is extended and made more generic, allowing "Vulnerability Instance" events to be passed from the SIM to JNDMS in addition to the IDS alarm events that are currently supported.

Pertinent to this option, the SIM to JSS interface is also under consideration to be extended and made more generic to accommodate correlated and aggregated events in support of scalability issues. Taken together these changes would be a resource efficient way to leverage a much greater amount of native SIM functionality, and would yield the further side benefit of vastly increasing the number and variety of supported security device products which are supported by JNDMS (by virtue of their support by the SIM vendor).[1] Another advantage of creating an enhanced SIM to JSS interface is that any changes in the security products' data delivery mechanisms (and data normalization for signature and plug-in based alarms) are handled by the SIM vendor under product maintenance rather than requiring code changes to JNDMS, as is currently the case.

This approach also updates the JNDMS support for the Nessus vulnerability scanner at no additional cost or effort. Currently the "native" JSS support of Nessus is out of date, and requires engineering resources to update the Nessus to JSS interface if not replaced by the proposed SIM to JSS interface. Future updates to Nessus may also necessitate further updates to the Nessus to JSS interface.

Another benefit to the SIM integration is that ISM will have the vulnerability instance information available to augment the rules. These rules would be used to determine the escalation of events into alerts (see Section 2.2.5, Intellitactics Security Manager (ISM) Task).

Intellitactics provides support for nCircle IP360 through a variety of mechanisms, described in the following sections.

[1] See http://intellitactics.com/int/support/supporteddevices.asp for a list of devices which would "fully supported" (the Network-Based Intrusion Detection and Vulnerability Scanners) or "partially supported" (the other device classes, by virtue of the "correlated events" they produce).

2.3.1.3.1.1 SNMP Trap support

ISM supports the receipt of SNMP traps from IP360. Based on the sample IP360 traps furnished by Intellitactics each trap appears to report on a single vulnerability instance on a host. Discussion with an IPSS engineer suggested that the SNMP facility is configured in a very manual fashion and may not be suitable for a large deployment, but no documentation or product was made available to verify this. Discussion with an Intellitactics' device support developer confirmed that this type of support was only used by a single customer, and that it was not the preferred method of integration with the SIM.

It is also not known if this type of integration is achieved through the VnE or the SiH, but it could be assumed that this seemingly limited type of data export might be a constraint of the VnE, and only used by customers who do not deploy the SiH.

ISM support consists of two stages, as follows.

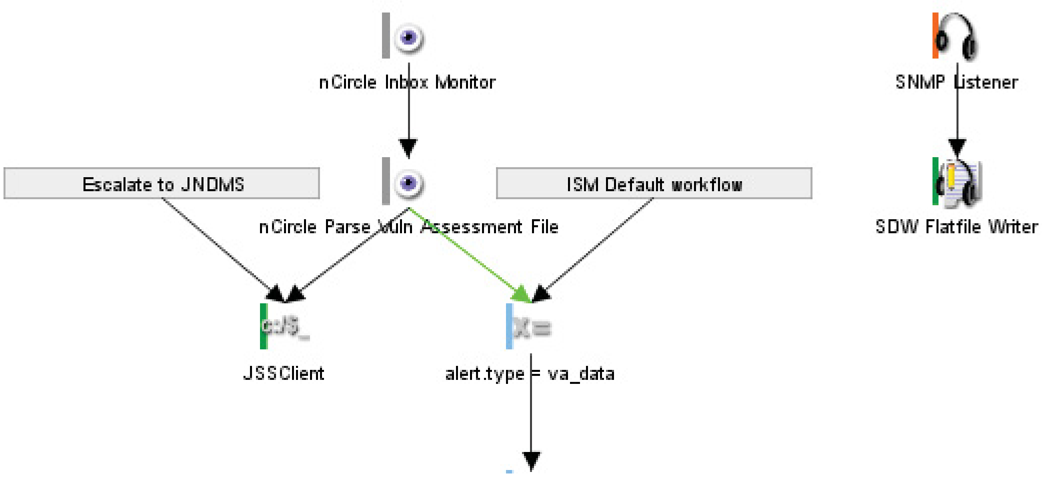


Figure 8: ISM IP360 SNMP Input Rule

An ISM SNMP listener receives the trap from IP360 and writes the data to an IP360 "trap file" on disk. Every 60 seconds an "inbox monitor" checks for all new IP360 trap files and processes them through a Device Modeling Framework Definition (DMFD) parse rule. The DMFD parser contains a number of (configurable) sub-processes to parse, tokenize, label, normalize, and timestamp all the messages. Each message results in a Vulnerability Instance event, which in this prototype is "forked" to the JSS Client.

2.3.1.3.1.2 XML Report support

The preferred method of ISM support for IP360 is through formatted data extracts from the SiH. Again based on the sample IP360 data furnished by Intellitactics there appear to be two different styles of data extract available called XML2 and XML3.

The two types of reports appear to contain different data attributes and one may be more appropriate than another for JNDMS purposes. By visual inspection of the sample data, the XML2 style report appears to be very similar in structure and content to the Nessus 2.0 XML scan output. The XML3 style report contains a great deal more in terms of the data attributes associated with each vulnerability instance, including such items as CVSS scoring information, required patch(s) to mitigate the vulnerability, CVE IDs, bugtraq references, and more. Extending the JNDMS Vulnerability Instance interface to take advantage of this additional data would require substantial additional effort for potentially little benefit, if as expected most of this "enhanced" data is already present in the JNDMS system from the National Vulnerability Database (NVD) feeds.

Both types of reports are supported by ISM through the same data acquisition and device modeling rule consisting of a single stage, as follows.

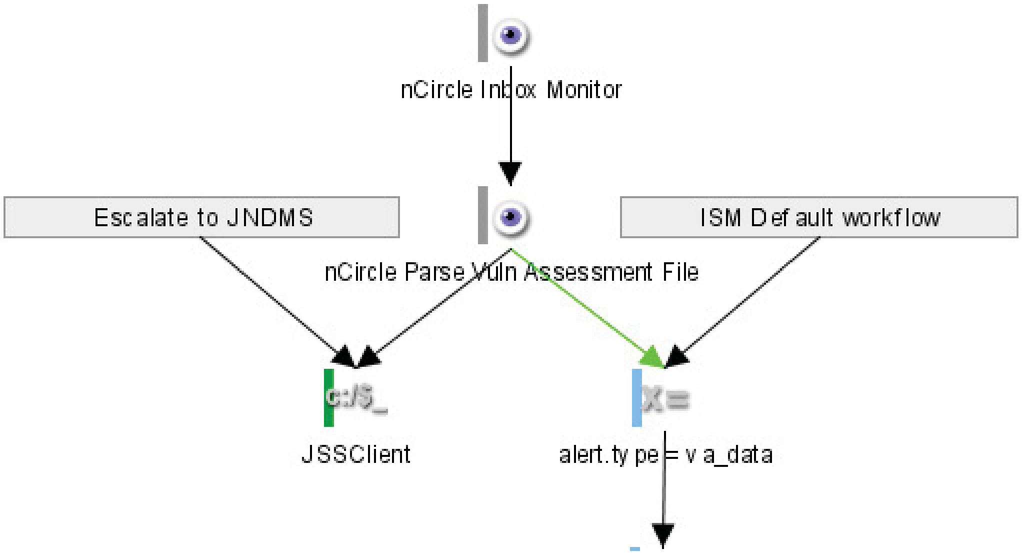


Figure 9: ISM IP360 Flatfile Input Rule

It is not currently known how the SiH delivers the XML reports, but it is likely that a mechanism such as an SCP copy triggered by a regular cron job would be used to pull the reports from the SiH into the appropriate ISM inbox. From there the ISM inbox monitor would poll that location on disk for new reports and process them through a Device Modeling Framework Definition parser rule. The DMFD process contains a number of (configurable) sub-processes to parse, tokenize, label, normalize, and timestamp all the messages. As with the SNMP option, each message results in a Vulnerability Instance event, which in this prototype is "forked" to the JSS Client.

