

995015D

**Automated Computer Network Defence Technology
Demonstration Project Architectural Design Document**

for the
Automated Computer Network Defence (ARMOUR)
Technology Demonstration (TD) Contract
Contract No. W7714-115274/001/SV
ARMOUR DID No. SD002

Prepared For:

Defence Research & Development Canada (DRDC) - Ottawa
3701 Carling Avenue
Ottawa, Ontario K1A 0Z4

Prepared By:

GENERAL DYNAMICS
Canada
General Dynamics Canada Ltd.
Land and Joint Solutions
1020-68th Avenue N.E.
Calgary, Alberta T2E 8P2

16 December 2014

Contract Scientific Authority: Nacer Abdellaoui, DRDC – Ottawa Research Centre, (613) 998-4582

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.

Defence Research and Development Canada

Contract Report

DRDC-RDDC-2015-C006

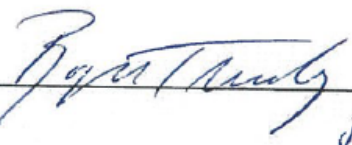
January 2015

© Her Majesty the Queen in Right of Canada, as represented by the Minister of National Defence, 2014

© Sa Majesté la Reine (en droit du Canada), telle que représentée par le ministre de la Défense nationale, 2014

GENERAL DYNAMICS CANADA LTD.
LAND AND JOINT SOLUTIONS

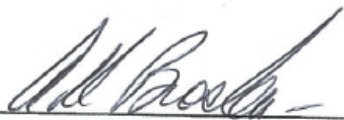
ARMOUR Technology Demonstration Contract

Author  for M. Rossiter

15 Jan / 15
Date

Lead System of Systems Architect  R. Tremblay

15 Jan / 15
Date

ARMOUR Project Manager  A. Brosha

15 Jan 2015
Date

Quality Specialist, Quality Management  J. Ko

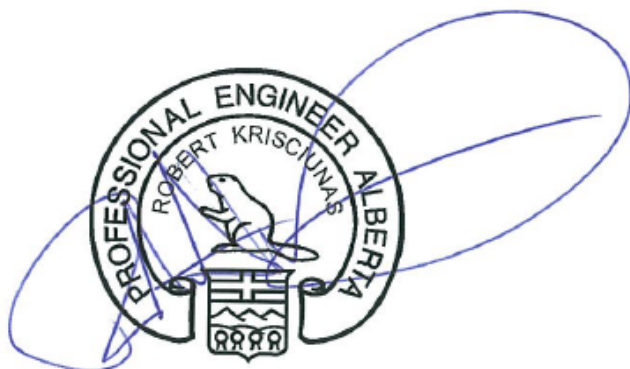
15 Jan 15
Date

Commercial Manager  K. Rokne

19 Jan '15
Date

Use or disclosure of this data is subject to the restriction on the title page of this document.

GD Canada Land and Joint Solutions Permit to Practice Number P06579



15 JAN 2015

Date

Use or disclosure of this data is subject to the restriction on the title page of this document.

REVISION SHEET

DOCUMENT NO.	VERSION	DATE	COMMENTS
995015	–	25 September 2013	Initial release.
995015	A	11 February 2014	Updated to address comments from DRDC. Revision bars () appear in the right margin to indicate changes from the previous version
995015	B	13 March 2014	Addresses comments from DRDC for formal acceptance of the Phase I artifact. Revision bars () appear in the right margin to indicate changes from the previous version.
995015	C	07 August 2014	Updated to address Phase 3 Architecture as well as Phase 2 comments from DRDC. Document incorporates CCB-approved ICR #12804. Revision bars () appear in the right margin to indicate changes from the previous version.
995015	D	16 December 2014	Updated to address DRDC feedback received at the Phase 3 Readiness Review meeting for the Phase 3 ADD Submission. Document incorporates CCB-approved ICR #12965. Revision bars () appear in the right margin to indicate changes from the previous version.

Use or disclosure of this data is subject to the restriction on the title page of this document.

This page is left blank intentionally.

Use or disclosure of this data is subject to the restriction on the title page of this document.

TABLE OF CONTENTS

1.	ARMOUR HIGH-LEVEL ARCHITECTURE	1
1.1	Changes Introduced in this Version	9
2.	APPLICABLE DOCUMENTS.....	10
2.1	Government Documents	10
2.2	Non-Government Documents	10
2.3	Works Cited.....	10
3.	ARMOUR ARCHITECTURE.....	11
3.1	Architectural Concepts.....	11
3.1.1	Integration Framework.....	11
3.1.2	Data Model.....	16
3.1.2.1	Selective Rollback Approaches.....	18
3.1.2.2	ARMOUR Phase 3 Data Model Coverage	20
3.1.2.3	CND Dataset Grouping.....	23
3.1.2.3.1	Asset Dataset.....	24
3.1.2.3.2	Known Software, Installed Software and Running Applications Dataset	24
3.1.2.3.3	Network Topology Dataset	25
3.1.2.3.4	Vulnerability Assessment Dataset.....	25
3.1.2.3.5	Dependencies Dataset	26
3.1.2.3.6	Event Dataset	26
3.1.2.4	Computational Services Dataset Grouping	26
3.1.2.5	Application Support Dataset Grouping.....	27
3.1.3	Risk Treatment	28
3.1.4	Data Presentation	32
3.2	ARMOUR Phase 3 Logical Architecture.....	34
3.3	ARMOUR Technology	36
3.3.1	Technology Justification	36
3.3.1.1	Rapid Technology Integration Framework	36
3.3.1.2	SMARTHawk	37
3.3.1.3	Ozone Widget Framework for Graphical User Interfaces	39
3.3.1.4	PostgreSQL Database	40
3.3.1.5	SNORT IDS	42

Use or disclosure of this data is subject to the restriction on the title page of this document.

TABLE OF CONTENTS (cont'd)

3.3.1.6	MulVAL	44
3.3.1.7	AssetRank	45
3.3.1.8	COADS	46
3.3.1.9	National Vulnerability Database	48
3.3.1.10	Data Source Connectors	48
3.3.1.11	Barnyard2	49
3.3.1.12	WSO2 Balana	51
3.3.1.13	Effector Connectors	51
3.3.2	Technology Mapping	52
3.3.3	Technologies Under Evaluation	54
3.3.3.1	OSSEC	54
3.3.3.2	Cauldron (Jajodia & Noel, 2009) (Jajodia, Noel, Kalapa, Albanese, & Williams, 2011)	55
3.3.3.3	Nmap – Network Mapper	56
3.3.3.4	SCCM – System Center Configuration Manager	58
3.4	ARMOUR Subsystems	60
3.4.1	Data Source Connectors	60
3.4.1.1	Infrastructure Data	61
3.4.1.2	Security Data	63
3.4.1.3	Non-Infrastructure Data	64
3.4.1.4	Operations Data	66
3.4.2	Data Storage	67
3.4.3	Data Presentation	68
3.4.3.1	Data Presentation Interfaces	70
3.4.3.2	Data Presentation View Capabilities – Graphical Displays	71
3.4.3.2.1	Object Representation	71
3.4.3.3	Data Presentation View Capabilities – Tabular Displays	72
3.4.3.3.1	Tabular Details	73
3.4.3.4	ARMOUR TD Views	73
3.4.3.4.1	Infrastructure View	73
3.4.3.4.1.1	Topology, Inventory and Node Details Widgets	74
3.4.3.4.1.2	Host Configuration List Widget	75
3.4.3.4.1.3	Host History List Widget	75
3.4.3.4.1.4	Link Details List (Reachability) Widget	75

Use or disclosure of this data is subject to the restriction on the title page of this document.

TABLE OF CONTENTS (cont'd)

3.4.3.4.1.5	Incident, Incident Details and Alert Toolbar	76
3.4.3.4.1.6	Vulnerability Details Widget.....	77
3.4.3.4.2	Operational View	78
3.4.3.4.2.1	Command View.....	79
3.4.3.4.2.2	Operations View.....	80
3.4.3.4.2.3	Operational Data Input View	80
3.4.3.4.3	Security Action Status View	81
3.4.3.4.4	Attack Path View	82
3.4.3.4.5	COA View	82
3.4.3.4.6	Incident Analysis View	83
3.4.3.4.7	Supporting Views.....	83
3.4.3.4.7.1	Verification Policy Input	84
3.4.3.4.7.2	Work Flow Ticket System.....	84
3.4.3.4.7.3	Administrator Views	85
3.4.3.4.7.4	Report Views.....	89
3.4.4	Computational Services	90
3.4.4.1	Data Normalization.....	90
3.4.4.2	Cross Source Correlation	91
3.4.4.3	Common Infrastructure Abstraction	92
3.4.4.4	Data Analysis and Action.....	94
3.4.4.4.1	Reachability Analyzer	94
3.4.4.4.2	Operations and Infrastructure Analyzer	96
3.4.4.4.3	Proactive and Reactive Attack Graph Generator	100
3.4.4.4.4	Proactive and Reactive Attack Graph Analyzer.....	101
3.4.4.4.5	Incident Analyzer	102
3.4.4.4.6	Course of Action Analyzer.....	103
3.4.4.4.7	Semi-Automated Response	106
3.4.4.4.8	Automated Response	109
3.4.5	Effector Connectors	111
4.	ARMOUR SECURITY ARCHITECTURE	113
4.1	One-Way Data Diode	115
4.2	Preventative Measures for Software Modules	116

Use or disclosure of this data is subject to the restriction on the title page of this document.

TABLE OF CONTENTS (cont'd)

5.	PERFORMANCE	117
6.	NOTES	118
6.1	Abbreviations	118

FIGURES

1	The ARMOUR System Enclave	2
2	OODA Loop	4
3	High-level Architecture for the ARMOUR System	6
4	ARMOUR IF within ARMOUR Architecture	15
5	Selective Rollback: Snapshot Approach	18
6	Selective Rollback: Reverse Operation Approach	19
7	High-Level ARMOUR Data Model	22
8	ARMOUR UML Class Diagram	23
9	Risk Treatment within ARMOUR Architecture	31
10	Data Presentation within ARMOUR	33
11	ARMOUR System Phase 3 Logical Architecture	35
12	Infrastructure View	75
13	Incident Summary Widget	76
14	Incident Details Widget	77
15	Alert Toolbar	77
16	Sample Command View Wireframe	79
17	Sample Operations View Wireframe	80
18	Sample Operational Data Input View Wireframe	81
19	Verification Policy Input Widget	84
20	OWF Administration View	85
21	LDAP Administration View - D389 Management Console	86
22	PostgreSQL Administration – PG Admin	87
23	ARMOUR IF Administration View – HawtIO	88
24	Report Template Edit Example	89
25	Previously Run Report Widget.	90
26	Common Infrastructure Abstraction Example	93

Use or disclosure of this data is subject to the restriction on the title page of this document.

FIGURES (cont'd)

27	Reachability Graph Example	96
28	OIA Information Stack and Information Flow.....	98
29	Incident Analysis Flow	102
30	COA Analyzer	105
31	COA Analysis Process for Semi-Automated Response	106
32	Semi-Automated Response Process.....	108
33	Analysis Process for Automated Response.....	109
34	Automated Response Process	110
35	Effector Connector Framework	112
36	Hardened RTIF Environment.....	114
37	Data Diode Flow Control.....	115
38	Data Diode Internals	115

TABLES

I	ARMOUR Subsystems (3 sheets).....	7
II	Benefits of the Use of the ARMOUR IF	14
III	Product Functionality Mapping	53
IV	ARMOUR Data Sources	60
V	Data Presentation Widgets	68
VI	Assignment of Widgets to User Roles (2 sheets).....	69

APPENDICES

A	SRS Requirements Mapping
---	--------------------------

Use or disclosure of this data is subject to the restriction on the title page of this document.

This page is left blank intentionally.

Use or disclosure of this data is subject to the restriction on the title page of this document.

1. ARMOUR HIGH-LEVEL ARCHITECTURE

This Architectural Design Document (ADD) describes the systems, sub-systems and their interaction in the performance of meeting the Automated Computer Network Defence (ARMOUR) Technology Demonstration (TD) technical requirements. This ADD was prepared in accordance with ARMOUR TD Project Statement of Work (SOW) paragraph 2.1, Contract Data Requirements List (CDRL) #10 and Data Item Description (DID) SD 002.

The ARMOUR System is General Dynamics Canada Ltd.'s (GD Canada's) solution to the ARMOUR TD Project technical requirements, which are described in the System Technical Specification (STS) included in the ARMOUR TD Project Request for Proposal (RFP).

The ARMOUR System provides an integrated environment of leading-edge network cyber security tools to provide a solution that accelerates the Canadian Department of National Defence's (DND's) ability to protect and defend its networks, as shown in Figure 1.

Use or disclosure of this data is subject to the restriction on the title page of this document.

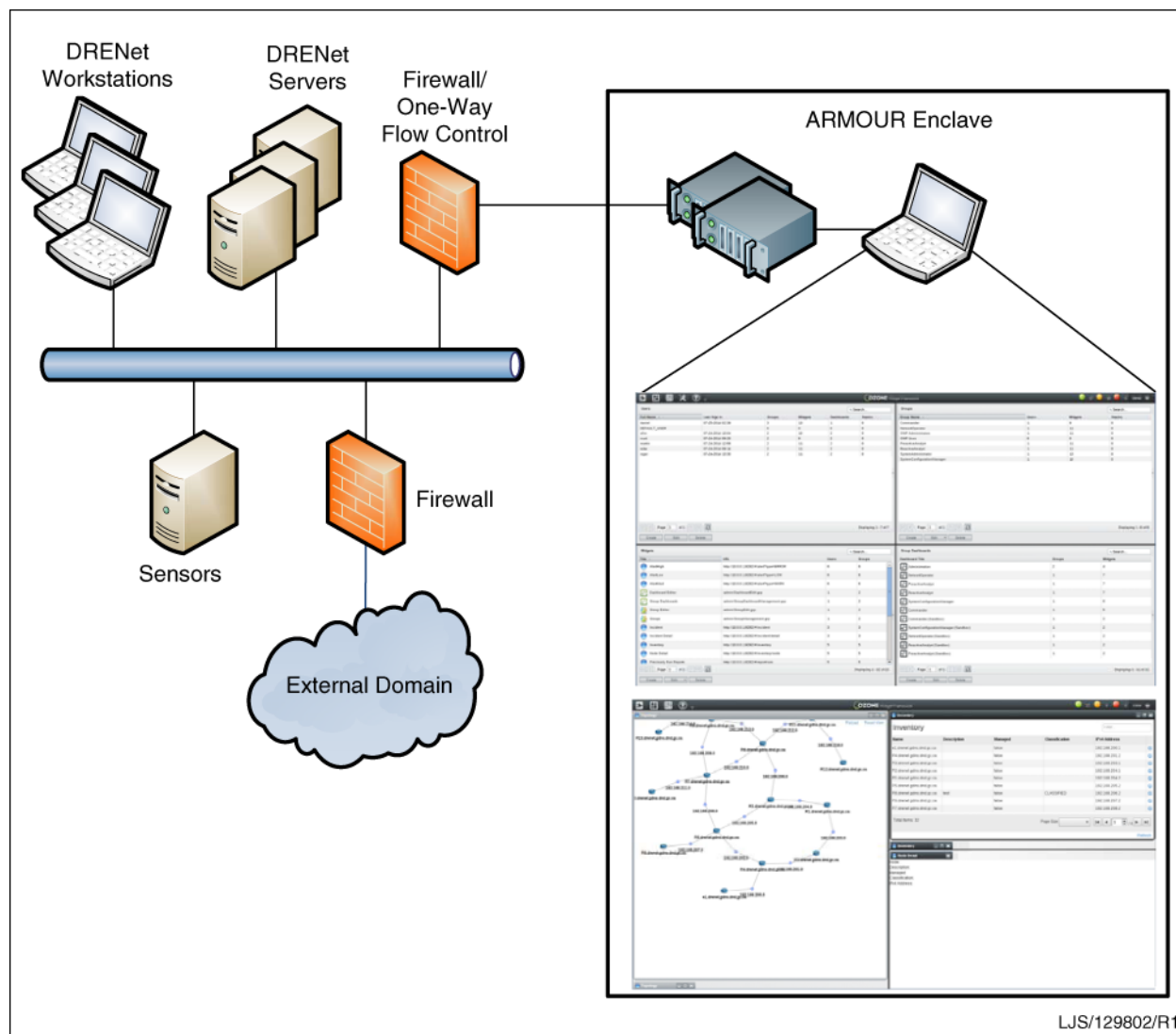


FIGURE 1. The ARMOUR System Enclave

At the core of the ARMOUR System is an integration of open source and developmental software components that are integrated to existing sensors and effectors within the Defence Research and Development Canada (DRDC) Defence Research Establishment net (DREnet) unclassified network. While the focus of these tools is to accelerate the operator's abilities to defend the network environment, the means to accomplish this is by supporting the Observe, Orient, Decide and Act (OODA) loop functions as applied to Computer Network Defence (CND). In addition to accelerating the OODA loop, the ARMOUR TD Project will also explore the challenge of the CND-related "big/fast data" problem. As the networks being defined become more complex and increasingly integrated to the operational tempo, it is difficult to discern which Course Of Action (COA) will best mitigate the attack with minimum impact on the users. In addition to increases in complexity is the intensity and tempo of attacks. Hostile agents are using ever improving tools that enable attack vectors to multiply exponentially. The challenge then with selecting the correct COA is not only ensuring that the attack is mitigated

Use or disclosure of this data is subject to the restriction on the title page of this document.

most effectively, but that it is done in the timely manner. The ARMOUR TD Project will address these challenges by establishing a modular integration framework that enables modules to be exercised and optimized to best meet the specific needs of a particular network environment.