2.3.1.3.1.3 Prototype SIM interface

A partial prototype of an IP360 to ISM to JNDMS interface has been built as part of the initial investigation, however it does not work end to end, and the IP360 data used in the prototype was canned sample data and did not involve configuration or use of the IP360 product.

The portions of the SIM interface that have been successfully tested or prototyped are:

- the flat file (XML) IP360 data can be acquired by ISM
- the ISM internal fork of vulnerability events to the custom escalation rules
- synthetic ISM vulnerability instance events escalate to prototype “JSS Listener” code

2.3.1.3.2 Direct JNDMS/JSS Integration

There are two main approaches by which nCircle IP360 could be directly integrated into JNDMS via the JSS. These include a “pull” based method by which the JSS could periodically poll the IP360 database, or a choice of slightly different “push” based methods where IP360 would send vulnerability instance data to a purpose built destination in the JSS.

2.3.1.3.2.1 Database Polling

For this approach the JSS would periodically poll the IP360 SiH database to extract new vulnerability instance records. As the SiH uses standard relational database technology for its back end storage, this could be achieved through extensions to the type of functionality that the JSS currently uses to interact with the JNDMS Data Warehouse (JDW).

A sample IP360 SiH database, based on MS SQL Server was provided by DND for use in experimentation and planning for JNDMS. This sample database contained scan information on over 1000 distinct IP addresses that existed in a lab environment during a period of their internal testing of the IP360 product.

No schema documentation for the database is currently available to the JNDMS project, however the full database dump is archived in the Ottawa JNDMS lab for inspection or experimentation as required. Also of note is that the current JNDMS database interface is to Oracle rather than MS SQL, however as these relational database interfaces are standardized the necessary modifications for this difference would presumably be small, or potentially IP360 could be configured to use an Oracle backend.

2.3.1.3.2.2 Direct Acquisition of Data by JSSClient

Currently JNDMS supports acquisition of vulnerability instance data through a JSSClient interface. This JSSClient interface is a java client application which takes a Nessus 2.0 formatted vulnerability scan report as an argument, and passes the vulnerability instance data to the JSS and on into the system. The JSSClient can be invoked in a number of ways (dependant on the type of data being submitted) but the data is passed either as attributes on the command line with the client invocation, or inside of the file referenced by name on the command line. In particular Nessus results are acquired through a “vuln scan” method of the JSSClient, taking the name of a formatted XML file as an argument.

A similar approach might be possible for IP360 scan results using a modified “vuln scan” method in the JSS Client. Presumably an appropriate XML schema definition file is available from nCircle, or could be reverse engineered from sample data, and would allow direct acquisition of vulnerability instances in the same way that the “vuln scan” method does for Nessus scan reports. The JSS Client could also implement the database polling method described above.

It is not known if a command line alerting function similar to the SNMP trap function exists in IP360, but if one exists that supports variable substitution it would be possible to send vulnerability instances directly through JSSClient requests. This could be modeled after the “SIM methods” (for example) which are used to escalate IDS alarms, since attributes of each vulnerability instance on a host would appear as a set of parameters for each JSSClient invocation.

2.3.1.3.3 JNDMS User Interface Changes

Based on our current understanding that the IP360 Vulnerability Instance data is functionally equivalent to the Nessus Vulnerability Instance data (and corresponding UI views) from a JNDMS perspective, changes to the JNDMS UI are unnecessary.

2.3.1.3.4 Interface Scalability

Work on this task is underway within the constraint that no product or detailed technical documentation was available during the investigation. Based on evidence at hand the IP360 product itself can easily scale to a DREnet size deployment. Questions as to how scalable the JNDMS system will be in a DREnet deployment with respect to Vulnerability Instance data are still under investigation.

2.3.2 Prototyping IP360

The options from the initial feasibility were reviewed and it was decided to build the primary prototype, and to plan on deploying with an update to the JSS client that is capable of polling the IP360 SIH database. This choice provides us with the most flexibility until the final IP360 hardware and licenses are available.

The IP360 database provided allowed us to model the tools schema. Figure 10 shows a partial schema extracted from the available IP360 database. The portion shown identifies the core tables and attributes that we would require to integrate with.

Figure 10: IP360 Partial Database Schema

The initial prototype updated the JSS client to provide it with the ability to periodically poll the IP360 database. The use of the JSS client for this polling mechanism is critical if the IP360 deployment is considered an operational tool by DREnet. In this case the project's ability to connect to the database from the core of JNDMS may be jeopardized, as well as our ability to deploy complex tools on the database host. The JSS Client allows a relatively simple integration tool to run on the SIH host and would be easier to audit and update if required.

The JSS Client will use the time stamps provided by IP360 to ensure an efficient polling mechanism. It will see if there are any new vulnerability audits performed since the last poll, and create the XML required by JNDMS to report on vulnerability scans. It was found that the information available from IP360 is very close, conceptually, to what is available from Nessus so that little has to be done within the core of JNDMS to support this tool.

One of the advantages noted in the feasibility study of having the SIM provide the integration point for JNDMS is that the SIM would be able to use the vulnerability information to better escalate events into alerts. This would be a significant source of information, however it was decided not to use this as the primary method for four reasons.

The first is that we have more direct control over all of the data when we have access to the schema. It may prove difficult to pass all required information through the SIM, even though the primary attributes may be available. The second reason is that we may have an easier time deploying the JSS client on an operational server than trying to determine the access controls required for ISM. The third reason is that even if ISM was to gain vulnerability information directly from IP360, there may very well be other sources of vulnerabilities available to JNDMS, such as through operation input. It would be preferable to make these available to the SIM as well. The final reason is that we can still use the native ability of ISM to ingest vulnerabilities. This is seen as a straight forward configuration and would still allow the SIM the ability to use vulnerability information from IP360 to escalate alerts. The last option provides us with the most information and the least risk.

2.4 Planning deployment and experiments of JNDMS on the DREnet

This task was to evaluate what was required to deploy JNDMS on the DREnet. There were a number of tasks that had to be undertaken to ensure that we knew the issues that we would face, the cost drivers as well as the start of discussions with DREnet network management.

The following tasks were undertaken:

- Drafting of a DREnet deployment document (DN0916).
- Meeting with DRDC network personnel.
- Drafting of a Data Sanitization Report (see Section 2.4.1).
- Updates and information on licenses required.
- Examining key integration points.
- Merging the results of all of the other tasks (Scalability, IP360, and GIS) with the findings of this task to determine a plan for deployment.

2.4.1 Data Sanitization

The JNDMS system will require sample data from the following sources to be exported for development:

- Spectrum (Network discovery)
- Snort (Intrusion detection)
- IP360 (Vulnerability assessment)
- Check Point FW-1 (Firewall policies)

The data will be retrieved from the DREnet production environment and then all IP addresses contained in the datasets will be sanitized via a prefix-preserving algorithm to be exported to a development facility. This data will be used for testing JNDMS scalability outside of the DREnet NCC where a consistent set of data is required for development purposes. All IP addresses will be extracted from the data files, and then be processed through the “anonymizing” routine. Each octet of the original IP address is transformed individually base on a pre-generated 256 bit key. This is done using the IP::Anonymous perl module available from CPAN. The transformed address will then replace the original address in all of the data files.

Sample IP Address transformations:

Original		Transformed
128.43.95.254	→	111.181.64.50
192.168.128.8	→	29.87.159.137
192.168.128.9	→	29.87.159.136
192.168.128.10	→	29.87.159.138
192.168.128.11	→	29.87.159.139

It must be noted that access to the information on DREnet, such as firewall policies, must be granted. The details of what access would be available and the restrictions on the resulting information are to be negotiated as part of the deployment efforts.