The high-level Concept of Operations (CONOPs) of the ARMOUR TD Project is then:

Observe	Collect the “correct and complete” data to enable thorough network understanding and allow for an accurate, real-time situational awareness that is then integrated into the operational situational awareness, creating a fully integrated Common Operating Picture (COP) needed to enact the protective measures.
Orient	Extend the COP to reflect the relevance and importance of its inherent parts in support of the operational/mission requirements and the possible threat and attack vectors against the network resources.
Decide	Once the relevance and vulnerability is established for the network elements, identify remediation steps for each element and the system as a whole, in response to the identified attack paths.
Act	Respond to an attack to maintain the operational posture in the most effective and timely manner.

For the ARMOUR TD, the OODA process is shown in Figure 2. As improving the OODA process is a fundamental goal of the ARMOUR TD Project, the architecture will be positioned in order to understand how the elements and their integration accelerate the OODA steps. The OODA loop will influence the architectural decisions during all design activities.

Use or disclosure of this data is subject to the restriction on the title page of this document.

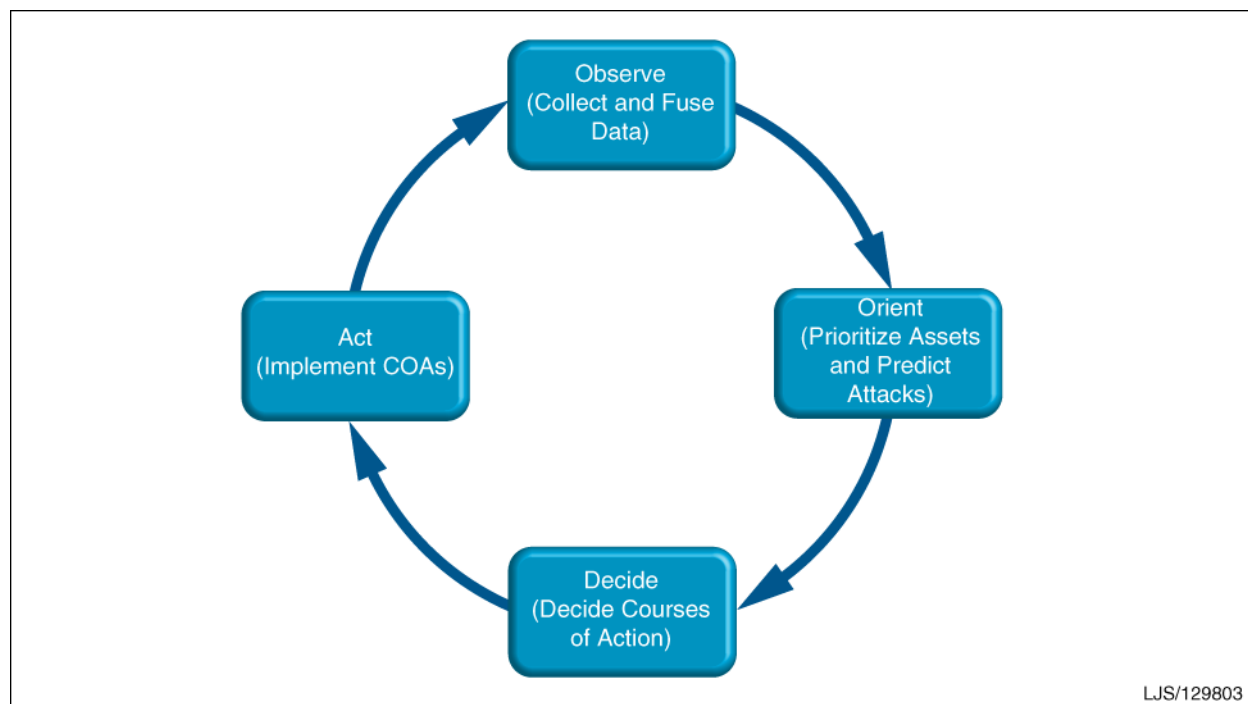


FIGURE 2. OODA Loop

For the ARMOUR TD project, a flexible open architecture ARMOUR System is provided enabling:

- a. An Automated CND demonstration platform to identify, prioritize and execute automated courses of action + metrics of effectivity; and
- b. A flexible open integration framework that supports a cyber-range for research, sharing and advancing innovations with allies, institutions, academia and commercial industry.

The ARMOUR System architecture blends the benefits of modularity with performance through the Integration Framework subsystem. (Further details on Integration Framework are found in subsection 3.1.1.)

As shown in Figure 3, the Integration Framework binds other subsystems to address the specific requirements found in the ARMOUR TD STS while also meeting the operational and performance requirements. These subsystems are:

Data storage

This subsystem supports the storage, access and retention of all information required for the ARMOUR System to function. As the environments become more complex and the operational tempo increases, this subsystem must address one of the pressing challenges in CND, being that of “big data”.

Use or disclosure of this data is subject to the restriction on the title page of this document.

Data Presentation	This subsystem supports all of the Human-Machine Interface (HMI) for ARMOUR. As the goal for ARMOUR is to support a variety of operational environments, a key aspect of the data presentation subsystem is flexibility in how the information is presented and explored.
Data normalization and correlation	Many disparate data sources provide ARMOUR with information to manage and monitor the security posture of the network. This subsystem transforms the data to a model useable by ARMOUR and correlates duplicated data sent by the various sources into a single instance.
Data analysis and action	This subsystem represents the computational elements that are the core of the ARMOUR System. It is within the Data analysis and action subsystem that the problem space is fully understood and mitigated.
Connectors	The connector subsystem represents the interaction between ARMOUR and the network infrastructure to facilitate the understanding of the current environment (i.e., network topology, operational moods, sensor feeds, etc.) and the means by which to alter the network to mitigate undesirable behaviour (i.e., attacks, compromise, suspicious traffic, etc.)

The ARMOUR TD subsystem architecture is shown in Figure 3.

Use or disclosure of this data is subject to the restriction on the title page of this document.

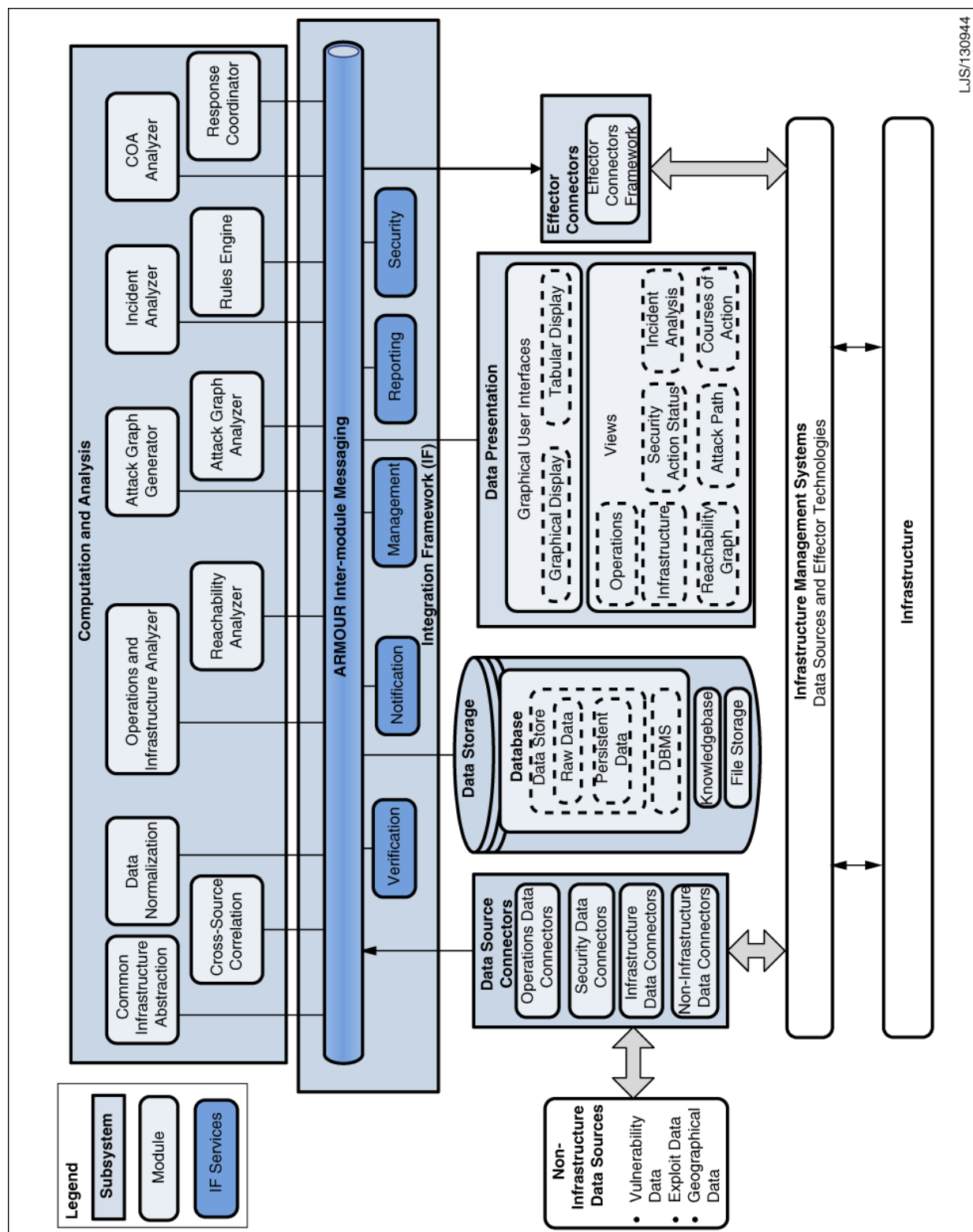


FIGURE 3. High-level Architecture for the ARMOUR System

Use or disclosure of this data is subject to the restriction on the title page of this document.

To address the ARMOUR TD problem, each of these architectural subsystems have been analyzed to identify constraints that must be met in addition to the functional requirements. The selection of Rapid Technology Integration Framework (RTIF) for the integration framework subsystem provides a mature, solid foundation for the integration of Commercial Off-The-Shelf (COTS), Open Source Software (OSS), and other third-party CND products. Table I presents each of the ARMOUR architectural subsystems, with modules comprising the subsystem, the high level function of the subsystem and its interface(s) with other architectural subsystems. The actual solutions selected to provide the functionality of the modules are described in the ARMOUR TD Detailed Design Document (GD Canada document No. 741349) and will include the integration of OSS, developed COTS and previously developed solutions. This will ensure a more mature and scalable ARMOUR capability while reducing technical and schedule risk and increasing the ability for targeted research and development into future algorithm evolution.

TABLE I ARMOUR Subsystems (3 sheets)

ARMOUR Subsystem	Module(s)	Function	Subsystem Interfaces
3.1.1 Integration Framework	<ul style="list-style-type: none"> Data Verification Data Service Inter Module Messaging Security Services Report Generation Logging Notification 	<p>The ARMOUR Integration Framework (IF) is based on Rapid Technology Integration Framework (RTIF) and provides a mature foundation for the integration of COTS, OSS and custom applications, modules and services.</p> <p>Binds Data Storage, Data Presentation, Data Normalization and Correlation, Data Analysis and Action, and Connector architectural subsystems to enable architecture/interface independent integration</p> <p>Use of open, widely used standards enables third party academia, government, or industry to easily integrate new features/modules using the RTIF Integration Development Kit (IDK)</p>	<ul style="list-style-type: none"> Data Source Connectors Data Storage Data Presentation Computational Services <ul style="list-style-type: none"> Data Normalization and Correlation Data Analysis and Action Effector Connectors
3.4.1 Data Source Connectors	<ul style="list-style-type: none"> Operations Data Connectors Security Data Connectors Infrastructure Data Connectors Non-Infrastructure Data Connectors 	<p>Ingest the information produced by data sources and provide that data to the analysis engines and presentation displays in compatible/normalized formats</p> <p>Facilitates a holistic understanding of the state of the network and individual assets</p>	<ul style="list-style-type: none"> Integration Framework Infrastructure Management Systems (external to ARMOUR) Data Sources
3.4.2 Data Storage	<ul style="list-style-type: none"> Knowledgebase Database File Storage 	<p>The database is the repository for all data collections. Information can be accessed by operators and analysis engines as required and contains long-term data storage¹</p> <p>ARMOUR IF allows for selection of desired database foundation (MySQL, Oracle, etc.)</p>	<ul style="list-style-type: none"> Integration Framework Data Presentation

¹ Long term data storage will be discussed further when ARMOUR data storage capability is known. .

Use or disclosure of this data is subject to the restriction on the title page of this document.

TABLE I ARMOUR Subsystems (3 sheets)

ARMOUR Subsystem	Module(s)	Function	Subsystem Interfaces
3.4.3 Data Presentation	<ul style="list-style-type: none"> Operations View COA View Incident Analysis View Infrastructure View Reachability Graph Generator Attack Graph Generator Security Status and Action View 	Data presentations provide the user with a graphics-based situational awareness of the network and CND events correlated to the nodes/paths comprising the network. Data presentation supports rapid indications of critical information that illustrate areas of immediate concern to the operator	<ul style="list-style-type: none"> Integration Framework Data Storage
3.4.4.1 Computational Services - Data Normalization and Cross-Source Correlation	<ul style="list-style-type: none"> Normalization Cross Source Correlation Common Infrastructure Abstraction 	<p>Reduce the volume of data and information gathered by removing redundant information collected by various tools regarding a single asset and store this data in a common database structure.</p> <p>Additionally responsible for identifying common or similar hosts within the network to allow for automated grouping and abstraction (For example, white lists managed by IT allow for aggregation of assets to be represented as a single element.)</p>	<ul style="list-style-type: none"> Integration Framework
3.4.4.3 Computational Services – Data Analysis and Action	<ul style="list-style-type: none"> Reachability Analyzer Operations/Infrastructure Analyzer Attack Graph Generator Attack Graph Analyzer COA Analyzer Incident Analyzer Response Coordinator (Semi-Automated vs Automated) Rules Engine 	<p>Provide key analysis on system information collected and provide results to the data storage / data presentation architecture subsystems.</p> <p>This subsystem is comprised of several modules that each perform a specific function. Output from one module can be used as input to another. Each module is further discussed in later sections</p>	<ul style="list-style-type: none"> Integration Framework
3.4.5 Effector Connectors	Effector Connector Framework	Provide the interface to products employed to implement the COAs.	<ul style="list-style-type: none"> Integration Framework Infrastructure Management Systems

Section 3 of this document decomposes the ARMOUR architectural subsystems, exploring the challenges with each subsystem.

Use or disclosure of this data is subject to the restriction on the title page of this document.

1.1 Changes Introduced in this Version

Each version of the ADD must include comments pertaining to the changes introduced from previous versions and their supporting rationale. The changes in the current versions are mainly introduced as a result of the following:

- Updates to the ARMOUR Data Model in support of Phase 3 specific requirements: the previous version of the ADD introduced the Data Model and the rationale for evolving the Data Model over the development phases. For Phase 2, the focus was on the basic entities and on initial classes and sub-classes of data. Phase 2 implementation has allowed a better understanding of how entities will be designed and added to the Data Model over the next development phases and more specifically for Phase 3.
- Enhanced understanding of the implemented features and functionality: for Phase 2, the focus of the ADD was to facilitate the detailed design of the foundational subsystems, namely the Integration Framework and Data Presentation subsystems. The current implementation of the later subsystems and of the initial Data Model allow this version of the ADD to describe these subsystems and modules using a more detailed or relevant understanding of implemented features and functionality.
- Enhanced understanding of modules supporting Observe and Orient: the implementation of the foundational subsystems as well as the research and development work conducted during Phase 2 allow this version of the ADD to better describe architectural concepts, subsystems and modules, especially those to be implemented in Phase 3 that support the Observe and Orient CONOPS phases. This ADD version also benefits from current architectural and design efforts being more focused towards Phase 3 requirements rather than on the foundational subsystems. The following modules directly support the Observe and Orient CONOPS phase and therefore are better defined in this version of the ADD:
 - Data Source Connectors including:
 - Operations Data Connectors (e.g., Operational Priority and Dependency information);
 - Security Data Connectors (e.g., Firewall and IPS configuration);
 - Infrastructure Data Connectors (e.g., Host and Network information);
 - Non-Infrastructure Data Connectors (e.g., Vulnerability information);
 - Cross Source Correlation;
 - Reachability Analyzer;
 - Operations and Infrastructure Analyzer;
 - Attack Graph Generator; and
 - Attack Graph Analyzer.
- Support to the Security Assessment and Authorization approach and the associated need to apply controls from the DND IT Security Controls Profile: Designated-Medium-Medium.

Use or disclosure of this data is subject to the restriction on the title page of this document.

2. APPLICABLE DOCUMENTS

The following documents were used in the development of this document.

2.1 Government Documents

The following documents were referenced in the development of this document and are applicable to the extent specified herein.

ARMOUR TDP Contract W7714-115274-SV Annex A	Statement of Work for the ARMOUR TD, 03 May 2013, v2.1
ARMOUR TDP Contract W7714-115274-SV Annex B	System Technical Specification for the ARMOUR TD, 23 May 2013, v2.1

2.2 Non-Government Documents

The following documents were referenced in the development of this document and are applicable to the extent specified herein.

740930	ARMOUR TD System Requirements Specification
741349	ARMOUR TD Detailed Design Document
741925	ARMOUR Algorithm Research and Development Report

2.3 Works Cited

Jajodia, S., & Noel, S. (2009). *Topological Vulnerability Analysis*. Fairfax, VA: Center for Secure Information Systems, George Mason University.

Jajodia, S., Noel, S., Kalapa, P., Albanese, M., & Williams, J. (2011). Cauldron, Mission-Centric Cyber Situational Awareness with Defense in Depth. *Military Communications Conference - MILCOM*, (pp. 1339-1344).

Sawilla, R. E. (2011). *Ranks and Partial Cuts in Forward Hypergraphs*. PhD Dissertation, Queen's University, School of Computing.

Sawilla, R., & Burrell, C. (2009). *Technical Memorandum 2009-130 - Course of action recommendations for practical network defence*. Ottawa: Defence R&D Canada - Ottawa.

Sawilla, R., & Ou, X. (2007). *DRDC Ottawa TM 2007-205 Googling Attack Graphs*. Defence R&D Canada - Ottawa.

Sawilla, R., & Ou, X. (2008). *DRDC Ottawa TM 2008-180 Identifying critical attack assets in dependency attack graphs*. Defence R&D Canada - Ottawa.

Use or disclosure of this data is subject to the restriction on the title page of this document.

3. ARMOUR ARCHITECTURE

3.1 Architectural Concepts

The following subsections discuss the architectural elements that provide the foundation for the ARMOUR TD solution. There are four main elements discussed:

1. Integration Framework;
2. Data Model;
3. Risk Treatment; and
4. Data Presentation.

Open Source Solutions, Research and Development (R&D) as well as COTS products provide solutions to fulfill these architectural concepts. These products will be used to create a Service-Oriented Architecture (SOA) foundation that will provide encapsulation of services, ease of module integration, scalability and discoverability.

3.1.1 Integration Framework

An integration framework that provides a foundation for quick and easy integration of Off-The-Shelf or custom services/applications/modules is necessary to support the collaborative development environment where research institutes, academia and commercial industry are able to develop and integrate new technologies and capabilities within the ARMOUR System. The ARMOUR IF is based on RTIF, a product developed by GD Canada. RTIF was selected as the ARMOUR IF solution due to its maturity, flexibility and scalability. RTIF is an open source, standards-based Enterprise Integration Framework (EIF), packaged and configured to enable distributed, decoupled integration of technologies, systems and data.

The ARMOUR IF provides the back-end services that allow the ARMOUR subsystems and modules to communicate between one another and the interfaces to external systems and services to exchange data and messages. The following list provides a brief description of each ARMOUR IF service. Additional details on the services can be found in the ARMOUR DDD (GD Canada document No. 741349).

1. **Messaging:** Messaging enables loosely coupled distributed communication in that a module can send a message to a destination and the recipient can retrieve the message without knowledge of each other. ARMOUR subsystems and modules communicate via the messaging bus.
2. **Data Service:** A generic, extensible service that supports any Data Model or any back-end database. It provides an Application Programming Interface (API) with methods to persist, find, query, remove and purge objects from the database.

Use or disclosure of this data is subject to the restriction on the title page of this document.

3. **Verification Service:** The Verification service validates the format and content of data against user configured eXtensible Access Control Markup Language (XACML) policies prior to persistence to the database, as well as prior to delivery to an Effector endpoint. Malformed data is rejected while objects that conform to the ARMOUR Data Model format are passed. The service also ensures that data containing executable code and SQL injection methods is rejected.
4. **Notification Service:** Notifications are used to alert subsystems and modules to new events. The Notification API provides the ability to publish notifications as well as to subscribe to notifications of interest. Notifications are used to inform the user or system of events of interest such as security event alerts (in the case of the user) and process status (in the case of the system).
5. **Reporting Service:** This service provides the user with the ability to generate reports on a data set of their interest. The service allows the user to generate report templates, run reports and retrieve previously run reports.
6. **Process Manager Service:** The Process Manager Service allows the system to collect the status of internal and external processes based on a subset of process properties (e.g., CPU usage, memory usage, process path, etc.) and identify processes that are taking too long (or stalled). When such instances are detected, the service generates a notification which other modules can subscribe to. Additionally, when a delinquent process is identified, a notification is presented to the user in the form of an Alert and will trigger an operator to take action.
7. **ARMOUR IF Logging:** This service logs ARMOUR IF messages according to message type and level, and stores them in the ARMOUR database.
8. **User Directory - Identification and Authentication (I&A):** The ARMOUR Lightweight Directory Access Protocol (LDAP) server supports authentication requirements for the integration framework. It prevents unauthenticated services from publishing data on the bus. The LDAP server also provides access control for the Data Presentation framework and the Database.

The ARMOUR IF enables rapid and reliable integration of COTS, OSS and R&D modules. It provides for the definition of technology agnostic Enterprise Integration Patterns (EIPs) to represent the functionality and flow of the messaging bus. EIPs are created to provide an interface between a particular module (product) and the Integration Framework, thus providing a methodology to integrate virtually any product into an Architecture that employs a middleware message bus. Within ARMOUR, the deployment of an EIP for a particular module will provide an interface for that module to communicate with the message bus. Essentially, the EIP provides an Application Program Interface (API) to the Integration Framework that will facilitate the translation and transfer of messages from the module's Data Model to the ARMOUR Data Model. This is beneficial in that virtually any technology can be integrated with a system that supports EIP through an Integration Framework, without having to modify the product to be integrated.

Use or disclosure of this data is subject to the restriction on the title page of this document.

When a developed, COTS or OSS product provides functionality for multiple capabilities, EIPs can be developed for each capability (or module). The benefit of this is that in the event where one product is discovered or developed that can replace one or more modules of a currently integrated product, an EIP can be developed that will seamlessly integrate the new product quickly and efficiently without major changes to the overall ARMOUR system. With EIPs, the integration is done at the boundary between the module and the ARMOUR system. Once integrated, the module is considered part of the ARMOUR System. Under-the-hood changes, which likely require significant effort, are not necessary as the integration effort is solely focused on the interconnection of the target product's API.

The following subsystems/modules will adhere to an EIP and provide an API for modularity:

1. Data Source Connectors:
 - (a) Security (Security Information and Event Management, Intrusion Detection System, Intrusion Prevention System);
 - (b) Infrastructure (Topology, Host Configuration, etc.);
 - (c) Non-Infrastructure (e.g., National Vulnerability Database);
 - (d) Operations Data; and
 - (e) Additional Data Source Connectors to be identified upon inspection of DRENet.
2. Computational Services Modules:
 - (a) Data Normalization;
 - (b) Cross Source Correlation;
 - (c) Common Infrastructure Abstraction;
 - (d) Operations and Infrastructure Analyzer;
 - (e) Reachability Analyzer;
 - (f) Attack Graph Generator;
 - (g) Attack Graph Analyzer;
 - (h) Incident Analyzer;
 - (i) Rules Engine;
 - (j) Course of Action Analyzer; and
 - (k) Response Coordinator.
3. Data Presentation back end; and
4. Effector Connectors.

Patterns are divided into seven types:

1. Messaging Systems – used for connecting various applications asynchronously. Messaging decouples the applications from data transfer to allow the message system to handle the data transfer.
2. Messaging Channels – a logical channel which is used to connect sender and receiver applications. One application writes a message to the channel and the other reads it from the channel;

Use or disclosure of this data is subject to the restriction on the title page of this document.

3. Message Constructions – provides the constructs and functions for creating and transforming messages;
4. Message Routing – used to connect different message channels;
5. Message Transformation – is an architectural pattern that translates data from one format to another;
6. Messaging Endpoints – used to connect an application to a messaging channel; and
7. System Management – monitor the amount of sent messages and their processing times without analyzing the message data.

Additional detail regarding the EIPs developed for the ARMOUR TD can be found in the ARMOUR TD Detailed Design Document (GD Canada document No. 741349). For more detailed information on EIP, visit <http://www.eaipatterns.com>.

The ARMOUR IF also provides standard security and event services for the ARMOUR System. Table II outlines the ARMOUR IF key benefits and features.

TABLE II Benefits of the Use of the ARMOUR IF

Feature	Benefit
Business and Information Technology (IT) Alignment	<ul style="list-style-type: none"> • Supports rapid integration of new interfaces and computational tools to reflect the changing operational environment • Design is driven by a market forces model (supply and demand) • Systems are grown to evolve with the environment rather than designed and built as a fixed structure
Adaptability	<ul style="list-style-type: none"> • Agility: allows for rapid enhancement of service capability through the current extensive support for open, standards based interfaces • Flexibility: enables on-demand composition and restructuring of services to meet business needs
Interoperability	<ul style="list-style-type: none"> • Priority on exposing capability for rapid consumption • Ability to support unanticipated utilization for emergent operational behaviours • Maximizes ability to mediate and transform disparate data and message types to achieve interoperability
Reuse	<ul style="list-style-type: none"> • Maximizes utility of the services provided • Maximizes utilization of existing services (eliminates/reduces development)
Scalability	<ul style="list-style-type: none"> • Distribution of effort: widely distribute the development of capability. The ARMOUR IF provides the ability to spawn instances of services on multiple processors and provides a means for them to communicate (similar to multithread applications) but distributed across the network. Parallel processing can be spread out across multiple systems, sharing the load. • Distribution of value: enables wide access to capability

The purple shaded subsystems/modules in Figure 4 depict where ARMOUR IF will be used in the ARMOUR Architecture.

Use or disclosure of this data is subject to the restriction on the title page of this document.

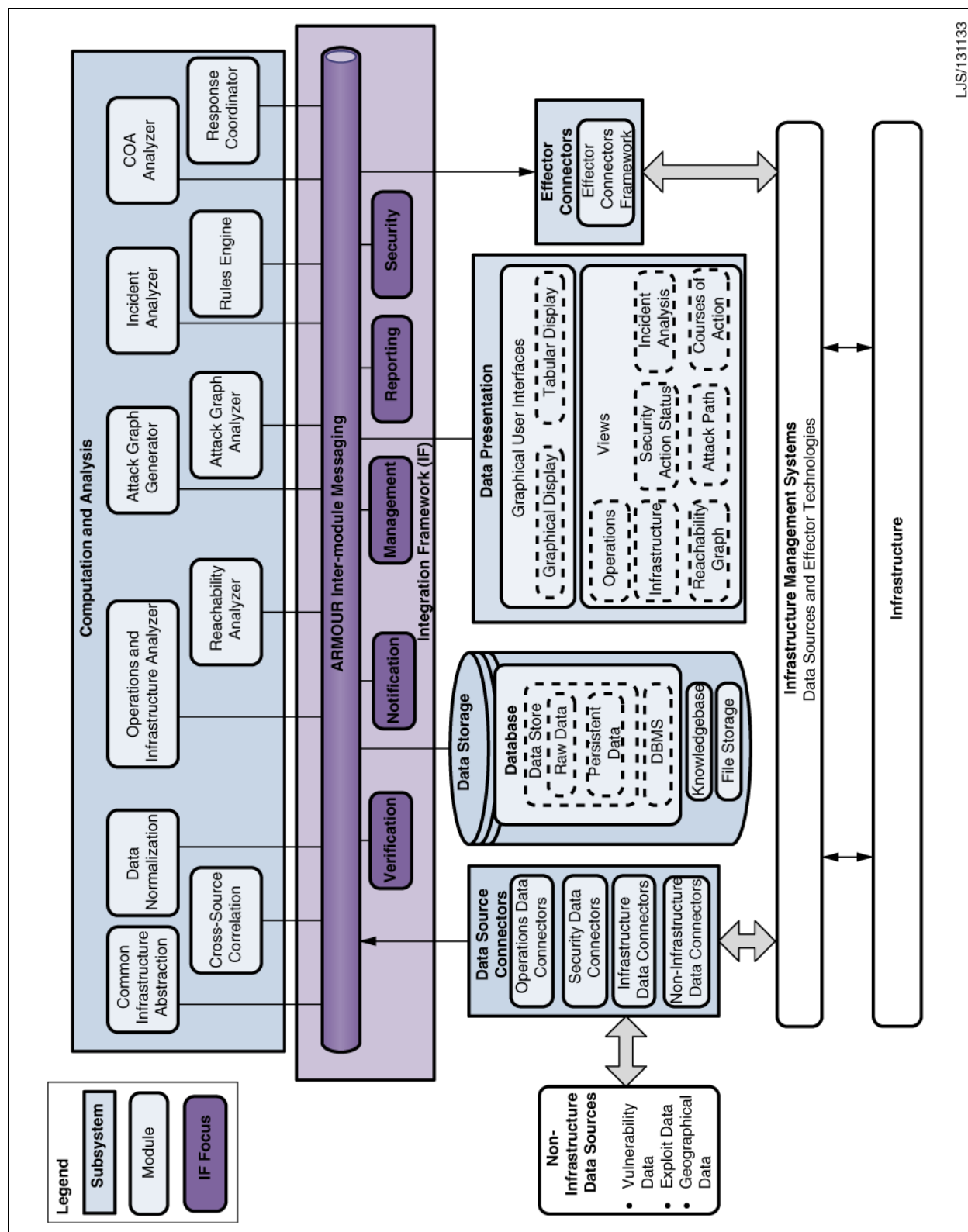


FIGURE 4. ARMOUR IF within ARMOUR Architecture

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.1.2 Data Model

The ARMOUR Data Model is designed to store data from various data sources that are integrated and queried by a variety of computation services. As such the Data Model needs to store domain specific data as well as ancillary framework data. The Phase 3 Data Model for ARMOUR extends and improves the Phase 2 Data Model based on lessons learned and on the additional requirements of this phase. The ARMOUR Phase 3 Data Model attempts to abstract or generalize datasets such that generic APIs can be written to describe the computation services and their input and output datasets.

The Phase 3 Data Model extends the Phase 2 Data Model by adding datasets to manage the following data types:

- a. Running Applications;
- b. Application Settings;
- c. Installed Software;
- d. Asset Dependencies;
- e. Known Software;
- f. Software Dependencies;
- g. Known Vulnerabilities;
- h. Access Control Lists;
- i. Resources;
- j. User Accounts;
- k. Signatures;
- l. Attacker Models;
- m. Attack Graph;
- n. Analysis Results;
- o. Metrics;
- p. Processing History;
- q. Ingestion History; and
- r. Operations Snapshots.

As a part of the Phase 3 Data Model evolution, an association between Assets (in the Phase 2 ADD this was referred to as the 'Host' dataset) and IDS Events will be created for quickly referencing assets that have been compromised due to IDS detected events.

Similar to the approach of Phase 2 where generic APIs were designed for the Transformation and Verification services, the computational services will be a collection of IComputationService implementers with pre-set input and output data types. These services will be registered as such in the application registry and they will be easily referenced in blueprint or Activator objects in the integration framework. Data Models are required to represent the various types of input data

Use or disclosure of this data is subject to the restriction on the title page of this document.

sources for ARMOUR. In developing the Data Model, there are two alternative approaches that could be followed:

- **Focused** - in the focused approach, the Data Model would be developed based on the exact input of a specifically identified upstream module which will utilize the data. In the case of ARMOUR, it would be constrained by the specific input requirements of a computational service module. Any future changes to the input requirements of the upstream module would require changes to the Data Model and associated areas.
- **Broad** - in the broad approach, a more generic Data Model would be developed based, the superset of which should be adaptable to limited number of identified upstream modules / third-party tools. This approach will require further time upfront to specify and develop but may reduce further rework in the future.

It is important that the Data Model for ARMOUR be flexible enough to allow expansion during and after the ARMOUR TDP, therefore, the broad approach will be taken. This approach will be supported with the use of loosely defined large objects. The content of these objects will be specified with a “class-type” or “data-type” field and will require validation prior to storing and retrieving data. The Data Model will also support object versioning, where multiple versions of a “class” will need to be supported over time. The OSGi architecture used by the integration framework supports this model. With this approach, computational services are able to collaborate with other services even though they may depend on different versions of the Data Model.

Because of the variety of data sources injecting raw data into ARMOUR, the Data Model must allow for some duplication of data, hence the row validation rules of certain tables will be loosened to prevent rejection of raw data. To support the multiple data sources, raw data will be tagged with provenance information (system id or host id) and time-stamped using host values – this will prevent spoofing of raw data entries by clients (and maybe used to identify directed attacks against ARMOUR).

The Data Model must also be able to support scalability and replication. To this end, all table primary keys will use Globally Unique Identifiers (GUIDs) whenever possible. Also, once data enters the system, users will not be authorized to delete records. All tables will be equipped with an “archived” Boolean field that will provide a deletion mechanism for the Data Presentation layer. This is done mainly to maintain referential integrity in the RDBMS, but will also be useful when performing Rollback operations or to allow administrators to correct user errors. At any given time, however, an ARMOUR Administrator will be capable of performing a Database Clean-Up operation, where all “archived” records will be removed and the supporting indices and table statistics updated. The Database Clean-Up can also be scheduled for regular maintenance. This will ensure optimal RDBMS performance over time.

The ARMOUR Data Model defines entities, the attributes they contain and the relationships between them. The ARMOUR Data Service supports the Data Model and supports additional features, required to inject multi-version, timestamped data and prepares the data for ingestion in version restricted computational services. This will be achieved by defining very simple service API that will be implemented by versioned services. The services will specify which dataset and

Use or disclosure of this data is subject to the restriction on the title page of this document.

version of the dataset they require and produce. The ARMOUR Data Service will also support data rollbacks as described below in a version sensitive context.

3.1.2.1 Selective Rollback Approaches

The ARMOUR STS requires that a rollback approach be implemented. Two approaches are described here: snapshot and reverse operation rollbacks.

The simplest rollback approach is to take database snapshots at appropriate times (e.g., prior to applying a COA) and whenever the administrator wants to rollback, restore a snapshot. Activity data (or Actual data) between the snapshot time and the rollback time may be lost unless it is selectively reapplied. The simplicity of this method is attractive for performance and ease of implementation reasons.

In the snapshot approach shown in Figure 5, the administrator wishes to rollback operations 2 and 3. The database is rolled back to Snapshot #1 and the subsequent operations are re-applied.