2.4.1.1 Anonymization Function and Implementation

The data will be sanitized using the Cryptography-based Prefix-preserving Anonymization function described referred to at:
http://www.cc.gatech.edu/computing/Networking/projects/cryptopan/

A formal description of this particular approach, including analysis of the security of the method can be found in xu02prefixpreserving.pdf (Prefix-Preserving IP Address Anonymization: Measurement-based Security Evaluation and a New Cryptography-based Scheme, Jun Xu, et al. 2002)

A perl implementation of this approach is available through CPAN (IP:Anonymous) and has been used for a prototype of the proposed JNDMS DREnet data sanitization method.

2.4.1.2 Prefix-Preserving Anonymization

2.4.1.2.1 Description of concept

Prefix-Preserving IP address Anonymization is formally described as having the property that if two original IP addresses share a k-bit prefix, their Anonymized mappings will also share a k-bit prefix.

In real terms what this means is that a given set of 255 addresses in a class C block (a /24 CIDR block), will be mapped to a given set of 255 addresses (in a different order) in a different class C block. More importantly *any* set of addresses contained in a given CIDR block of *any* size will remain in a CIDR block of the same size when mapped to a new set of addresses. Therefore two adjacent IP addresses (representing a /31 CIDR block – say, 1.1.1.1 and 1.1.1.2) will remain adjacent after the mapping (so perhaps 23.43.10.8 and 23.43.10.7). This holds true for whatever size of CIDR block is desired. Of particular interest, 4 contiguous class C networks (a /22 network) will end up as 4 contiguous class C networks after mapping. (See jumble_sample.xls)

2.4.1.2.2 Rationale for usage

It is important to preserve the original subnet prefix so that the topology of the network remains intact and continues to depict a valid representation of how the original network is defined after the information has been sanitized. Firewall rules must also be taken into account. Firewall policies define subnet-blocks, as well as individual hosts, within their rules, and this information must match up with the transformed network topology in order to correctly define zones and valid subnets.

In the case of a Firewall policy, where a subnet mask has been defined, the address range will be separated, and redefined in the transformed output. If the address range contains multiple contiguous Class C address blocks, the range will be broken down into separate Class C address blocks. If the address range is smaller than a Class C address (contained less than 255 addresses) the range will be separated into individual hosts.

Consider the following simple firewall policy rules:

SRC_IP	SRC_PORT	DST_IP	DST_PORT	RULE
128.43.112.0/22	ANY	24.119.29.12	443	ALLOW
128.43.2.0/29	ANY	72.110.12.209	80,443	ALLOW

After the transformation:

SRC_IP	SRC_PORT	DST_IP	DST_PORT	RULE
111.181.124.0/22	ANY	227.86.218.113	443	ALLOW
111.181.34.120/22	ANY	165.172.20.216	80,443	ALLOW

Without such a prefix-preserving scheme it would not be possible for the JNDMS system to do any computational analysis of network topology, vectoring safeguards (e.g. firewall rules), zone risk, or anything involving “subnets”. A strictly random IP address anonymization scheme (with a static mapping function that could be reused between data sets) would only allow correlation between single hosts – for example, an IDS alarm targeting an asset could still be correlated with a vulnerability scan of that asset. In this latter scenario, the usefulness of the data exports to JNDMS development and testing would be tremendously less.

2.4.1.3 Security Evaluation

A rigorous security evaluation of this Prefix-Preserving anonymization technique is in sections 3, 4, and 5 of xu02prefixpreserving.pdf. Relevant portions of this are summarized here. The attacks analyzed conservatively assume that the attacker has access to known plaintext (in this case a handful or more of real IP addresses and their mapping to anonymized IP addresses) and can do frequency analysis on anonymized data (i.e. that they have access to a significant sample of anonymized data and can count the frequency with which various IP addresses appear to make educated guesses about what real IP addresses they correspond to).

2.4.1.3.1 Cryptographic attacks

In the referenced paper care is taken to rigorously prove that the 256 bit AES key used to seed the transform function for the IP mappings is secure from cryptographic attacks. They do this by showing indistinguishability between this anonymity function and a random prefix preserving function.

Essentially this means that the entire mapping function used for the whole of the 32-bit address space would never be exposed via cryptographic analysis.

2.4.1.3.2 Semantic Attacks

“Semantic” attacks which assume that the attacker has, or can guess, some of the mappings using publically disclosed information about the DREnet infrastructure (say from our public DNS records for the web servers, mail servers and such) would allow an attacker to compromise, or partially compromise a portion of the “anonymity mapping”. The entropy property of any given section of the anonymity mapping varies from “region” of the IP address space to the next, based on the key used, so visually inspecting the mapping of a given “region” of IP addresses for several different seed keys would be a good idea – for example looking at how “random” the anonymized DREnet public addresses are for several keys, and choosing the key that produces the “best” output.

Regardless of the key chosen, frequently occurring addresses and/or publically disclosed addresses (i.e. “DNS addresses”) are relatively easy to “de-anonymize”. Additionally the more anonymized data the attacker has access to (in terms of known anonymous → real mappings and large data sets) the more of the puzzle can be pieced together.

Given this, even the anonymized data will still be somewhat sensitive in nature, and should not be widely distributed, and should be limited in the size of the data exported to only what is required for the task at hand.

2.4.2 Resources and Costs

This section discusses the tasks and estimates for effort required for deployment. At the top level the tasks are as follows:

- Project management. This task includes any activities to support, manage, track and report on the deployment effort.
- NIO SOC Activities. This task represents the setup and configuration of the NIO SOC, including support infrastructure and key integration points.
- System Preparation. This task represents changes that must be done to the JNDMS before deployment.
- System Deployment. This task represents the transition from updates and testing to a deployed system.
- System Support. This task represents support activity to aid DRDC in demonstrations or investigations.

The following table outlines the expected tasks and effort estimates:

Table 8: Estimated Effort Required for Deployment

WBS	Task	Estimate Effort (Days)	Notes
1	Deployment	720.5	
1.1	Project Management	146	This task represents all project management activities.
1.1.1	PM Staff	81	
1.1.1.1	PM	35	Estimate based on approximately 6 days per month with partial support in October and March.
1.1.1.2	PE	46	This task is the project engineering with time to oversee the technical management. (approximately 2 days per week).
1.1.2	PM Support	35	
1.1.2.1	Contracts	2	
1.1.2.2	QA	5	
1.1.2.3	ERT	4	External Review Team.
1.1.2.4	PMA	18	Accounting and project support services. Based on approximately 3 days / month
1.1.2.5	CM	2	Support, in addition to other lab support, for configuration management.
1.1.2.6	DM	4	Covers monthly reports and minutes (as required) only
1.1.3	Detailed Planning	21	
1.1.3.1	Kick Off Meeting	10	Assume trip to Ottawa

WBS	Task	Estimate Effort (Days)	Notes
1.1.3.2	Initial Planning	6	This task is to compare results of tasks completed in TA1 and verify priority and scheduling of TA2 tasks.
1.1.3.3	PMP Update	5	This task is to update the project management plan with the details of this task authorization.
1.1.4	Final Report	9	This is the authoring of the final report that documents the activities of this task authorization.
1.1.5	Travel	0	Travel based on 52 person days for kickoff meeting, initial deployment and ongoing support. The days are allocated directly to the tasks they support.
1.1.6	Misc	0	Shipping costs.
1.2	NIO SOC Activities	185.5	The activities within this task represent setting up the servers and workstations required for JNDMS, as well as the effort required to integrate into the network and ensure that the appropriate data is available to the system.
1.2.1	Infrastructure	17	
1.2.1.1	Network connectivity	0	This task is to ensure that the JNDMS demo centre (expected to be in NIO Building 94) is connected to the DREnet. This will include ensuring the fibre is configured to allow direct access to network resources. Assumptions: For planning purposes it is expected that required equipment, such as fibre media converters, have already been purchased and installed by CRC/DRDC personnel prior to the execution of this task authorization.