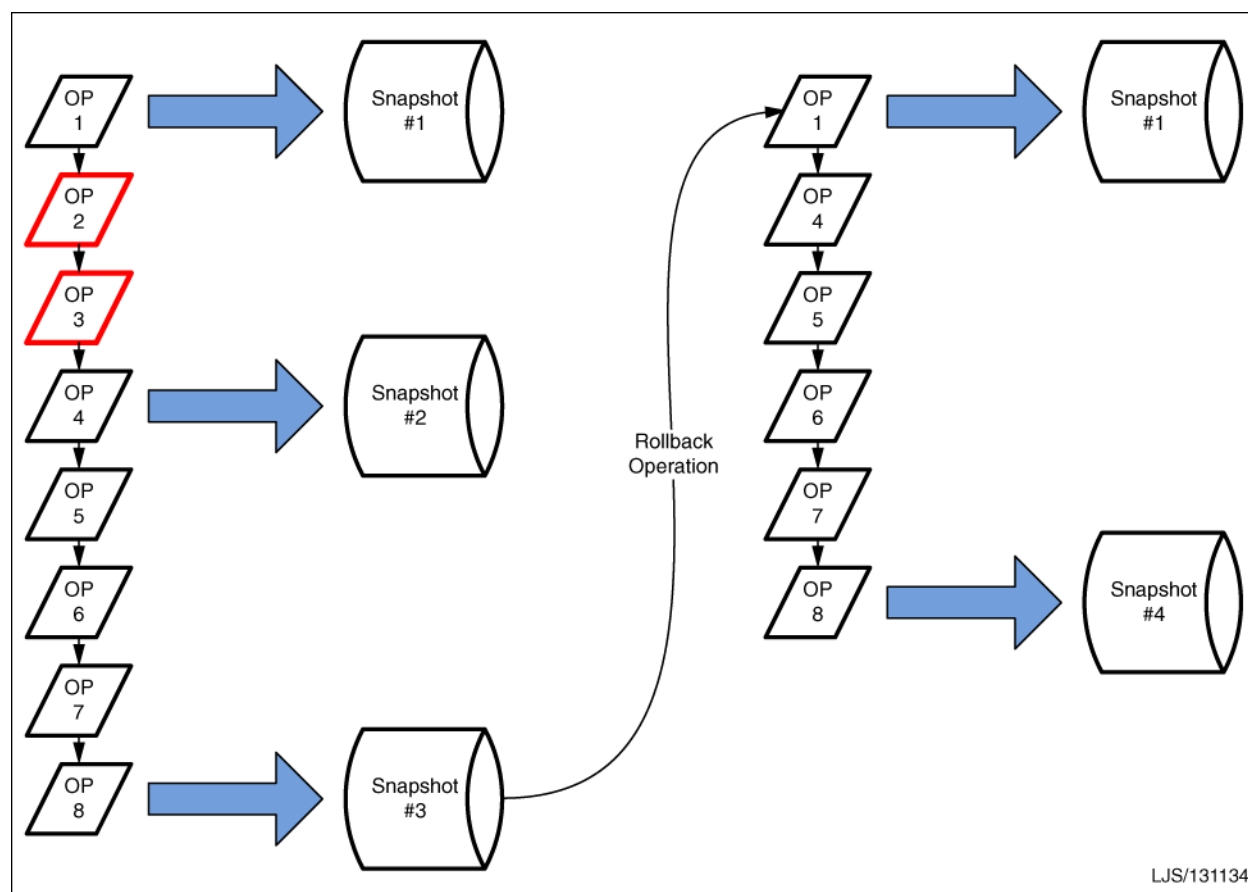


FIGURE 5. Selective Rollback: Snapshot Approach

Use or disclosure of this data is subject to the restriction on the title page of this document.

Another approach is to log all database operations with timestamp data and change information. To rollback to an earlier point in time *reverse* operations have to be applied: similar to an Undo command in a word processing application. This is not as elegant but may provide for more filtering options, such as rollback all *delete* operations or only rollback a specific database operation while leaving other items as they are. Another advantage of this approach is that the database server does not have to be placed in single-user mode while the operation is taking place.

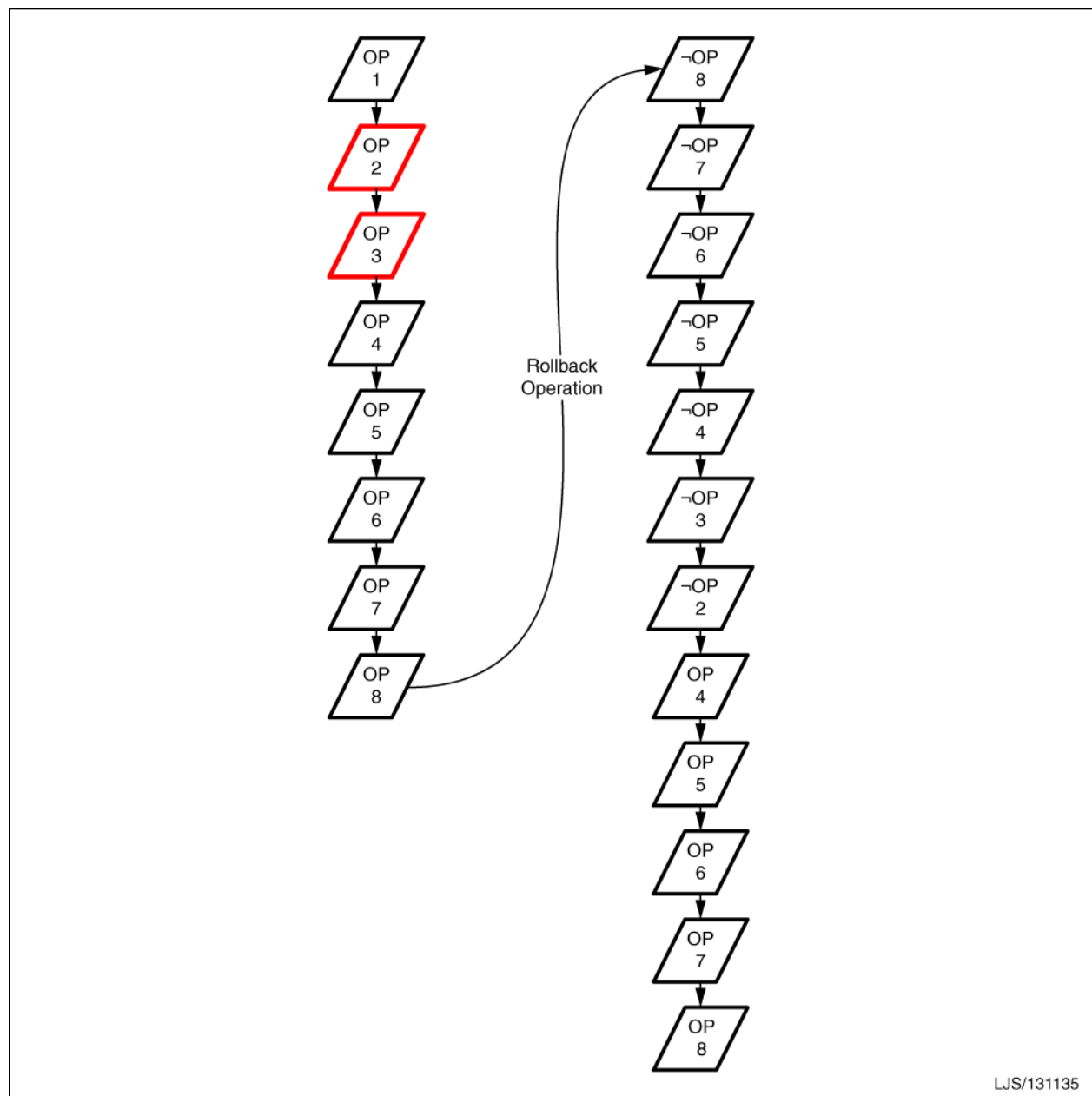


FIGURE 6. Selective Rollback: Reverse Operation Approach

Use or disclosure of this data is subject to the restriction on the title page of this document.

Both the snapshot and reverse operation approaches may exhibit problems for referential integrity, such as missing referenced data. The implications of rolling back multiple days' worth of data are difficult to evaluate without further experimentation. As the final solution to provide the rollback functionality has not been determined, this document will describe the selected approach in future revisions.

3.1.2.2 ARMOUR Phase 3 Data Model Coverage

For Phase 3 of the ARMOUR TDP, the Data Model will cover the following areas:

- Assets (includes data such as IP address, network interfaces, geographic location, etc.);
- Software and Services;
- Persons, Organizational Roles and User accounts;
- Network Topology;
- Vulnerability Assessments;
- Network Security Policy;
- IDS Events;
- Attack Graphs;
- Computational Services Interim Data;
- Metrics;
- Operational and Infrastructure Data; and
- Application specific storage (User Settings and Preferences, Report Templates and Report Storage, Knowledgebase, Notifications, Ingestion History, Data Consumption Status, Health Monitoring Data and Rollback Mechanism).

In Figure 7, the datasets are separated into three different data types to provide dataset groupings:

- a. Computer Network Defense (CND);
- b. Computational Services; and
- c. Application Support dataset groupings².

All datasets within the CND dataset groupings are time stamped and allow time-based snapshots to be analysed and processed. These datasets constitute the “data under observation” for the purpose of Proactive and Reactive network defense.

The datasets within the Computational Services dataset groupings are by-products of the system and user requests for analysis of the CND datasets. Some Computational Services will require CND datasets to be assessed at a specific point in time. Other such services will collect changes over a period of time to determine topology changes over time or to detect other “signature”

² For readability, a Dataset is a logical entity. A Dataset grouping is a functional grouping of select datasets from the Data Model.

Use or disclosure of this data is subject to the restriction on the title page of this document.

events³. The datasets within the Computational Services dataset groupings will not be affected by data rollbacks. The analysis results and interim datasets will be available upon request for user viewing (as graphical data or reports) and for other systems to base the next level of analysis on.

The datasets within the Application Support dataset groupings consist of user settings and other system ancillary data. These datasets are provided to ensure proper system functioning and configuration. These datasets will not be affected by data rollbacks.

In Figure 7, datasets in dark blue depict new datasets to support Phase 3 functionality.

³ The word signature is used here in reference to templates that help identify attacks, verification, processes, etc.

Use or disclosure of this data is subject to the restriction on the title page of this document.

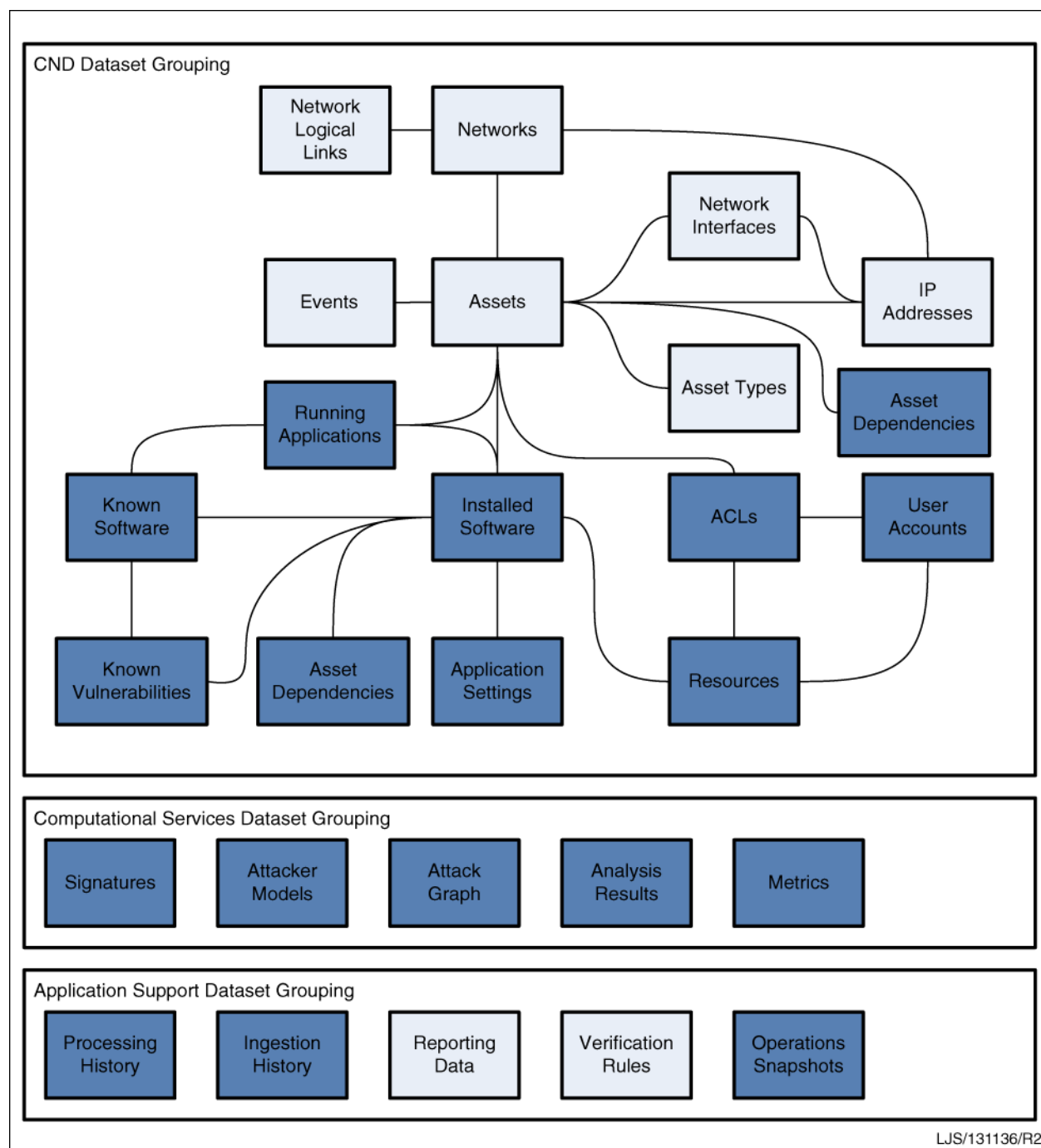


FIGURE 7. High-Level ARMOUR Data Model

The following subsections describe the planned model coverage for Phase 3.

Figure 8 is a UML representation of the CND dataset grouping in Figure 7 showing the relationships of the logical entities. Datasets in blue represent the logical entities of the CND dataset grouping added to the ARMOUR Data Model for Phase 3.

Use or disclosure of this data is subject to the restriction on the title page of this document.

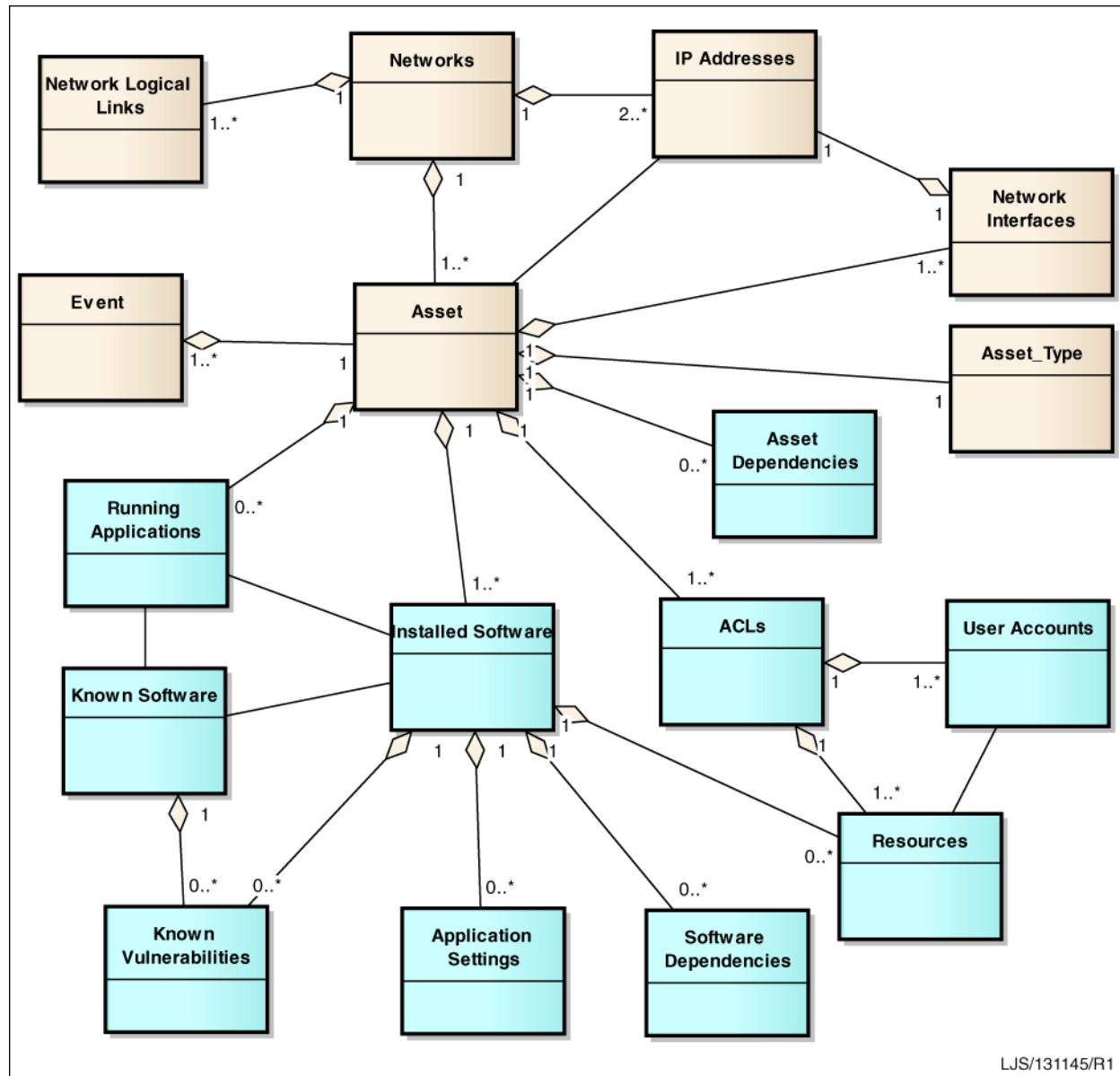


FIGURE 8. ARMOUR UML Class Diagram

3.1.2.3 CND Dataset Grouping

The CND dataset grouping is comprised of the datasets that define the Asset, Asset Configuration, Dependencies, Network Topology and Vulnerabilities data structures, and is described in the following subsections.

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.1.2.3.1 Asset Dataset

For Data Modelling purposes, an asset is any device which has at least one network address, (usually an IP address), is composed of hardware and software located at a physical geographic location, and is capable of data communication. An asset can be a physical or a virtual machine and runs only one instance of an operating system. Examples include servers, workstations, laptops, virtual machines, printers, routers, network switches, firewalls, Personal Digital Assistants (PDA), etc.

3.1.2.3.2 Known Software, Installed Software and Running Applications Dataset

The “Installed Software and Running Applications” dataset is used to identify software installed on assets, active processes and current application configuration. Operating System specific agents will be assessed as described in subsection 3.4.1.1 to select a suitable information source. The ARMOUR Host Information data source will gather host information for a subset of host (site, network, domain) and send the information to the ARMOUR Data Source Connector.

The model references the following main entities:

- **Installed Software** - This dataset represents a single installation of specific software on a specific host. This software can be a product, tool or operating system component. The list of installed software for a given asset is analysed against the list of known software and associated vulnerabilities by a computational service. The dataset allows computational service to tag known vulnerabilities for later analysis. The dataset also contains a field to store if the software matches against the Common Platform Enumeration (CPE) dataset.
- **Known Software** - This dataset will be obtained for an accredited and knowledgeable source of known software packages for a variety of operating systems and versions. The dataset will be updated regularly to provide a most accurate picture of current known software on the world market. The dataset may also be enhanced by ARMOUR users to provide complete coverage. A good source of data is MITRE. The data from MITRE is well structured, reliable and updated regularly. The software data contains a field to identify the type of software (library, OS), the manufacturer, version and whether it operates as a service or a user foreground process.
- **Running Applications** - This dataset represents processes running on assets. The running application will relate to the “Installed Software “dataset. It may contain configuration and status information. The network ports in use may be in a listening, established or closed wait state.
- **Application Settings** – This dataset represents the configuration parameters of an application such as registry entries, shared resources and TCP/UDP network ports.
- **Resources** - This dataset represents shared resources exposed or accessed by the application. These will be provided in Universal Resource Identifier (URI) form.

Use or disclosure of this data is subject to the restriction on the title page of this document.

- **Access Control Lists (ACLs)** - This dataset describes the relationship between Resources and users/roles who can access them and in what way (read, write, delete, create). This is to provide a method to identify a system-wide resource by storing URIs.
- **User Accounts** - This dataset represents user accounts.

3.1.2.3.3 Network Topology Dataset

Network topology can be modeled in a number of ways. For the purpose of ARMOUR there are essentially three ways to model the network topology.

- **Hierarchical Network Topology** - Models the network topology as a tree of subnets. Each subnet is composed of one or several IP address ranges.
- **Logical Network Topology** - Models the logical network connections between subnets. Each node is a subnet or a host. An edge between two nodes means that there exists a connection between the corresponding nodes.
- **Physical Network Topology** - Models the physical connections between subnets and hosts. Each node is a subnet or a host. An edge between two nodes represents a physical connection such as a cable, For example, workstations physically connected to a switch, to specific ports.

The ARMOUR network topology dataset uses logical links. The logical topology will allow hierarchical and physical network topologies to be modelled. The logical network topology will also describe virtual links which would not be revealed in a physical network topology.

The Network Topology Dataset is depicted by the following logical entities:

- a. Networks (in the Phase 2 ADD this was referred to as the 'Subnet' dataset);
- b. Network Interfaces;
- c. IP Addresses; and
- d. Network Logical Links.

3.1.2.3.4 Vulnerability Assessment Dataset

Vulnerability assessment models the vulnerability detected on hosts of a network by vulnerability scanning tools such as Nessus, OpenVas etc. Vulnerability data is one of the essential inputs to the generation of attack graphs, as it provides vulnerabilities which can be remotely exploited. The Vulnerability Data Model has the following essential entities.

- **Scan Report:** This Analysis Result dataset provides details about the vulnerability scan tool, timestamp and parameter used for scanning. The report result is stored in a large text object field. The report results are available to other computation services based on the Analysis Result Type.

Use or disclosure of this data is subject to the restriction on the title page of this document.

- **Vulnerability Instance:** This dataset models a vulnerability instance detected on a specific asset by a vulnerability scanner or any other source (including mapping of vulnerabilities to installed software). It may be related to an Installed Software and/or to a Running Application and be exploitable on one or several ports.
- **Known Vulnerability :** This dataset models information about known vulnerabilities published by trusted sources. Multiple vulnerability data sources are supported by providing a datasource and version field. The definition field will support large text objects where the originating data source information will be stored. Computation services requiring Vulnerability information apply a filter based on the data source type in order to adequately parse the data in a meaningful fashion. Known vulnerabilities are not necessarily indicative of compromised assets within the target network.

3.1.2.3.5 Dependencies Dataset

Dependency data provides the means for ARMOUR to store dependency relationships between assets, applications and services. Dependency relationships are used to identify the relative importance of one asset, application or service to another and to serve as input to the computational services modules. The Asset Dependency dataset depicted in Figure 8 describes one asset's dependency on another while the Software Dependency data set is used to describe one application or service's dependency on another.

The Dependencies dataset is depicted by the following logical entities in Figure 8:

- a. Asset Dependencies; and
- b. Software Dependencies.

3.1.2.3.6 Event Dataset

This dataset models any event such as a network security event on a network between source and a destination hosts. The Event dataset also contains source and destination port for the event, if available. This information may be used by computational services to identify vulnerable software that may be the target of an event.

The IDS Event Dataset is depicted by the IDS Event logical entities in Figure 8.

3.1.2.4 Computational Services Dataset Grouping

The computational services dataset grouping contains the datasets required to provide data, and to store results and metrics resulting from computations. Because a variety of computational services will be implemented during Phase 3, the model itself was decoupled to represent only the metadata of the actual dataset under process. As such, when a computational service queries the Data Service for a dataset, it will specify the data type that it can ingest. Similarly, when the computational services write data to the analysis dataset, they will also specify the analysis result type. The actual content of the data will be serialized as domain specific XML data in a large text object. As storing of XML in large objects may be problematic, additional analysis and

Use or disclosure of this data is subject to the restriction on the title page of this document.

assessment will be performed to find the best possible solution and will be described in subsequent versions of this document.

This decoupling of data will enable new computational service chains to be created with their own specific data types. This approach will provide a more flexible computing chain for developers and will assure that the Data Model will not require changing every time a new datasource or computational service is added to ARMOUR. Refer to the Phase 2, Detail Design Document (DDD) for an example of an actual decoupled dataset implementation with the ARMOUR IF Banded report engine (see subsection 5.1.6 of ARMOUR Phase 2 DDD, GD Canada document No. 741349A).

3.1.2.5 Application Support Dataset Grouping

The Application Support Dataset grouping provides ancillary data used to organize and record ARMOUR operations. The following datasets are included:

- a. **Ingestion History:** whenever new data is received by the Data Source Connectors, a record is added describing the source, the time, the data type and the location of the received data. This data will be used by computational services when new data needs to be transformed to a format that is readily understood by other services in the chain.
- b. **Processing History:** these datasets record when a computational service was run, against which dataset and whether the processing was successful or not. A link to a processing result can be stored in the result column. The Notifications dataset also falls under this dataset type. The Notifications dataset is used to support the Notification API.
- c. **Reporting Data:** these datasets consist of Report Templates and Run Report results. The template records contain the report engine specific report template detail as well as ancillary data such as report name, shared information and report engine type. The run report rows collect the path of the generated report file on the local file system and the name of the user who ran the report.
- d. **Verification Rules:** This is a list of rules used by the verification computational service. Currently, only the WSO2 Balana XACML verification engine is implemented (see subsection 3.3.1.12 for the justification for using Balana). The rule definition in XACML compliant XML is stored along with the direction of verification, the class of objects it applies to and the type of rule being verified.

Use or disclosure of this data is subject to the restriction on the title page of this document.

- e. **Operations Data Snapshots:** The operations data snapshot allows users to prioritize operations, assets, networks and applications while understanding dependencies amongst them. Moreover, all the CND data may be linked in order to provide a planned view of what components are used to perform a mission. The operations data snapshot provides the ability to tag data as simulated or real. This data might be retrieved from operational systems. Data models specific transformers for Command and Control Information Exchange Data Model (C2IEDM), Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM), Shared Operational Picture Exchange Services (SOPES) and other similar operational models may provide significant value.

3.1.3 Risk Treatment

The Risk Treatment context provides the ARMOUR System with the capability to proactively predict attack paths, evaluate the risk of exposure and determine the best course(s) of action to minimize the risk, all within the global context of the network. By evaluating asset attributes (such as connectivity, configuration, dependencies and vulnerabilities), threat attributes and reachability (logical network topology), Risk Treatment provides detailed attack graphs that describe the potential attack vector(s) given a starting point (that is, by assigning an attacker a location, inside or outside the network). Using factors such as likelihood of attack methods, likelihood of successful attack and asset ranking, Risk Treatment provides actionable information about critical risk exposures and probable avenues of attack (predictive or what-if capability). Once the attack path is understood, mitigation options are evaluated based on the global context and provided to the operator in a prioritized recommendation list. One objective of Risk Treatment is to provide the end user with the information required to make an informed decision regarding actions in the network.

Risk Treatment is part of both the proactive and reactive workflows. When used as part of the proactive flow, the current stature of the network is evaluated, and preventative measures can be evaluated in order to improve the security posture of the network and overall system. As part of a reactive workflow, the detection of an incident can trigger generation of an attack graph and evaluation of the new security posture. Assuming this can be computed in a timely manner, recommended COAs can be evaluated and automatically applied based on the threshold configured in the Course of Action Analyzer module.

The Risk Treatment context resides within the Computational Services subsystem. Several computational services modules operate in conjunction to provide the functionality. Using the various infrastructure, non-infrastructure, security and operational data points collected by the system, Risk Treatment leverages the Attack Graph Generator, Attack Graph Analyzer, and Course of Action Analyzer modules and applies them in a cooperative fashion to identify and manage the risks. It is anticipated that these modules will be fulfilled by MulVAL (see subsection 3.4.4.4.3, Proactive and Reactive Attack Graph Generator), AssetRank (see subsection 3.4.4.4.4, Proactive and Reactive Attack Graph Analyzer) and Course of Action Decision Support (COADS) (see subsection 3.4.4.4.6, Course of Action Analyzer) respectively. The Risk Treatment process is then performed in three main stages (in terms of the projected software solutions):

Use or disclosure of this data is subject to the restriction on the title page of this document.

1. MulVAL generates an Attack Graph from:
 - (a) Connectivity/Reachability information;
 - (b) Host data, including running services, service configuration, and presence of vulnerabilities;
 - (c) Security reference data, including detailed vulnerability data; and
 - (d) Operational dependencies.
2. AssetRank performs analysis on the Attack Graph to identify attack assets (assets critical to the success of the attack).
3. COADS is used to identify a set of recommended Courses of Action to manage the risk. This set will maximally increase the security of the system, which is evaluated via the Security Posture Metric, which is computed by the Operations and Infrastructure Analyzer. It is important to understand that in order to evaluate the Security Posture of various configurations, the Security Posture Metric (SPM) will likely have to be recomputed for each configuration, and this may require computation of a new attack graph each time. The results from COADS will also include the impact to the critical attack assets as computed by AssetRank (in terms of the sum of all “rank” removed from the graph by performing the COA) in order to provide the end user with the information required to make a decision as to what action is best suited for the network at the current moment in time.

MulVAL is a scalable end-to-end logic-based reasoning system that is able to conduct multi-host, multi-stage vulnerability analysis. By analysing data collected from the sources identified above and stored within the ARMOUR database, MulVAL is able to produce an understanding of the overall security by uncovering potential multi-stage, multi-host attack paths. The output of MulVAL is an AND/OR directed attack graph that is a complex mathematical abstraction of the details of the possible attacks. MulVAL uses a wide array of data points from various Data Sources and supporting analysis modules that have been stored in the ARMOUR database in the ARMOUR Data Model format.

The Asset Rank algorithm, based on Google’s PageRank algorithm, leverages the information provided by MulVAL along with various metrics including the likelihood of successful attack, maturity of attacker tools, and value of resources to the mission, to generate an understanding of asset values by assigning a rank based on dependencies (e.g., the value of an asset to the mission and the value of assets to an attacker). In order to determine the impact of security problems within a network, the dependency relationships generated by MulVAL and the vulnerability data (such as Common Vulnerability Scoring System) are used to compute a metric that represents the importance of a privilege or vulnerability to an attacker.

COADS, as currently implemented, takes as input an attack graph ranked by AssetRank, along with vertex removal costs and a maximum removal budget, and computes an optimal set of assets for removal within the budget, forming the Courses of Action set. This is computed based on which removal of those vertices that result in the maximum reduction in total “rank” in the attack graph. Note that this must be modified to satisfy the ARMOUR requirements, which indicate COAs should be evaluated based on overall improvement to the Security Posture (as

Use or disclosure of this data is subject to the restriction on the title page of this document.

computed by the Operations and Infrastructure Analyzer (OIA) module). However, rank removal is still important as an indicator as to how impactful a course of action would be.

When COADS is used with a MulVAL-generated attack graph, the Courses of Action relate to the facts within the graph, and usually relate to applying patches, shutting down services, and cutting network routes in order to improve the security posture and disrupt attackers.

APIs will be developed for each software component described above. These APIs will enable communication with ARMOUR IF using ARMOUR Data Model compliant (or verified) messages. RITF provides the middleware that facilitates the interaction between one Risk Treatment module and another. Communication amongst these modules will be through ARMOUR IF unless the design can justify direct module-to-module communication. Architecting the concept with this methodology provides the capability to replace individual modules as required without having to redesign the entire Risk Treatment solution.

The orange shaded modules in Figure 9 depicts where Risk Treatment will be implemented in the ARMOUR Architecture.

Use or disclosure of this data is subject to the restriction on the title page of this document.

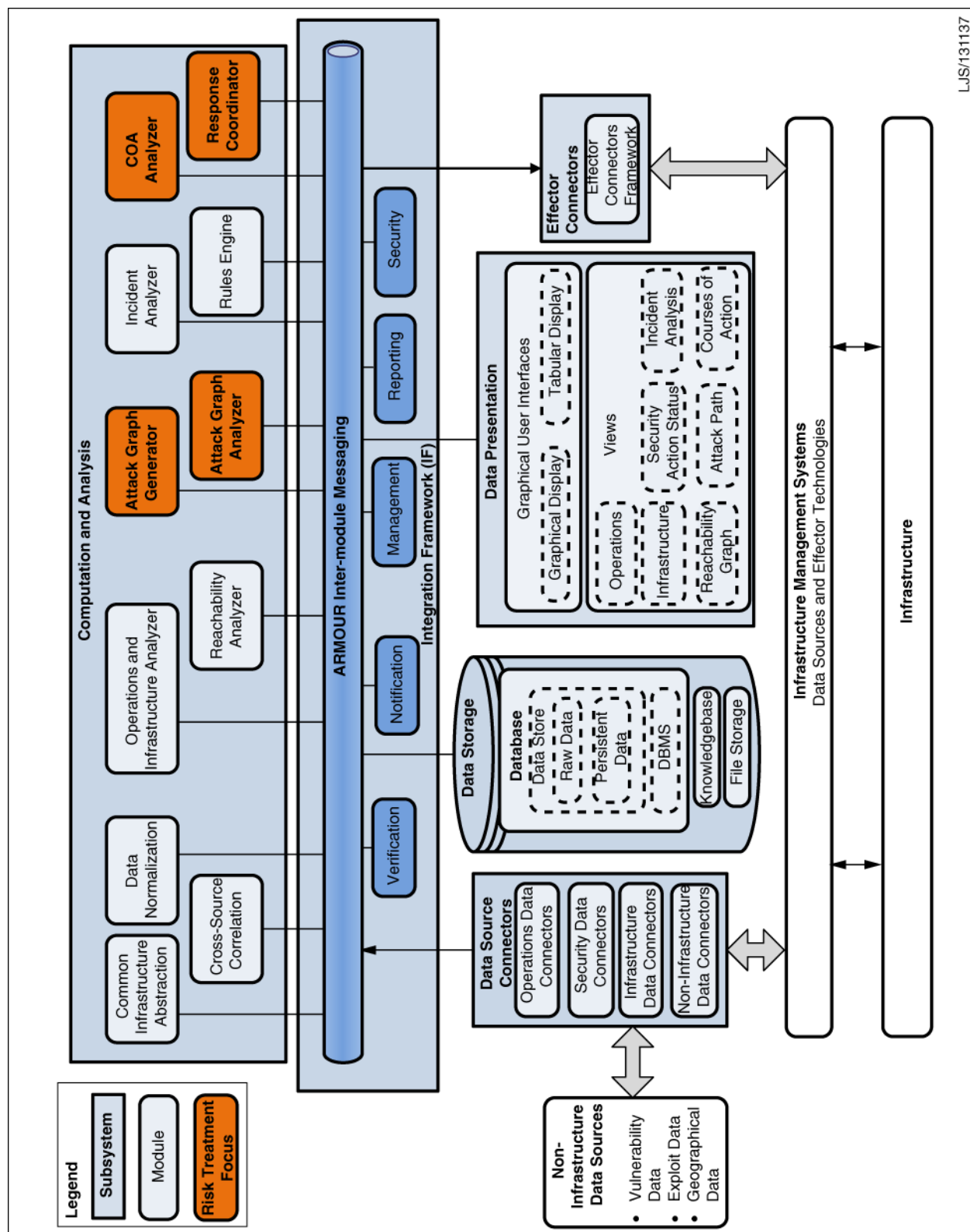


FIGURE 9. Risk Treatment within ARMOUR Architecture

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.1.4 Data Presentation

The ARMOUR System requires a rich and flexible presentation layer to seamlessly incorporate the various OSS, COTS and developed products that display mission, resource, and threat information coherently. The Data Presentation architectural concept is used to represent a capability that supports open standards and centralizes information in a single, coherent dashboard.