WBS	Task	Estimate Effort (Days)	Notes
1.2.1.2	Network configuration	3	This task includes: - Effort to configure main firewall for connectivity, such as modifications to firewall policy. This effort must include the time to identify, plan, approve, deploy and test. -Our team will have to identify our needs, then the DREnet management team would implement the changes. -This task would also include the effort for the LAN configuration in building 94. Assumptions: The effort for DRDC or the DREnet management team is covered under their normal duties.
1.2.1.3	Network coordination	14	This task is to provide support in the coordination of the various configurations and install tasks. This task provides some oversight and additional effort to meet with stake holders.
1.2.2	Servers	76.5	
1.2.2.1	Specify / procure	0	Assumptions: It is expected that the servers will be procured outside of this task to ensure that the required equipment is available near the beginning of this task authorization.
1.2.2.2	Server Base Install	22	This task will include the following: - Install in racks (shelves for towers) -Operating system install -System hardening (shutdown unnecessary services, enable automatic patching, removing unneeded accounts, tightening file/account permission) -Install anti-virus -Configuration of disk encryption Assumptions: - The system will be procured outside of this task authorization. -The operating system and anti-virus will be procured with the servers. Risks: - Disk encryption may have performance impact

WBS	Task	Estimate Effort (Days)	Notes
1.2.2.3	System Installs	35.5	
1.2.2.3.1	SIM (Intellitactics Security Manager) Install	11	This task must include: - renew of ISM lab license -install and configure -Set up of custom data acquisition integration such as have system fork inboxes and copy securely. Assumption: - Need clarification of lab license use.
1.2.2.3.10	Bugzilla Install	0.5	Bugzilla will be used to track and allocate issues during the deployment effort.
1.2.2.3.11	JNDMS Portal Install	1.5	This is to install the portal and ensure all JNDMS components are functioning. The portal must be configured to use SSL and certificate authentication.
1.2.2.3.2	NSM Install	2	This task is to configure NSM event console with other EIM tools such as Spectrum and to forward events to JNDMS. Assumptions: - The effort for CA to set up and support this task is covered separately and not as actual days on this task.
1.2.2.3.3	Spectrum Install	3	This task includes the following: - installation / configuration of spectrum -collecting initial baseline of topology -forwarding discovery and availability events to JNDMS -validating and testing the system Assumptions: - The effort for CA support is covered as a separate cost and not allocated as days to this task. -This task must also evaluate any issues with NAT devices that might prevent the scanning of required networks.
1.2.2.3.4	IP360 Admin Install	1	The IP360 administration is not likely to be a separate machine, just a software server on an existing server.

WBS	Task	Estimate Effort (Days)	Notes
1.2.2.3.5	Centennial Install	10	This will involve familiarization with the tools as well as coordination with the sites. Assumptions: - License and installation of Centennial is done by DRDC corporate. Only installation and configuration of JNDMS integration components required. -An existing deployment of Centennial already exists and this task is just to provide integration effort with JNDMS (development of new tools is part of the preparation tasks below). - This task must also evaluate any issues relating to NAT devices that might prevent collection of software inventory data.
1.2.2.3.6	Mapping Server Install	3	This task is to install and configure any required map services for JNDMS. This will include installing existing map caches and configuring any external map services from public sources or within DRDC. Assumptions: - ESRI will not be used as part of JNDMS. -If Google enterprise would be used existing licenses would be leveraged.
1.2.2.3.7	Database server Install	1	Install and configure Oracle.
1.2.2.3.8	Subversion Install	0.5	This task must provide a subversion repository that is backed up. Assumption: - Version 1.5 of subversion will be used.
1.2.2.3.9	Hudson Server Install	2	This is to install and configure Hudson to be able to build and test the deployments as well as test configurations.
1.2.2.4	Authentication	8	
1.2.2.4.1	Configure certificate authority	6	This task includes configuration of port to use certificate authorization as well as setting up the ACLs required.
1.2.2.4.2	Issue certificates	2	

WBS	Task	Estimate Effort (Days)	Notes
1.2.2.5	Validation of Server installs	11	Mini certification and accreditation This task is to address any concerns the individual sites may have. This is not a formal process of accreditation. Part of this task would likely include a vulnerability scan, however the specific process will be determined by the "system accreditation process" task. This task must also ensure that the implemented audit facilities are working correctly and tracking activity.
1.2.3	Appliances	23	
1.2.3.1	IP360	23	
1.2.3.1.1	Install / Configure IP360	15	This task includes familiarization, installation, and configuration of the IP360 tools. This will also include identification of target systems. This task must identify the firewall policies that must be in place. Assumptions: - The target systems will be less than 100 hosts and likely target internet exposed hosts only. - The IP360 appliances, servers, software and licenses have been procured outside of this task authorization.
1.2.3.1.2	Network / SIM integration of IP360	5	This task is to fork the data from IP360 to the SIMs and to JNDMS. It is expected that both data feeds will be used in parallel.
1.2.3.1.3	Test / Verify IP360	3	
1.2.4	Network configure	55	
1.2.4.1	Firewall policy configure	6	Alter or configure firewall policies for all tools required: - SIM / ISM - IP360 - EIM (NSM, Spectrum, Centennial) - JNDMS client tools

WBS	Task	Estimate Effort (Days)	Notes
1.2.4.2	Data routing configuration	13	Duplicate data feeds for security events, discovery events and availability events. This task must develop the process required to fork data feeds and to shunt data from DREnet to the JNDMS NOC with no queuing.
1.2.4.3	NIDS configure	0	Assumptions: - No Changes required.
1.2.4.4	Vulnerability Scanners configure	6	It is expected that the primary vulnerability scanner will be IP360, although the final disposition of Nessus on DREnet and the ability of Centennial to provide vulnerability assessments (Security Advisor) are still a possibility. This task provides a few days of effort as risk mitigation to configure additional feeds if required. This task does not include any effort to integrate new scanners.
1.2.4.5	Configure discovery roll up points	0	The currently deployment of Centennial on the DREnet has no roll up points and the product does not have the ability to configure a single roll up with using the existing configuration. The information from Centennial will be queried from each separate site. The deployment of Spectrum that is expected will done from a single server and will require no additional configuration.
1.2.4.6	System accreditation process	11	This task is to identify and document the accreditation process required for the servers and analyst stations for JNDMS. This accreditation process will be to address key issues but not to formulate a formal process.
1.2.4.7	Data Sanitization	19	This task is to implement and test the tools and procedures. Assumptions: - The prototype approach identified in TA1 is acceptable.
1.2.4.8	CA Support	0	CA support for trial software has been provided as a single value without breaking out specific resources.

WBS	Task	Estimate Effort (Days)	Notes
1.2.5	Analyst Stations	14	
1.2.5.1	Specify / procure analyst stations	0	Assumptions: - The analyst stations will be specified and procured with the servers, all in advance of this task authorization.
1.2.5.2	Base Install of analyst stations	6	The base install for the analyst stations include the following: - Setup in the demo room. -Network patch into NIO SOC LAN -Operating system install -Hardening (shutdown of unnecessary services, enable of automatic patching, remove unneeded accounts, and tightening of file/account permissions) -Install/configure of anti-virus Assumption: - The operating system and anti virus will be procured with the systems and not included in this task -Any required updates to the demonstration room, such as upgrading of the data projector, are not part of this task.
1.2.5.3	System Installs	6	This task includes the install and configuration of the analyst stations. - JNDMS Console (browser). The browser must be IE with Java installed. -Spectrum One Click Console - Centennial console - ISM Admin console - IP360 Admin console - Unicenter Console - Entrust Security Provider
1.2.5.4	Validation of analyst stations	2	Mini certification and accreditation
1.3	System Preparation	103	

WBS	Task	Estimate Effort (Days)	Notes
1.3.1	Lab setup	6	This task includes ensuring the Hudson build system can build and deploy the system. This task must also include the effort to upgrade the JNDMS lab subversion repository to v1.5. This is expected to ease the patch management between separate repositories. Assumptions: - A separate subversion repository will be required on the DREnet. A procedure for migrating paths between repositories will have to be used.
1.3.2	System Updates	53	
1.3.2.1	Audit	12	It is expected that the ability to audit a user's activity will be required. This task is to ensure that key components such as the portal are configured to track or log significant activity.
1.3.2.2	Initial Issue Resolution	41	The initial issue resolution represents tasks that should occur before the primary deployment. These tasks will also be influenced by early integration and testing efforts to ensure that high priority tasks have been addressed before the initial deployment. Assumptions: The tasks identified for TA1 have been completed.
1.3.2.2.1	Portal	5	Continue key updates to query optimization and page views
1.3.2.2.3	JSS	4	

WBS	Task	Estimate Effort (Days)	Notes
1.3.2.2.3.1	Location inference	4	The location information in JNDMS is generally entered as part of the operation data, however in most instances a new IP address will exist in the same location as the rest of the subnet. The network subnets are generally strongly related to the physical locations. To ease the assignment of location information for the much larger networks this task would add the ability to infer the location of the newly reported asset based on the location of other assets in the same subnet or zone. For example we could see if any assets in the same subnet have a different location, and if they don't is should be reasonable to assume that the location of the new asset is the same. This ability should be able to be turned off if required.
1.3.2.2.4	DSS	32	
1.3.2.2.4.1	DSS Modularization	27	The DSS currently runs in a single thread separate from all other processing. The DSS itself performs several functions, many of which are independent. For example the correlation and risk calculations do not directly depend on each other. This task would be to modularize the DSS such that database access, risk calculations, correlation, impact assessment, etc, can all be done in a modular fashion and many of the tasks in parallel. This task would depend on the database optimization task being done so that the in memory models can be leveraged to ease the modularization. Another advantage of this task would be in future development and in risk mitigation. A more modular DSS would ease integrating additional processing or alternate algorithms. For example an alternate impact assessment would be easier to implement on a more modular DSS.

WBS	Task	Estimate Effort (Days)	Notes
1.3.2.2.4.2	DSS Algorithm optimization	5	There are a number of areas that can be optimized by reviewing processing loops. This task provides some effort to evaluate bottlenecks identified by the scalability study. Small changes done during scalability investigations for this task have already provided about 50% performance improvement in some areas.
1.3.3	System Integration	30	
1.3.3.1	IP360 Integration	15	The initial prototype of the IP360 integration is expected to be done as part of TA1. This task is to assess that effort and to ensure that full integration occurs with the deployed IP360.
1.3.3.2	Centennial Integration	15	This task is to provide integration with existing Centennial tools and deployments. Assumptions: - Key JNDMS attributes are available and that Centennial provides adequate integration points.
1.3.4	Scenario Validation	5	This task is to run through scenarios developed during TA1 to ensure that the new deployment conforms to expected performance and functional expectations.
1.3.5	System readiness review	9	This task is to review the status of the system and assess its readiness for deployment. The priority of the initial system updates as well as the scenario validation are important inputs to this decision. If additional actions must be taken the effort would be taken from the system support task.
1.4	System Deployment	55	
1.4.1	Test rollout	17	The test rollout will provide a rollout to verify all system readiness activities.
1.4.1.1	Verify audit / hardening	11	
1.4.1.2	Initial deployment	6	

WBS	Task	Estimate Effort (Days)	Notes
1.4.2	CND Data	31	This task is to provide the operational and infrastructure context that would not be automatically discovered on the network.
1.4.2.1	Operation data	11	
1.4.2.2	Service data	11	
1.4.2.3	Safeguard data	9	
1.4.3	Deployment review	7	
1.5	System Support	231	
1.5.1	Scenario updates	18	This task is to provide additional support in the form of general support, updates to scenarios, etc. The effort identified would be used as required.
1.5.2	Scalability Issues	122	The scalability issues noted in this task were coordinated with the priority tasks identified in TA1. During the detailed planning of TA2 these issues will again be reviewed to align with the accomplishments of TA2 and to update their priorities.
1.5.2.1	Portal	34	
1.5.2.1.1	Data Lists	0	This task would include: - Change drop downs to ajax updates to improve page load responsiveness -Update list contents via ajax to further improve page load responsiveness. The current scalability study has shown that this is more than just a performance issue, with larger data sets prevent pages from loading all together. Using the filter to limit the amount of data has allowed pages to load. Assumptions: - This task was scheduled in TA1 and should just be part of the planning and prioritization review.

WBS	Task	Estimate Effort (Days)	Notes
1.5.2.1.2	Portal SQL Query optimization	6	This task would include a few days to identify top queries, then several hours per query to optimize. Assumption: - The scalability task in TA1 provides sufficient insight to identify query bottlenecks.
1.5.2.1.3	Portal Database connection optimization	5	The use and maintenance of connections to the database can be improved by the use of connection pooling software similar to the way the JSS manages connections.
1.5.2.1.4	Navigation List Optimization	5	Some of the lists contain data in tables that is itemized by location. These could be slow and may be improved with Ajax updates.
1.5.2.1.5	Portal System upgrades	3	There is a significant update to Tomcat (v6.0) available as well as a new update to the Liferay portal. It has been identified that the newer Tomcat may provide performance improvements during initialization and during heavy loads. The new Tomcat can also be easier to maintain. An optional part of this task (time permitting) would be to migrate the data store of Liferay into the Oracle database. The current Liferay installation uses HSQL and the use of a single data store for JNDMS would ease maintenance and possibly perform better under high loads. Assumptions: - Part of this task was completed under TA1, with additional effort required for look and feel and verification of the upgrade.
1.5.2.1.6	General Navigation or view updates	15	There are several areas in which updates to the display would be beneficial. This would include: - Updates to the event views to hide child incidents. -Tweaks to the navigation based on usability. This would include a few updates to views to facilitate the creation or editing of relationships from additional details pages. This task will provide additional time to prioritize and implement key changes.

WBS	Task	Estimate Effort (Days)	Notes
1.5.2.2	Mapping	18	
1.5.2.2.1	Map Query optimization	5	This task will evaluate and update database queries for the mapping code. This will also include evaluating connection pooling. Assumption: - This task assumes that the GIS task of TA1 has completed and migrated the mapping component to a more modular approach.
1.5.2.2.2	Thread Safety	3	There have been some issues noted with thread safety within the mapping core that may cause issues as the processing load scales. Assumptions: - This task assumes that the GIS task of TA1 has modified the mapping core which will make the code more manageable.
1.5.2.2.3	Map integration issues	10	The mapping subsystem is expected to be upgraded as part of the first task authorization. This task is to address issues related to a significant upgrade just before integration.
1.5.2.3	Topology	21	
1.5.2.3.3	Updates to visual cues (topo)	2	
1.5.2.3.4	Layout algorithm improvements (topo)	10	They layout of very large dataset can be an issue. This task can address some issues, however this is not a major overhaul to the layout. Some tasks to consider include: - Ensuring the dynamic loading doesn't have too many assets on the screen at one time. -Allowing the operator the ability provide hints or clues to the layout, such as selecting an area. -Improve grouping of icons. Assumptions: - This will be investigated during TA1 to determine what can be done within the allocated scope.