The solution that implements the data presentation architectural concept enables situational awareness and allows data visibility as well as coordinated decision making through interaction, exploration, and action. The data presentation provides diverse support with:

- a. Embedded browser technology for presentation;
- b. Data Notification subscription to provide automatic data updates to immediately populate visual interfaces;
- c. Linked views to provide selected asset identification across multiple widgets both graphical and tabular; and
- d. Visual analytics for quick data sorting and analysis with support for more complex computational data fusion tools.

Since the ARMOUR System may incorporate a variety of OSS, COTS and developed products, a flexible Graphical User Interface (GUI) framework is required to concisely present required information and controls.

The OZONE Widget Framework (OWF) provides this flexibility as it is capable of re-using OSS, COTS and developed products user interface as much as possible. OWF is a framework for visually organizing lightweight web application (known as widgets) within a user dashboard. This framework provides a dashboard on which widgets are contained and managed. Widgets are implemented using Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript, and as such, can essentially contain any User Interface (UI) element necessary. The dashboard is the layout used to arrange and display customized widgets that are developed for the ARMOUR presentation views.

The green modules in Figure 10 depict where modular functionality will be fulfilled by the Data Presentation subsystem within the ARMOUR Architecture. Each module (or sub-module) can be implemented as different dashboards within the OWF, each containing a set of widgets appropriate to the view's purpose.

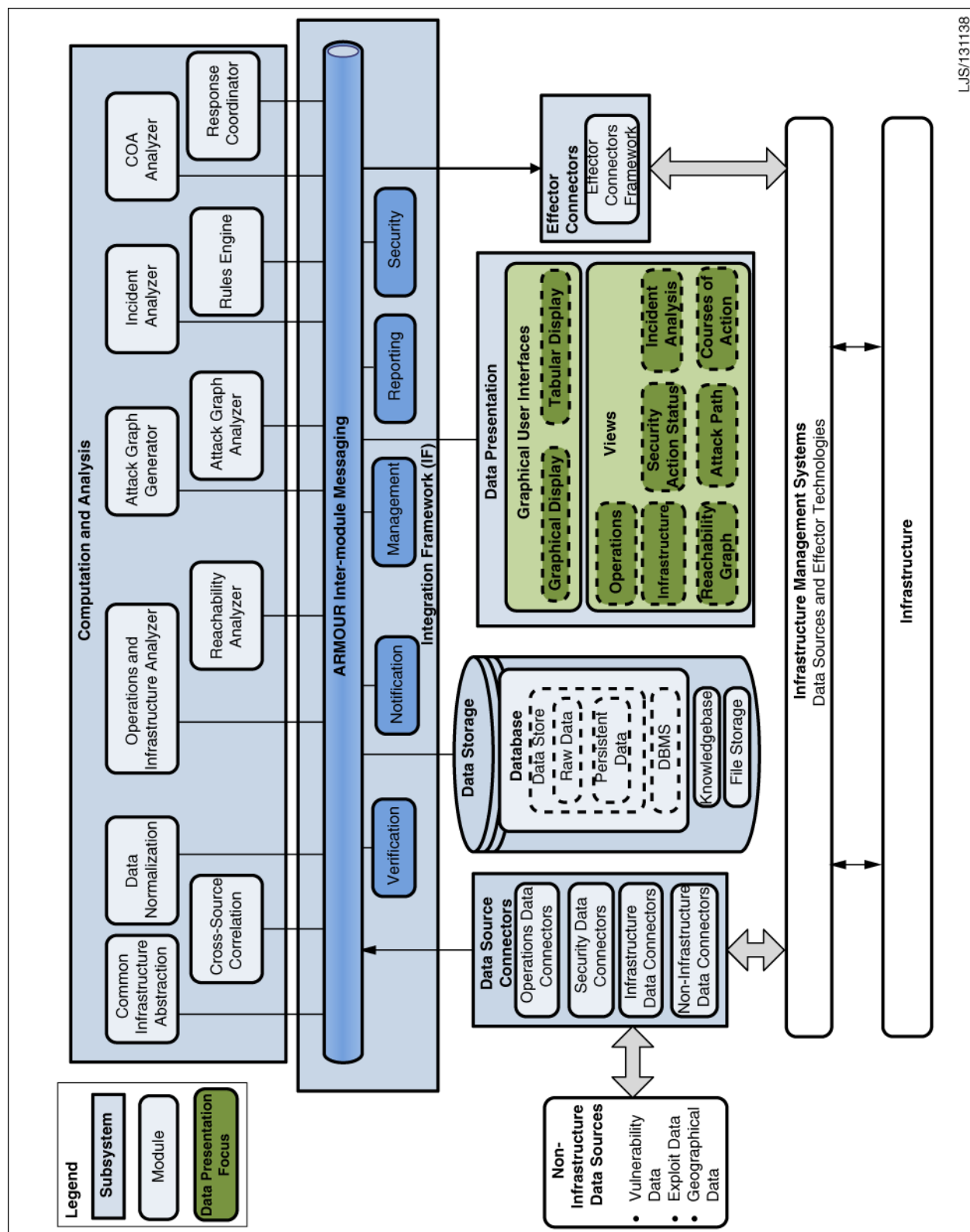


FIGURE 10. Data Presentation within ARMOUR

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.2 ARMOUR Phase 3 Logical Architecture

Figure 11 depicts the logical architecture for the development of Phase 3 of the ARMOUR TD project. This diagram shows how the ARMOUR subsystems and services interact with the Integration Framework. It can be used to describe generic use cases or scenarios as well as to describe the information flow between subsystems and services. The diagram helps illustrate how the Integration Framework provides a message bus that enables the integration of developed, COTS or OSS modules through the use of APIs and the development of technology agnostic EIPs.

The Phase 2 ADD included Use Case scenarios in order to better describe the interaction between the foundational modules and the user. Use Cases were further developed in the Phase 2 Detailed Design Document (GD Canada document No. 741349). The following use cases were included:

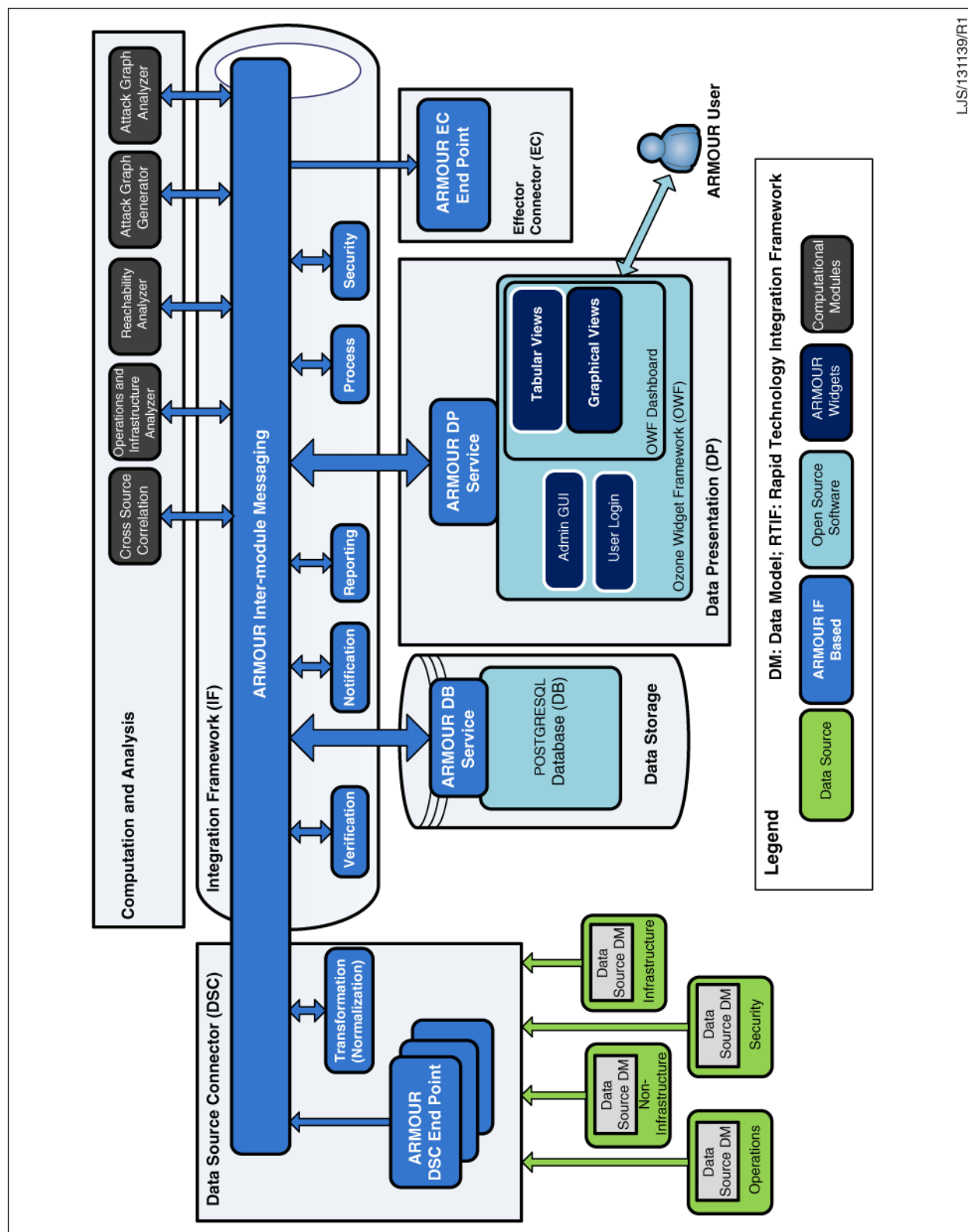
1. User Identification and Authentication;
2. User Data Request;
3. Network Data Collection and Presentation;
4. Reaction to Events; and
5. Report Generation.

For Phase 3, Use Cases that focus on the new modules/functionality will be developed to contextualize their interactions with other modules or subsystems or with the user. The use cases will be presented in the Detailed Design Document (DDD). Examples of Use Cases that cover the main areas are provided below:

1. Operational and Infrastructure analysis: User entering operational data;
2. Reachability analysis: User specifying hosts and determining reachability (the user determines reachability by taking a number of steps including specifying host); and
3. Attack graph generation and analysis: User specifying input for risk assessment.

Subsequent revisions of this document will contain additional Use Cases pertinent to the development phase requirements.

Use or disclosure of this data is subject to the restriction on the title page of this document.



LJS/131139/R1

FIGURE 11. ARMOUR System Phase 3 Logical Architecture

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.3 ARMOUR Technology

The following subsections discuss the various technologies currently being used for the Phase 2 implementation and Phase 3 development, and their relations to system functionalities / capabilities.

3.3.1 Technology Justification

Throughout the Phase 2/3 design of the ARMOUR TD, COTS, OSS and development solutions have been evaluated for inclusion in the ARMOUR solution based on numerous quality attributes for functionality and applicability. Among these attributes are:

1. API Characteristics;
2. Compliance with Recognized Standards;
3. Support Availability;
4. Functionality;
5. Licensing Cost;
6. Integration Ease;
7. Scalability;
8. Security Features;
9. Trusted Source;
10. Ubiquity;
11. Extensibility;
12. Source (organization/company) Characteristics (size, stability); and
13. IP Related Issues and Restriction for the Use of the Product.

The following subsections summarize the justifications for product selection for the ARMOUR design. Licenses for the products are compatible with their intended ARMOUR use.

3.3.1.1 Rapid Technology Integration Framework

Attribute	Product Evaluation/Assessment
Version reviewed	RTIF 1.4
API Characteristics	RTIF API is well known and documented
Compliance with Recognized Standards	RTIF is a collection of open source enterprise application integration (EAI) tools
Support Availability	RTIF 1.4 is a GD Canada product and thus support will be available for the duration of the project.

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Functionality	RTIF meets the IF requirements as outlined in the Bid. RTIF is an EIF that enables rapid and reliable integration of COTS and OSS modules Integration framework for Northern Watch
Licensing Cost	Open Source Version (\$0)
Integration Ease	Enables rapid and reliable integration of services and modules. SDKs and APIs available to support integration activities
Scalability	RTIF provides the ability to spawn instances of services on multiple processors and provides a means for them to communicate (similar to multithread applications) but distributed across the network. Parallel processing can be spread out across multiple systems, sharing the load
Security Features	TBD
Trusted Source	GD Canada Developed
Ubiquity ⁴	RTIF has a proven track record as it has been used in numerous projects: <ul style="list-style-type: none"> - Evolution of SODA - Integration framework for CoCommand - Integration framework for JIIFC
Extensibility	Supports rapid integration of new interfaces and computational tools to reflect the changing operational environment Design is driven by a market forces model (supply and demand) Systems are grown to evolve with the environment rather than designed and built as a fixed structure
Source Characteristics	GD Canada is a World Wide leader in Military Defence System Integration. The company is stable and reliable having been around for over 50 years.
IP Related Issues	None.

3.3.1.2 SMARTHawk

Attribute	Product Evaluation/Assessment
Version reviewed	SMARTHawk 3.3
API Characteristics	SMARTHawk has a well-defined API for exporting its data to third-party applications. The API has been used for integration with tools from other vendors such as Lumeta as well as by end-user networks such as WIDE in Japan, McGill University and Ryerson University.

⁴ Meaning common or widespread acceptance or influence.

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Compliance with Recognized Standards	SMARThawk supports a number of IETF standards including RFC 2328 (OSPF), RFC 2674 (VLAN MIB), RFC 1213 as well as other standards such as SOAP from W3C.
Support Availability	SMARThawk is a Solana Networks product and support is available for the contract.
Functionality	<p>SMARThawk is an intelligent network discovery, mapping and route analytics tool. It meets the requirements to provide data for the Infrastructure view and information for the Reachability Analyzer. Key features include:</p> <ul style="list-style-type: none"> • Network discovery – Routers, Switches, Hosts, Servers and other devices • Network Mapping – Layer 2 and Layer 3 (routers, switches, hosts etc.) • VLAN Discovery & Mapping • Network performance monitoring – QoS • Path Computation <p>Due to its use of SOAP/XML for data export, SMARThawk data can be exported to third-party applications running any operating system, including different variants of Linux and Windows.</p>
Licensing Cost	A binary version of the SMARThawk back-end will be included as part of the product package released for restricted open-source licensing by DRDC as part of its web portal community of interest.
Integration Ease	SMARThawk exports its distilled data including the discovered network infrastructure map and reachability information. This data is exported via a well-defined SOAP/XML API. Integration with third-party applications such as Armour is seamless. Development of adapters to receive the SMARThawk data, translate it into a native Data Model and store it in the third party application is a task that does not take much time.
Scalability	SMARThawk can scale to the largest networks in the world. It has been OEMed to Lumeta for use in networks with more than 100,000 users.
Security Features	SMARThawk supports secure network discovery for topology discovery (OSPF) as well as secure access to SNMP data.
Trusted Source	Developed by Solana Networks. Deployed on US and Canadian Government Networks.
Ubiquity	Not applicable.
Extensibility	Network architectures continue to evolve. SMARThawk's infrastructure discovery and reachability analysis are augmented by the R&D team to support any new methods of discovering network topology and performance. The Tool can scale to larger and larger networks through use of multiple devices.
Source Characteristics	Solana Networks network discovery and mapping technology provides one of the most accurate and rapid discovery techniques available on the market. The Solana team has applied rigorous and innovative R&D methodology to bring this product to market. The company was founded in 2003 and continues to grow year over year.
IP Related Issues	N/A.