WBS	Task	Estimate Effort (Days)	Notes
1.5.2.3.5	Additional views (topo)	9	This task is to provide some support for minor updates to the views.
1.5.2.4	JSS	0	The effort expected for the JSS, as a component, is addressed as part of other tasks to facilitate integration.
1.5.2.5	DSS	49	
1.5.2.5.1	Correlation updates	8	The current implementation has two forms of correlation. The first represents cause and effect and is identified as a parent/child relationship in the portal. The other type of correlation (coincident) is more general and is noted as a percentage of correlation. It has been noted that, especially as the number of events grow, these two views are related. This provided two possible changes: - Allow the user to 'fuse' the events to create the parent/child relationship -If the coincident correlation is strong enough the events should be fused automatically. This task should also integrate the time of events in the correlation scheme.

WBS	Task	Estimate Effort (Days)	Notes
1.5.2.5.2	Correlation optimization, Risk updates	7	The scalability study has shown that one of the bottlenecks in the DSS is the correlation calculations. An initial review has identified the database handling as a key issue (addressed in the 'database optimization task') but also noted that additional effort directly on the correlation calculation can be done. The effort allocated for the 'modularization' task will also likely impact the performance of the correlation. As part of this task the ability to defer the calculation will be looked at. This may involve waiting until a request is made or setting up a timed event. It is, however, expected, that the database and modularization efforts combined with the algorithm review will be sufficient. This task would include the effort to use JGraphT to create an in memory model of the incidents and events. This would allow not only for decreased database access but also for faster processing. The current investigations into this task have noted that our current processing is in the order of O(n^2) with respect to the number of incidents so as the number of incidents grow the problem gets exponentially worse. The algorithm may be reviewed to assess the possibility, for example, of only correlating the 'aggregated' events instead of all events. This would be more beneficial in conjunction with the 'correlation updates' task and the 'ISM optimization' task. This task will also include updates to risk assessment.

WBS	Task	Estimate Effort (Days)	Notes
1.5.2.5.3	DSS Database optimization	10	It has been identified that database interaction is causing several issues with the DSS. These issues include long query times and access of the database at inappropriate times such as in the middle of processing. This task would involve the following: - Review and optimization of key queries -Ensuring connection pooling is properly done in all instances -Maintain more information in memory so that queries do not have to be done as often. Database queries and updates should be confined to initialization and clean up routines. -Effort to ensure that all in-memory information is consistent with the database. Risk: - Moving more information into in memory data structures may cause memory issues, however initial investigations show that this is manageable.
1.5.2.5.4	ISM Optimization	8	The identification of the alert propagation through ISM is to be completed here.
1.5.2.5.5	Vulnerability Definition/Instance Optimization	8	The scalability effort so far has identified issues with the handling of large numbers of definitions and instances. This task would be to eliminate the current bottlenecks.

WBS	Task	Estimate Effort (Days)	Notes
1.5.2.5.6	Vulnerability scan updates and patches	8	One of the issues identified would be in dealing with patches and what vulnerabilities are covered. Currently identified data sources do not readily provide this information and tool vendors such as nCircle or Symantec consider this information proprietary. Each of these companies will maintain their own mapping of patches/fixes to vulnerabilities to try and provide a competitive edge for their respective tools. Currently the CVE data feeds provide information about patches in the reference section of each reported vulnerability. These are processed by JNDMS and included in the 'reference' and 'safeguard' sections of the vulnerability, however JNDMS can't automate the processing and application of these patches. The information on how these patches are applied or how they might be reported by asset inventory tools is not readily available. To address this issue during deployment a number of things should be done: - Update vendor and product information to conform to the CPE standard. This is the closest thing the industry has as a standard. -Draft common procedures to manually updating patch/fix information. This manual process should allow for common or high priority fixes to be addressed by the operators. -The credibility of the source should be considered when integrating results from multiple scanners. In this case we can identify which sources are preferred when assessing what the current list of vulnerability instances are active. This will allow tools that have access to detailed mapping to override internal calculations or identification by other tools that may not have the same level of insight.
1.5.3	General Issue resolution	91	The issue resolution task will track issues experienced during the deployment and demonstration efforts. It is expected that other tasks identified in the scalability efforts will be ongoing during this time as well and Bugzilla can be used to prioritize issues. This task includes general time over and above other tasks identified and will be used as required.

In addition to the labour costs involved there would be GFE required and it is expected that an update to the Intellitactics lab license would be required and that some travel would be required. The details of the exact costs for labour and non labour components are provided in the proposal for Task Authorization #2 [R3].

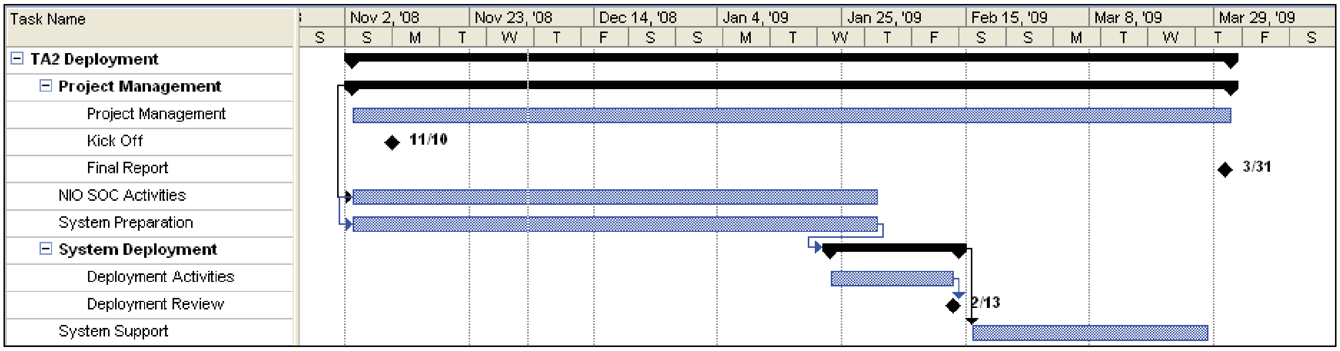
In addition to the existing equipment the following is also required:

- Core JNDMS servers for deployment (estimated at 8 servers) with operating systems installed
- Rack space and connectivity for deployed setup
- Access to data feeds and licenses for the following (access required to complete integration efforts and testing as well as deployed system):
    - IP360 (appliances purchased as part of TA #1)
    - Intellitactics data stream duplicated from existing ISM installs
    - Centennial discovery and asset information
    - Spectrum discovery information

2.4.3 Schedule

The following outlines key schedule milestones, issues or constraints:

Task/Milestone	Date	Notes
Task start	Nov 1, 2008	This task must start in very early November.
Lab readiness	Dec 15, 2008	Access to all expected hardware and software will be required early on so that initial issues with integration and policy can be mitigated.
NIO SOC Activities	Jan 30, 2009	The represents the point when all of the SOC integration efforts and setup have been complete enough for deployment.
System Preparation	Jan 30, 2009	The represents when any required updates for deployment to the JNDMS have been done.
System Deployment	Feb 13, 2009	This is when the system will be deployed.
System Support	March 31, 2009	This is the end of the support activities.



Task Name
⊟ **TA2 Deployment**
⊟ **Project Management**
Project Management
Kick Off
Final Report
NIO SOC Activities
System Preparation
⊟ **System Deployment**
Deployment Activities
Deployment Review
System Support

2.4.4 Assumptions

During the execution of Task Authorization #1 MDA has developed a number of assumptions that are important to the successful outcome of our plan deployment:

- Access to GFE is required by end of November.
  - JNDMS servers and connectivity
  - Licenses and software for Centennial and IP360
  - Access to key data feeds including security and infrastructure events
- Support from DRDC and DREnet personnel on as required/when needed basis
- The majority of work can be done in Halifax and synchronized with on-site lab to minimize travel.
- Spectrum can cover required scope of network from a single server. This is based on recommendations from CA as well as Spectrum's ability to be flexible in its configuration.
- Spectrum's current reporting path through NSM will be sufficient.
- The data sanitization approach presented will be acceptable to DREnet.