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.3.1.3 Ozone Widget Framework for Graphical User Interfaces

Attribute	Product Evaluation/Assessment
Version reviewed	OWF 7.0
API Characteristics	OWF API is well documented and readily available. APIs are primarily developed with the Dojo JavaScript library. All APIs are available by referencing the hosted OWF library JavaScript file. <ul style="list-style-type: none"> - Eventing API - Preferences API - Logging API - Widget Launcher API - Drag and Drop API - Chrome API
Compliance with Recognized Standards	Supports Internet Explorer 7, 8 & 9, Firefox Requires Java 1.6 (or higher), Tomcat 7 Built with ExtJS front-end and Grails back-end
Support Availability	Limited support available through online communities: http://groups.google.com/group/ozone-developers <ul style="list-style-type: none"> - OZONEPLATFORM-ANNOUNCE@googlegroups.com - OZONEPLATFORM-USERS@googlegroups.com - OZONEPLATFORM-DEV@googlegroups.com
Functionality	A customizable open-source web application that assembles the tools you need to accomplish any task and enables those tools to communicate with each other. OWF is a web based application and thus can be run on virtually OS. Supports Internet Explorer 7, 8 & 9, Firefox
Licensing Cost	Open Source Version (\$0)
Integration Ease	Enables rapid and highly customizable development of presentation views using common coding languages. SDKs and APIs available to support development and integration activities
Scalability	Highly scalable. Supports deployment of Application Server, OWF server and Application user interface on a single HW resource or can be expanded to individual resources if necessary. OWF supports Thick, thin and web interfaces.
Security Features	OWF is controlled via plugin modules that allows custom access on a user-by-user basis. The flexibility of the framework enables widget developers to easily include their own widget-based security (i.e. you can give a user access to OWF but exclude them from specific widgets).
Trusted Source	Open Source (will be vetted through GD Canada Supply Chain Management electronic Software Authorization Checklist (eSAC) process)

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Ubiquity	OWF has a proven track record as it has been used in DND and DoD infrastructure or systems such as LCSS.
Extensibility	<p>Extensible Map Platform is an open technology stack that implements the CMAPI and provides map widget functionality that programs can use as-is, or extend for their particular needs. The stack includes the following:</p> <ul style="list-style-type: none"> • A fully functional map widget that is designed to be extensible by other programs for their particular needs, and currently provides Google Earth, OpenLayers, Cesium and WorldWind renderers • A MIL-STD 2525 symbology renderer widget that can be extended to non MIL-STD symbols • JavaScript library implementation of the CMAPI that can be used to make both map widgets and data widgets CMAPI conformant • JavaScript library for GeoJSON to KML conversions • Ecosystem of map drawing tools that work with any CMAPI (v1.2 or later) conformant map widget
Source Characteristics	OWF is an Open Source Solution.
IP Related Issues	None.

3.3.1.4 PostgreSQL Database

Attribute	Product Evaluation/Assessment
Version reviewed	PostgreSQL 9.2
API Characteristics	The PostgreSQL API is well documented and readily available on the PostgreSQL website (http://www.postgresql.org/)
Compliance with Recognized Standards	Its SQL implementation strongly conforms to the ANSI-SQL:2008 standard. It has full support for subqueries (including subselects in the FROM clause), read-committed and serializable transaction isolation levels. And while PostgreSQL has a fully relational system catalog which itself supports multiple schemas per database, its catalog is also accessible through the Information Schema as defined in the SQL standard.
Support Availability	<p>PostgreSQL has a wide variety of community and commercial support options available for users. The Community section of their website details the support options available to users from the PostgreSQL community, including mailing lists and IRC. FAQs and documentation are also available on the website.</p> <p>Commercial support is also available from one of the many companies providing professional services to the PostgreSQL community. A listing of companies that provide hosting with PostgreSQL access is also available through the website.</p> <p>Bug reporting can be accomplished through the PostgreSQL Bug reporting form that can be found on their website.</p> <p>http://www.postgresql.org/support/</p>

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment																
Functionality	<p>PostgreSQL is a powerful, open source object-relational database system. It is fully ACID compliant, has full support for foreign keys, joins, views, triggers, and stored procedures (in multiple languages). It includes most SQL:2008 data types, including INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, and TIMESTAMP.</p> <p>PostgreSQL runs on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows.</p> <p>PostgreSQL is available for almost every brand of Unix (34 platforms with the latest stable release), and Windows compatibility is available via the Cygwin framework. Native Windows compatibility is also available with version 8.0 and above.</p> <p>It includes most SQL:2008 data types, including INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, and TIMESTAMP. It also supports storage of binary large objects, including pictures, sounds, or video. It has native programming interfaces for C/C++, Java, .Net, Perl, Python, Ruby, Tcl and ODBC.</p> <p>The following website lists the features: http://www.postgresql.org/about/featurematrix/</p>																
Licensing Cost	PostgreSQL is released under the PostgreSQL License, a liberal Open Source license, similar to the BSD or MIT licenses.																
Integration Ease	With a readily available API and extensive documentation, Integration is expected to be relatively straight forward.																
Scalability	<p>PostgreSQL is highly scalable both in the sheer quantity of data it can manage and in the number of concurrent users it can accommodate.</p> <table> <thead> <tr> <th data-bbox="516 1113 803 1144">Limit</th><th data-bbox="836 1113 950 1144">Value</th></tr> </thead> <tbody> <tr> <td data-bbox="516 1165 803 1197">Maximum Database Size</td><td data-bbox="836 1165 950 1197">Unlimited</td></tr> <tr> <td data-bbox="516 1228 803 1260">Maximum Table Size</td><td data-bbox="836 1228 950 1260">32 TB</td></tr> <tr> <td data-bbox="516 1291 803 1323">Maximum Row Size</td><td data-bbox="836 1291 950 1323">1.6 TB</td></tr> <tr> <td data-bbox="516 1354 803 1386">Maximum Field Size</td><td data-bbox="836 1354 950 1386">1 GB</td></tr> <tr> <td data-bbox="516 1417 803 1449">Maximum Rows per Table</td><td data-bbox="836 1417 950 1449">Unlimited</td></tr> <tr> <td data-bbox="516 1480 803 1512">Maximum Columns per Table</td><td data-bbox="836 1480 950 1512">250 - 1600 depending on column types</td></tr> <tr> <td data-bbox="516 1543 803 1575">Maximum Indexes per Table</td><td data-bbox="836 1543 950 1575">Unlimited</td></tr> </tbody> </table>	Limit	Value	Maximum Database Size	Unlimited	Maximum Table Size	32 TB	Maximum Row Size	1.6 TB	Maximum Field Size	1 GB	Maximum Rows per Table	Unlimited	Maximum Columns per Table	250 - 1600 depending on column types	Maximum Indexes per Table	Unlimited
Limit	Value																
Maximum Database Size	Unlimited																
Maximum Table Size	32 TB																
Maximum Row Size	1.6 TB																
Maximum Field Size	1 GB																
Maximum Rows per Table	Unlimited																
Maximum Columns per Table	250 - 1600 depending on column types																
Maximum Indexes per Table	Unlimited																

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Security Features	<p>PostgreSQL boasts sophisticated features such as Multi-Version Concurrency Control (MVCC), point in time recovery, tablespaces, asynchronous replication, nested transactions (savepoints), online/hot backups, a sophisticated query planner/optimizer, and write ahead logging for fault tolerance.</p> <p>Data integrity features include (compound) primary keys, foreign keys with restricting and cascading updates/deletes, check constraints, unique constraints, and not null constraints.</p> <p>PostgreSQL has earned it a strong reputation for reliability, data integrity, and correctness</p> <p>PostgreSQL publishes all known security vulnerabilities here: http://www.postgresql.org/support/security/</p> <p>Security updates are primarily made as minor version updates.</p>
Trusted Source	Open Source (will be vetted through GD Canada Supply Chain Management eSAC process)
Ubiquity	PostgreSQL is used in a wide variety of commercial and non-commercial applications.
Extensibility	<p>PostgreSQL source code is available to all at no charge.</p> <p>PostgreSQL runs stored procedures in more than a dozen programming languages, including Java, Perl, Python, Ruby, Tcl, C/C++, and its own PL/pgSQL, which is similar to Oracle's PL/SQL. Included with its standard function library are hundreds of built-in functions that range from basic math and string operations to cryptography and Oracle compatibility. Triggers and stored procedures can be written in C and loaded into the database as a library, allowing great flexibility in extending its capabilities. Similarly, PostgreSQL includes a framework that allows developers to define and create their own custom data types along with supporting functions and operators that define their behavior.</p>
Source Characteristics	PostgreSQL has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness.
IP Related Issues	None

3.3.1.5 SNORT IDS

Attribute	Product Evaluation/Assessment
Version reviewed	SNORT 2.9.6.0
API Characteristics	SNORT API is readily available on its website http://www.snort.org
Compliance with Recognized Standards	SNORT supports SYSLOG and Common Event Format (CEF) standard outputs.

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Support Availability	<p>SNORT is the most widely used open-source NIDS module. As with all open source software, limited(free) support is available through online communities:</p> <p>snort-users@lists.sourceforge.net : General discussion about SNORT</p> <p>snort-sigs@lists.sourceforge.net : Discussion and development of SNORT rules</p> <p>snort-devel@lists.sourceforge.net : SNORT development discussion</p> <p>snort-openappid@lists.sourceforge.net : SNORT OpenAppId discussion</p>
Functionality	<p>SNORT is a powerful lightweight Network Intrusion Detection System, capable of performing real-time traffic analysis to detect cyber security threats. It uses a signature-based protocol analysis with content searching/matching to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probs, OS fingerprinting, etc.</p> <p>SNORT runs on all major operating systems, including Linux, UNIX(Solaris, BSD, IRIX, HP-UX, Mac OS X), and Windows</p>
Licensing Cost	SNORT is available under the free software/open source GNU General Public License(GPL) version 2.
Integration Ease	SNORT runs independently and provides several export methods suitable for variety integration requirements. A well-documented user manual can help with the use of these export methods.
Scalability	<p>SNORT can be deployed on any network environment.</p> <p>SNORT rules are developed separately using a simple rule description language. It is easy to use to develop new rules to detect new types of network attacks.</p>
Security Features	SNORT is one of the industry's premier technologies for detecting cyber security threats.
Trusted Source	Open Source (will be vetted through GD Canada Supply Chain Management eSAC process)
Ubiquity	SNORT is used in a wide variety of infrastructures.
Extensibility	<p>SNORT source code is available to anyone at no charge.</p> <p>SNORT utilizes a modular plug-in architecture which is easily extensible.</p> <p>SNORT rules are developed separately. New rules can be added using a simple rule description language.</p>
Source Characteristics	SNORT is an Open Source solution. It was created in 1998 and is still in active development. Cisco has been supporting the SNORT project as open source innovation when it acquired Sourcefire in 2013.
IP Related Issues	None

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.3.1.6 MulVAL

Attribute	Product Evaluation/Assessment
Version Reviewed	V1.1
API Characteristics	<p>N/A – MulVAL is a standalone application. GD Canada team plans on integrating MulVAL first by wrapping it with an API, and as time permits, refactoring by bringing the main processing engine out and making it available via a native API to improve performance.</p> <p>As a stand-alone application, input to MulVAL consists of files that contain a list of facts about the system to be analyzed (including infrastructure and host information), and a list of logical rules for combining these facts. MulVAL outputs Comma-Separated Values (CSV) files that contain the vertices and arcs for the attack graph. In this format, they are easily consumed by other applications.</p>
Compliance with Recognized Standards	<p>MulVAL is designed to work with input data produced from OVAL-compliant scanners, Nessus scans, and security data extracted from NVD. It includes translators that convert the raw data into the appropriate format for MulVAL to use.</p> <p>The semantics of the MulVAL-generated attack graphs (AND/OR directed graphs) are used by other attack graph generating tools (including Cauldron), and as such developing with MulVAL in mind still supports a modular attack graph generator.</p>
Support Availability	<p>MulVAL is open-source software developed at Kansas State University (KSU). One of the major contributors is still involved in the research area and is available to provide support.</p>
Functionality	<p>The main functionality provided by MulVAL is generating attack graphs given appropriate input data, including vulnerability information, host configuration information, and infrastructure information.</p> <p>At its core, MulVAL is a logic processing engine. Given the input listed above as a series of “facts” and a set of logical rules for combining these facts, the MulVAL engine logically computes the attack graph.</p> <p>The facts are easily generated from the data available in the ARMOUR database, and the logical rules, once defined, typically do not change. The set of rules being considered may change depending on the scenario, but the content would remain fixed.</p> <p>MulVAL has been tested by the developers on Linux and Mac OS X operating systems. GD Canada has used it on Linux systems. There is no immediately identified reason why it would not run on Windows, assuming the prerequisite software (XSB, GraphViz) is available. Note that as MulVAL as integrated in ARMOUR evolves, the need for GraphViz is expected to be removed.</p>
Licensing Cost	MulVAL is free, open-source software released under the GNU General Public License version 3.
Integration Ease	Initial integration by creating a wrapper for MulVAL is expected to be relatively easy. The second stage, creating a native API for the MulVAL engine, is anticipated to be more difficult.
Scalability	TBD

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Security Features	Not Applicable
Trusted Source	MulVAL was developed at KSU, a member of the GD Canada ARMOUR bid team. It is software that has been used by various DRDC and NATO research projects. The development was partially supported by the National Science Foundation, by the Air Force Office of Scientific Research, and by HP Labs Innovation Research Program.
Ubiquity	Not Applicable
Extensibility	The logical rules used by MulVAL to compute an attack graph are part of the input provided to the engine. As a result, it is relatively easy to fine-tune to create of an attack graph to better model the environments and security concerns in the context of ARMOUR. If new methods of exploitation are discovered/learned they can be added to MulVAL for inclusion when generating attack graphs.
Source Characteristics	MulVAL is a research tool developed by a team of professors and students at KSU. Members of this team have worked with GD Canada and DRDC in the past. It is an open-source project, and individuals at both organizations have had the opportunity to learn and understand it, addressing concerns about KSU being a single-point-of-failure for the knowledge. KSU received designation as a “National Center of Academic Excellence in Information Assurance/Cyber Defence Research” from the NSA for the 2014 through 2019 academic years.
IP Related Issues	None anticipated

3.3.1.7 AssetRank

Attribute	Product Evaluation/Assessment
Version Reviewed	V3.0.1
API Characteristics	AssetRank is distributed as a Python-developed set of libraries, including COADS. It is distributed with documentation outlining the basic functionality available, including sample function calls. There are also sample scripts that demonstrate how these tools can be used.
Compliance with Recognized Standards	AssetRank was designed to work with the output from MulVAL (CSV files containing the vertices and arcs for an attack graph), and computes a variety of information. As it is a Python library, the standard methods of interfacing with Python code apply, including recovering the applicable output data.
Support Availability	AssetRank is provided by DRDC for development purposes. GD Canada employees have worked with AssetRank in the past, and are familiar with both the source code and the academic papers describing the algorithms.

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Functionality	AssetRank, given an AND/OR dependency attack graph and various metrics information (including likelihood of attacker success, maturity of attack tools, likelihood of successful social engineering attacks, and identification of critical assets), assigns a rank weight to each vertex indicating the relative value of the vertex to an attacker. That is, it identifies the vertices that an attacker is most dependent on when attacking the network. As a Python library, there should be little issue with AssetRank being used on a variety of systems.
Licensing Cost	The AssetRank license is Proprietary, the software is not to be distributed without permission. The holder is Her Majesty the Queen in Right of Canada as represented by the Minister of National Defence.
Integration Ease	As AssetRank (and COADS) is provided as Python libraries, initial integration is expected to be relatively easy. Minor code modification may be required to support all the metrics required for ARMOUR, but again, this is expected to be relatively easy.
Scalability	TBD
Security Features	N/A
Trusted Source	AssetRank was developed by DRDC employees with support from members of the research community, including Kansas State University.
Ubiquity	Not Applicable
Extensibility	The AssetRank algorithm, as an evolution of Google's PageRank algorithm, is well documented in multiple papers and a PhD Thesis. (Sawilla & Ou, DRDC Ottawa TM 2007-205 Googling Attack Graphs, 2007) (Sawilla & Ou, DRDC Ottawa TM 2008-180 Identifying critical attack assets in dependency attack graphs, 2008) (Sawilla R. E., 2011) Because the source code is available, modifications can be made as required.
Source Characteristics	AssetRank is provided by DRDC.
IP Related Issues	None anticipated

3.3.1.8 COADS

Attribute	Product Evaluation/Assessment
Version Reviewed	V3.0.1
API Characteristics	COADS is distributed as part of the AssetRank Python-developed set of libraries. It is distributed with documentation outlining the basic functionality available, including sample function calls. There are also sample scripts that demonstrate how these tools can be used.
Compliance with Recognized Standards	COADS exists as an extension to AssetRank. It uses an AssetRank ranked graph as input and produces a number of sets of courses of action for output, depending on the options used. As it is a Python library, the standard methods of interfacing with Python code apply, including recovering the applicable output data.

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Support Availability	COADS is provided by DRDC for development purposes. GD Canada employees have worked with COADS in the past, and are familiar with both the source code and the academic papers describing the algorithms.
Functionality	<p>COADS, given an attack graph ranked using the AssetRank algorithm, costs for courses of action (COAs), and a budget, produces an (optimal) set of COAs to apply to the network that would result in the maximum disruption to the attacker's ability to exploit the network.</p> <p>There are three algorithms for determining the set of COAs implemented in COADS: a brute force algorithm, a greedy algorithm, and a hybrid algorithm. In practice, the brute force algorithm always returns a set that results in the removal of the most "rank" within the budget, but takes a significant amount of time to compute. The greedy algorithm, while much faster, may not return a set close to the optimal one. The hybrid runs the greedy algorithm, and then uses brute force on the set returned by the greedy algorithm in order to remove redundancies. This process is repeated until there is no room left in the budget.</p> <p>On other projects, COADS has been modified to make use of evaluation methods besides AssetRank, including the likelihood of survival for the length of the mission. In that case, COADS returns the set of COAs that would best improve the survivability. Similar modification is required for use within ARMOUR.</p>
Licensing Cost	The COADS license is Proprietary, the software is not to be distributed without permission. The holder is Her Majesty the Queen in Right of Canada as represented by the Minister of National Defence.
Integration Ease	As AssetRank, and thus COADS, are provided as Python libraries, initial integration is expected to be relatively easy. Code modification is expected in order to support the specific requirements of ARMOUR (in particular to support generation of a COA set based on improvement in the Security Posture Metric), but this is again anticipated to be relatively easy.
Scalability	TBD
Security Features	N/A
Trusted Source	COADS was developed by DRDC employees with support from members of the research community, including Kansas State University.
Ubiquity	Not Applicable
Extensibility	<p>The algorithms used by COADS are documented in multiple papers and a PhD thesis. (Sawilla & Burrell, Technical Memorandum 2009-130 - Course of action recommendations for practical network defence, 2009) (Sawilla R. E., 2011)</p> <p>Because the source code is available, modifications can be made as required.</p>
Source Characteristics	COADS is provided by DRDC.
IP Related Issues	None anticipated

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.3.1.9 National Vulnerability Database

Attribute	Product Evaluation/Assessment
Version Reviewed	V2.2
API Characteristics	N/A – NVD is a web-based Data Source that provides its Vulnerability Data Set as an Rich Site Summary (RSS) Feed
Compliance with Recognized Standards	NVD Feed is provided as an RSS Feed and XML format and represented using the Security Content Automation Protocol (SCAP) The SCAP data feeds include Common Vulnerabilities and Exposures (CVE), Common Platform Enumeration (CPE) and the Common Vulnerability Scoring System (CVSS)
Support Availability	Frequent updates to feeds and support available from NVD website: http://nvd.nist.gov/
Functionality	Provides up-to-date data on known vulnerabilities (CVEs) with metrics on the vulnerabilities (i.e. their severity/exploitability, existence of a patch, etc). This will be used as a Data Source for vulnerability data.
Licensing Cost	\$0
Integration Ease	Integration with ARMOUR is expected to be relatively easy. Policies will have to surround the collection of data and ingress to the ARMOUR system as ARMOUR will not be connected to the internet to download the RSS feeds.
Scalability	N/A
Security Features	N/A
Trusted Source	NVD is a trusted repository for the latest information on Security Vulnerabilities. NVD is the U.S. government repository of standards based vulnerability management data. This data enables automation of vulnerability management, security measurement, and compliance
Ubiquity	NVD is used by a wide variety of organizations.
Extensibility	N/A
Source Characteristics	NVD is the U.S. government repository of standards based vulnerability management data. It is a product of the National Institute of Standards and Technologies (NIST)
IP Related Issues	None

3.3.1.10 Data Source Connectors

Attribute	Product Evaluation/Assessment
API Characteristics	Implemented with EIP using RTIF API and Data Source APIs as required.
Compliance with Recognized Standards	Apache Camel Enterprise Integration Patterns
Support Availability	Support for RTIF is provided. Online discussion forms available for Apache support.