2.4.5 Risks

Risk	Mitigation	Impact	Probability
JNDMS System issues interfere with deployment	Effort allocated to areas where issues are expected.	High	Medium
Support from DRDC corporate is lacking	Work closely with DRDC corporate through NRNS and NIO group	High	Medium
Support from DREnet sites is lacking such that we cannot deploy	Work closely with sites. This will also be mitigated once support from DRDC corporate is clear.	Medium	Medium
IP360 not procured outside of TA or not available in a timely manner	The appliance has been procured as part of TA1 with DRDC procuring the licenses directly.	Medium	Low

Risk	Mitigation	Impact	Probability
Centennial deployment does not go ahead	Follow deployment plans closely and adjust preparation and deployment work of task accordingly. It is expected that enough deployment has already occurred that the core JNDMS tasks can continue.	Medium	Low
COTS complexity causes additional effort during integration	Early prototypes are done in TA1 Reliance on any single COTS is minimized Effort set aside for general issue resolution.	Medium	Medium
General scope creep occurs	Work closely with DRDC-PM and set clear expectations on each task	Low	Medium
Servers not procured outside of TA or not available in a timely manner	The procurement is in progress, however the delivery date is not known.	Medium	Medium

2.5 GIS Component Updates

The original goal of the GIS task was to add additional layers to the existing ESRI components, however it was noted that significantly better components exists that we could port the JNDMS views over to. The result was a request from DRDC for us to provide additional information. The information found in Section 2.5.1 was provided. The results of the task can be found in Section 2.5.2

2.5.1 GIS Task Update

This document describes implementation and integration options for improving the mapping related features of the JNDMS platform.

2.5.1.1 Some Driving Goals

The primary goal of this task is to improve the usability of the mapping component within JNDMS and to reduce the maintenance required. The following goals are expected benefits of this refactoring effort:

1. Enable wider interoperability of JNDMS entity geo data with third party mapping platforms.
2. Improve performance of default integrated map client user interface.
3. Improve overall ease/speed of use of mapping interface.
4. Maintain at a minimum the existing conceptual mapping requirements of JNDMS.
5. Expose mapping data using industry standard protocols where possible.
6. Minimize proprietary interfaces and protocols used in JNDMS.
7. Maximize code reuse by leveraging detail-view data access layer query generation (generic SQL) when accessing mapping data.
8. Provide an option(s) for displaying geo data in 2D or 3D view.
9. Add flexibility of controlling map layers visualization
10. Maintain the ability to not require a public internet connection to display map data
11. Reduce licensing costs required to operate JNDMS.
12. Reduce specialized skill base (ESRI related) required to add mapping related features.
13. Add support by default to integrate map layers from third party sources in standard formats

2.5.1.2 High level Implementation Strategy

The mapping client currently has three discrete components, the client side mapping component, a server side web application and a map server. These three components work together to provide the GIS capabilities within JNDMS.

Our implementation strategy is to maintain this basic architecture; however the responsibilities of the client and the web application components will be significantly changed. The core change is to have the server side web application provide as much information as possible in a standard protocol, and to drastically reduce the processing requirements and customization on the client side.

We have chosen Google's Keyhole Markup Language (KML) as the mechanism for the server to expose JNDMS specific content. The KML standard is becoming more widely used and provides a fairly rich set of primitives to communicate the geographical information required.

The use of KML that is annotated with not only the icons and positions but also the links and data for the popups results in a much simplified client. Our goal is to use a client that natively processes KML to minimize or even eliminate customization of the client. Another benefit of this approach is that the effort is put into describing the JNDMS views in KML that additional clients could be used as well. For example our default view within the JNDMS portal may use the OpenLayers client but Google Earth could be set to download the same KML and view the same content within its own interface.

The map server itself can be very generic with several options available. Our initial implementation would not use a custom map server, but rely on third party WMS map servers. There are several available that provide base maps including basic geographical boundaries or satellite imagery. Overlays or partial transparent maps are also available through WMS map servers.

Our initial implementation would also use TileCache to provide a view of readily available maps. This tool can cache requests to WMS servers and also provide a 'seed' capability. This allows map information to be stored locally or within the JNDMS lab environment to reduce or eliminate the requirements for network access at all times.

2.5.1.3 Task List and Resource Allocation

The following table shows a break down of tasks and estimated effort. This task was designed as a replacement for the current GIS task and will not require additional effort to execute.

Task	Estimated Effort (days)	Notes
KML adaptor	12	This task includes the creation of a KML adapter that will translate JNDMS entities, including active filters, into Google's KML (XML) format. This task includes server side activities such as integration and connection management. This task will include providing the icons for key entities such as vulnerabilities, locations, events or operations as layers within KML. This task will also include providing the links between entities described in KML.
OpenLayers Client	16	This task includes embedding the OpenLayers client within the portal and providing widgets to show the JNDMS layers (events, operations, etc) as well as available base maps and any available WMS layers. This task must ensure that navigation using the portal works, that filtered information is not shown, that the user can interact with the icons through popups and that the views are responsive.

Task	Estimated Effort (days)	Notes
Server side components	10	This task will include the setup of TileCache, choosing appropriate WMS sources for base maps and layers and evaluating the need for a MapServer installation.
Transition from ESRI	6	This will include migrating existing ESRI components where possible, removing old views and ensuring core functionality is preserved or improved upon in new version.
Design and component updates	25	This includes any effort to the build environment, testing and prototyping to finalize all of the above components.

2.5.1.4 Deliverables

This task will provide the working components to demonstrate the updated mapping client within the JNDMS portal as well as input to the design document describing how the new mapping functionality is integrated.

2.5.1.5 Assumptions

We have examined available WMS sources of geographical information and have assumed that existing, freely available sources, will be sufficient to reproduce or exceed the existing features found in JNDMS. Future sources of maps, such as Google Enterprise, can be tested in follow on tasks when available.

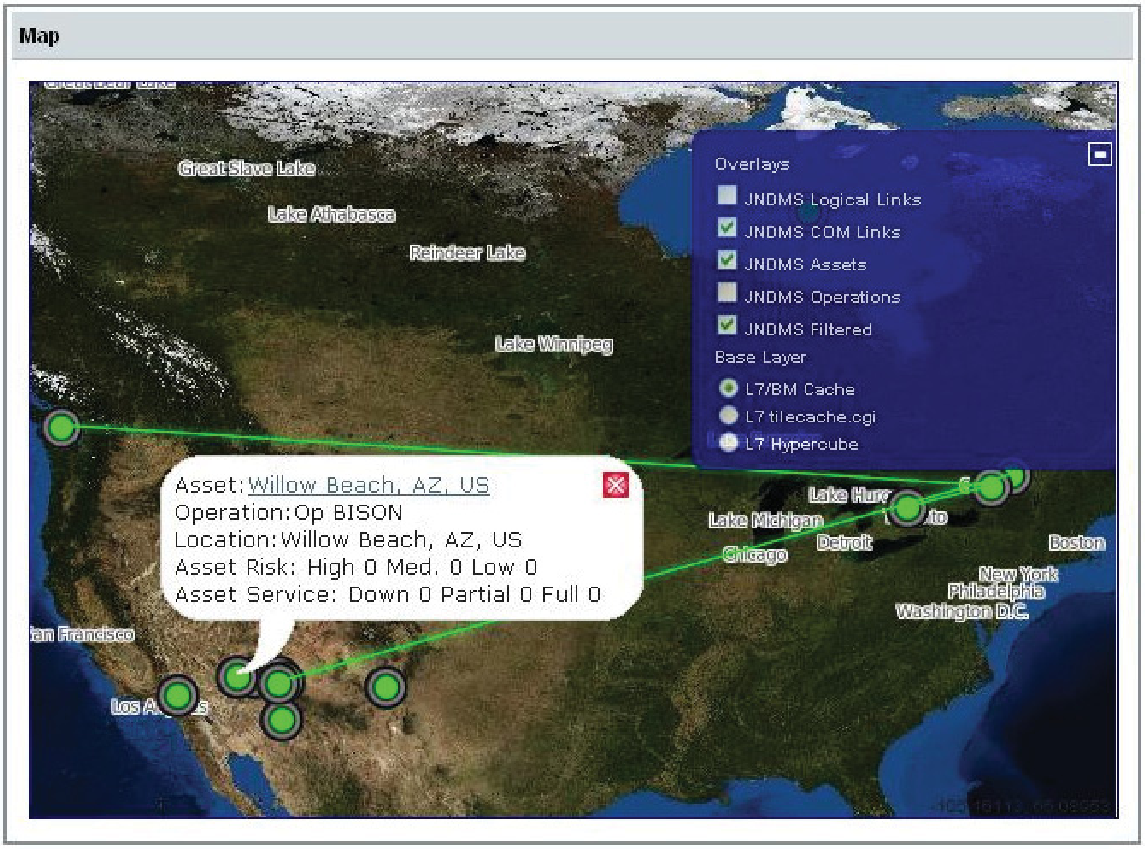
2.5.2 GIS Implementation

The goal of the GIS tasks was to replicate the functionality found in the current JNDMS GIS components, by implementing new components that would significantly increase performance.

The following tasks have been completed:

- ESRI code, references and dependencies have been removed
- OpenLayers 2D client has replaced ESRI 2 D Client
- Tilecache.cgi setup established for accessing remote WMS services
- Local network cache established for OpenStreetMap Overlays and World Topobathy Base Maps via TileCache seeding
- Google Base maps and other Mercator projection(i.e. Commercial) mapping base layers have been tested and integrated
- Ability to add custom map layers through configuration implemented
- Map queries are now derived from the same backend mechanisms as Topology and Detail JNDMS views, Generic SQL
- Layers dependent on Page view integrated (Filtered, and Com Links)
- Layers independent of Page view integrated (All Operations, All Assets)
- Inherent limitation of OpenLayers 2D client of click events not falling through past the top most layer has now been worked around
- Map data layer is now entirely KML driven
- Map place marks and vertices rendering is entirely KML driven
- Some previously slow map SQL queries have been optimized
- Upgraded OpenLayers to 2.7 taking advantage of feature and performance benefits

The GIS components have been replaced with an Open Layers implementation. Figure 11 shows the component within JNDMS including support of overlays, alternate base layers, pop ups and links.



Map

Overlays

- □ JNDMS Logical Links
- ☑ JNDMS COM Links
- ☑ JNDMS Assets
- □ JNDMS Operations
- ☑ JNDMS Filtered

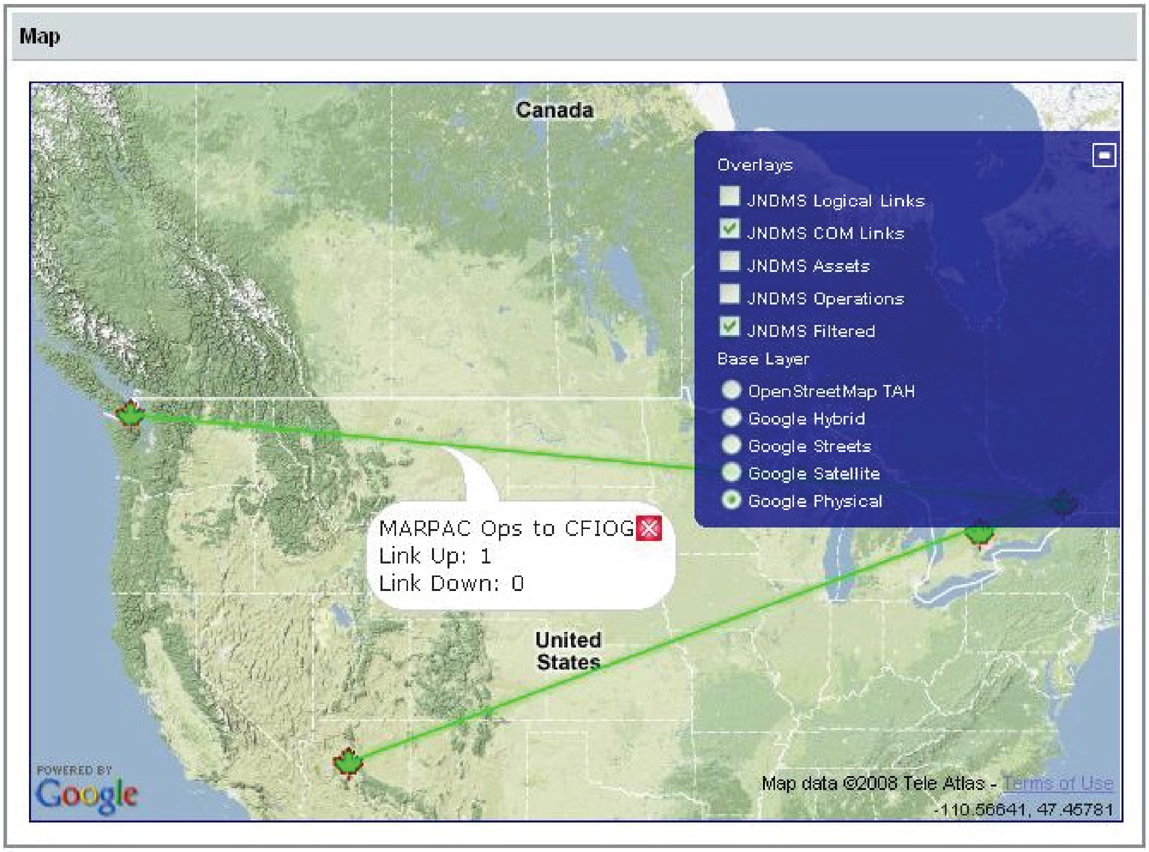
Base Layer

- ● L7/BM Cache
- ○ L7 tilecache.cgi
- ○ L7 Hypercube

Asset: Willow Beach, AZ, US
Operation: Op BISON
Location: Willow Beach, AZ, US
Asset Risk: High 0 Med. 0 Low 0
Asset Service: Down 0 Partial 0 Full 0

Figure 11: Updated GIS Component.

The new GIS implementation allows alternate base layers to be used from different sources. Figure 12 shows the base maps being provided by Google.



Map

Overlays

☐ JNDMS Logical Links
☑ JNDMS COM Links
☐ JNDMS Assets
☐ JNDMS Operations
☑ JNDMS Filtered

Base Layer

○ OpenStreetMap TAH
○ Google Hybrid
○ Google Streets
○ Google Satellite
● Google Physical

MARPAC Ops to CFIOG
Link Up: 1
Link Down: 0

Figure 12: JNDMS Using Google base map

2.6 Summary

The execution of TA1 activities resulted in significant improvements in the ability of JNDMS to scale to more realistic network environments and also resulted in updates to integration tools and methods that are expected to be required for deployment demonstrations.

The results of each of the tasks have been outlined above and the recommendations of how to carry the effort forward to be ready for a test deployment has been provided in section 2.4.2 (Resources and Costs for DREnet deployment).