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Functionality	Provides the message systems, channels, constructions, routing, transformation, endpoints and system management to enable communication between the data source and the Integration Framework (ARMOUR Data Model). Apache Camel is commonly used throughout the industry
Licensing Cost	\$0
Integration Ease	With access to ARMOUR APIs and Data Source APIs, the integration effort should be minimal.
Scalability	To Be Determined (TBD)
Security Features	<p>Camel offers several forms & levels of security capabilities that can be utilized on camel routes. These various forms of security may be used in conjunction with each other or separately. The broad categories offered are:</p> <ul style="list-style-type: none"> • Route Security - Authentication and Authorization services to proceed on a route or route segment • Payload Security - Data Formats that offer encryption/decryption services at the payload level • Endpoint Security - Security offered by components that can be utilized by endpointUri associated with the component • Configuration Security - Security offered by encrypting sensitive information from configuration files
Trusted Source	GD Canada developed.
Ubiquity	Not Applicable
Extensibility	Camel supports multiple languages in the DSL or XML Configuration for maximum extensibility
Source Characteristics	Apache Camel is Open Source
IP Related Issues	None

3.3.1.11 Barnyard2

Attribute	Product Evaluation/Assessment
Version Reviewed	2.1.13
API Characteristics	The Barnyard2 API is readily available on the https://github.com/firnsy/barnyard2 website.
Compliance with Recognized Standards	Barnyard2 supports SYSLOG and Common Event Format(CEF) standards.

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Support Availability	<p>Barnyard2 is open source software. Limited (free) support is available through online communities:</p> <p>barnyard2-users@googlegroups.com</p> <ul style="list-style-type: none"> • Discussion about user problems and issues <p>barnyard2-devel@googlegroups.com</p> <ul style="list-style-type: none"> • Barnyard2 development discussion
Functionality	<p>Barnyard2 is a dedicated output utility for SNORT. It parses SNORT Unified2 format files as input, processes and on-forwards to a variety of output format interfaces, including a variety of databases, Sguil system, SYSLOG, Prelude hybrid IDS System, Bro-IDS Instance and CEF. The aim of creating the Barnyard2 tool is to allow SNORT to run efficiently without missing network traffic and decouple the time-consuming output task from SNORT. Barnyard2 runs on all major operation systems, including Linux, uNIX (Solaris, BSD, IRIX, HP-UX, Mac OS X), and Windows</p>
Licensing Cost	<p>Barnyard2 is available under the free software/open source GNU General Public License(GPL) version 2</p>
Integration Ease	<p>Barnyard2 runs independently. A well-documented user manual can help with the Barnyard2 configuration to integrate with SNORT and the use of an export method.</p>
Scalability	None
Security Features	None
Trusted Source	<p>Open Source (will be vetted through GD Canada Supply Chain Management eSAC process)</p>
Ubiquity	Barnyard2 is a dedicated output utility for SNORT.
Extensibility	<p>Barnyard2 source code is available to all at no charge.</p> <p>Barnyard2 utilizes a modular plug-in architecture. It is allowed to be easily extensible.</p>
Source Characteristics	<p>Barnyard2 is an open source output tool for Snort. It's a the successor to Barnyard which was created in 2001 by the SNORT team. The SNORT team created Barnyard in order to allow SNORT to decouple output overhead of handling a variety of output formats, and improve its performance.</p>
IP Related Issues	None

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.3.1.12 WSO2 Balana

Attribute	Product Evaluation/Assessment
Version Reviewed	1.0.0
API Characteristics	Balana is an XACML 3.0 implementation library
Compliance with Recognized Standards	Compliant with XACML 1.0, 1.1, 2.0 and 3.0 specifications. Balana is XACML open source project which is based on the sunxacml http://sunxacml.sourceforge.net/
Support Availability	Balana is an open source software. Limited (free) support is available through the webpage http://xacmlinfo.org/category/balana/
Functionality	Balana is a verification engine supporting XACML
Licensing Cost	Barnyard2 is available under the free software/open source Apache license v2.0 (http://www.apache.org/licenses/LICENSE-2.0)
Integration Ease	As Balana is an implementation library, integration is basic
Scalability	None
Security Features	None
Trusted Source	Open Source (will be vetted through GD Canada Supply Chain Management eSAC process)
Ubiquity	Balana runs on all major operation systems
Extensibility	N/A
Source Characteristics	Balana source code is available to all at no charge. (https://svn.wso2.org/repos/wso2/trunk/commons/balana/modules/)
IP Related Issues	None

3.3.1.13 Effector Connectors

Attribute	Product Evaluation/Assessment
API Characteristics	Implemented with EIP using RTIF API and Effector APIs as required.
Compliance with Recognized Standards	Apache Camel Enterprise Integration Patterns
Support Availability	Support for both RTIF is provided. Online discussion forms available for Apache support.
Functionality	Provides the message systems, channels, constructions, routing, transformation, endpoints and system management to enable communication between the data source and the Integration Framework (ARMOUR Data Model). Apache Camel is commonly used throughout the industry
Licensing Cost	\$0

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Integration Ease	With access to ARMOUR APIs and Effector APIs, the integration effort should be minimal.
Scalability	To Be Determined (TBD)
Security Features	<p>Camel offers several forms and levels of security capabilities that can be utilized on camel routes. These various forms of security may be used in conjunction with each other or separately. The broad categories offered are:</p> <ul style="list-style-type: none">• Route Security - Authentication and Authorization services to proceed on a route or route segment• Payload Security - Data Formats that offer encryption/decryption services at the payload level• Endpoint Security - Security offered by components that can be utilized by endpointUri associated with the component• Configuration Security - Security offered by encrypting sensitive information from configuration files
Trusted Source	GD Canada developed.
Ubiquity	Not Applicable
Extensibility	Camel supports multiple languages in the DSL or XML Configuration for maximum extensibility
Source Characteristics	Apache Camel is Open Source
IP Related Issues	None

3.3.2 Technology Mapping

Table III shows where each ARMOUR Phase II capability is mapped to a COTS, OSS and Development effort.

Use or disclosure of this data is subject to the restriction on the title page of this document.

TABLE III Product Functionality Mapping

Functionality	RTIF	OWF	PostgreSQL	SNORT	NVD	MitVAL	AssetBank	COADS	SMARThawk	Dev / R&D	Technology Watch
Integration Framework	X										X
Data Sources											X
Infrastructure									X		X
Non-Infrastructure				X							X
Security			X								X
Operational										X	X
Data Source Connectors										X	
SMARThawk Connector										X	
SNORT Connector										X	
NVD Connector										X	
DRENet Data Source Connectors										X	
Database			X								X
Data Presentation		X								X	X
Data Normalization										X	
Cross-Source Correlation										X	X
Reachability Analyzer										X	X
Common Infrastructure Abstraction										X	X
Operations and Infrastructure Analyzer										X	X
Attack Path Generator					X					X	X
Attack Path Analyzer						X				X	X
Incident Analyzer										X	X
COA Analyzer							X			X	X
Semi-Automated Response										X	X
Automated Response										X	X
Effector Connectors										X	X

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.3.3 Technologies Under Evaluation

The following technologies are being considered for inclusion in ARMOUR. Further investigation into whether or not they are appropriate for ARMOUR will be done during implementation Phase 3.

3.3.3.1 OSSEC

Attribute	Product Evaluation/Assessment
Version reviewed	OSSEC 2.8
API Characteristics	OSSEC API is readily available on its website http://www.ossec.net/?page_id=19
Compliance with Recognized Standards	XML Format
Support Availability	OSSEC provides a wide variety of community support options available for our users, including: <ul style="list-style-type: none"> • OSSEC Users Mailing List – Installation, configuration and usage issues. • OSSEC Development Mailing List – Everything regarding development/code. • Getting Involved – How you can contribute to the project • Contact Us – All other questions: ossec@trendmicro.com • Commercial Support Options
Functionality	OSSEC is an Open Source Host-based Intrusion Detection System. For ARMOUR Phase 3, this includes performing log analysis and inspection (decoding). OSSEC has a log analysis engine that is able to correlate and analyze logs from multiple devices and formats.
Licensing Cost	OSSEC is available under the free software/open source GNU General Public License(GPL) version 2.
Integration Ease	OSSEC runs independently and provides several export methods suitable for variety integration requirements. A well-documented user manual can help with the use of these export methods.
Scalability	Client/server architecture
Security Features	
Trusted Source	Open Source (will be vetted through GD Canada Supply Chain Management eSAC process)
Ubiquity	OSSEC runs on all major operating systems, including Linux, UNIX (Solaris, *BSD), Mac and Windows

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Extensibility	OSSEC source code is available to anyone at no charge. OSSEC utilizes a modular plug-in architecture which is easily extensible. OSSEC rules are developed separately. New rules can be added using a simple rule description language.
Source Characteristics	OSSEC is an Open Source solution. In May 2009, Trend Micro acquired Third Brigade and the OSSEC project, with promises to keep it open source and free.
IP Related Issues	None

3.3.3.2 Cauldron (Jajodia & Noel, 2009) (Jajodia, Noel, Kalapa, Albanese, & Williams, 2011)

Attribute	Product Evaluation/Assessment
Version Reviewed	TBD – Current version unknown
API Characteristics	TBD
Compliance with Recognized Standards	Cauldron can import scans and other data from Nessus, Retina, FoundScan, Sidewinder firewall, Snort, and various commercial hardware firewalls/routers, and can interface with host-based asset inventory technologies such as Centennial Discovery and Symantec Altiris. Cauldron's vulnerability database can be populated with data from the National Vulnerability Database (NVD), Bugtraq, Symantec Deepsight, the Open Source Vulnerability Database (OSVDB), and the Common Vulnerabilities and Exposure (CVE) referencing standard. The attack graph produced has semantics similar to that produced by MulVAL (an AND/OR directed graph with vertices representing exploits and conditions).
Support Availability	TBD
Functionality	Cauldron interfaces with a number of tools that collect network information, including vulnerability scanners, host-based asset inventory tools, and firewalls/routers for connectivity information, and uses this information in conjunction with information regarding attack scenarios in order to produce an attack graph, visualizing the impact of vulnerabilities on the security of the network. Similar to MulVAL, this includes both individual vulnerabilities and combined vulnerabilities. Based on the graph, countermeasures in order to prevent the given attack scenario are presented. Previous versions of Cauldron could also be used in a disconnected scenario, with data being provided for ingestion.
Licensing Cost	TBD – Cauldron is a commercial product, and the cost of licensing is unknown.

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Integration Ease	TBD – There is not enough information currently available to assess ease of integration in ARMOUR. Note that within ARMOUR, the use of Cauldron would be as an Attack Graph Generator module, which may not make use of much of the data-collection functionality provided by Cauldron. It is likely that ARMOUR data will need to be converted to match Cauldron’s Data Model for processing.
Scalability	TBD – The authors of Cauldron have done work to enable the tool to scale (including aggregation ideas similar to the ARMOUR Common Infrastructure Abstraction module), but specific current performance details are not easily found.
Security Features	N/A
Trusted Source	Cauldron was developed by the Center for Secure Information Systems (CSIS) at George Mason University (GMU) in Fairfax, Virginia, USA, and is now (as of 2009) available as a commercial product from CyVision. Research was supported by Homeland Security Advanced Research Projects Agency (administered by the Air Force Research Laboratory/Rome), Air Force Research Laboratory/Rome, Federal Aviation Administration, Air Force Office of Scientific Research, and by the National Science Foundation.
Ubiquity	TBD – Early (pre-commercial) versions of Cauldron ran on Windows XP. It is not clear what the current requirements are. Aspects are apparently Java-based, so it seems likely that it will run on other operating systems.
Extensibility	TBD – Cauldron interfaces with a number of technologies, but it is unclear how easy it may be to add new “connectors” (if possible at all). This would be more important when passing data from the ARMOUR database into Cauldron for processing, as opposed to Cauldron collecting data from other sources.
Source Characteristics	Cauldron is distributed commercially by CyVision (http://cyvisiontechnologies.com/). They are partnered with GMU’s CSIS, and researchers from CSIS are involved in the company. CSIS was granted the designation as a “National Center of Academic Excellence in Information Assurance/Cyber Defence Research and Education” by the NSA for the academic years 2014 through 2021.
IP Related Issues	TBD – The licence terms for Cauldron are unknown. Note, several patents have been issued for aspects of Cauldron.

3.3.3.3 Nmap – Network Mapper

Attribute	Product Evaluation/Assessment
Version Reviewed	V6.46
API Characteristics	Nmap is a stand-alone application with an extensive list of command line options for scan configuration and output (including XML and “grepable” output).

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Compliance with Recognized Standards	<p>Nmap is capable of running on Linux, Windows, Mac OS X, Unix and BSD variants.</p> <p>It has a database of over 2200 “well-known” services that it associates open ports (TCP and UDP) with, and an additional database that uses to map the results of additional probing to in order to identify additional information, including application name, version number, hostname, device type, and so on. If possible, Nmap returns the Common Platform Enumeration (CPE) representation of the data.</p> <p>Nmap, if compiled to do so, can connect to SSL servers and obtain whatever information it can about the services behind the encryption layer.</p> <p>Nmap also uses “TCP/IP stack fingerprinting” to detect the names and versions of the OS running on remote hosts. Again, the CPE representation will be provided if available.</p>
Support Availability	<p>Nmap is provided without warranty, but is “well supported by a vibrant community of developers and users”. Archives for the Nmap development mailing list are available online (http://seclists.org/nmap-dev/).</p>
Functionality	<p>Nmap is a network discovery and security auditing tool. It discovers hosts on networks, and based on packet analysis, determines the services running on those hosts. It can also determine what operating systems (and versions) are running and information about firewalls in use.</p>
Licensing Cost	<p>Nmap is free and can be redistributed GNU General Public License version 2. For redistributing Nmap embedded in proprietary software, licenses are available. Cost of such licensing is unknown at this time.</p>
Integration Ease	<p>Nmap is distributed as a stand-alone program with both a GUI and command-line interface. In relation to ARMOUR, it would be used as a data source, and thus could likely be used unmodified. An appropriate Data Source Connector would have to be written in order to bring the results into ARMOUR for storage in the database, using the ARMOUR Data Model. Results can be output in a number of formats, including XML, which should help with processing.</p>
Scalability	<p>“Nmap has been used to scan huge networks of literally hundreds of thousands of machines.” “It was designed to rapidly scan large networks”.</p>
Security Features	N/A
Trusted Source	<p>Nmap is open source software that is included with many Linux/Unix distributions (including Redhat, Debian, Gentoo, FreeBSD, and OpenBSD), and has won several rewards.</p> <p>The software and vendor will be vetted through the GD Canada Supply Chain Management Electronic Software Authorization Checklist (eSAC) process.</p>
Ubiquity	<p>Source Code is available, compiles on Linux, Mac OS X, Windows, and several Unix platforms.</p> <p>Binaries are available for Windows, Linux (RPM), and Mac OS X.</p>

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Extensibility	<p>The need to extend Nmap is not anticipated, but if necessary, Nmap is open source, and thus modifications are possible. Note that the licence constraints will have to be considered.</p> <p>There are numerous Nmap related projects linked to from the Nmap website that extend Nmap, including parser libraries, frontends, and comparison tools.</p>
Source Characteristics	Nmap is open source, originally developed by Gordon Lyon. It was initially released in 1997, and is still actively developed (most recent release was in April 2014). Development is still led by the Nmap Project (through Insecure.Com LLC), but contributions from third parties are accepted for consideration for inclusion in future releases.
IP Related Issues	Depending on how Nmap is used, there may be issues with redistribution. Given the likely use only as a data source, it is anticipated that this will not come into play.

3.3.3.4 SCCM – System Center Configuration Manager

Attribute	Product Evaluation/Assessment
Version Reviewed	2012 R2
API Characteristics	<p>Microsoft provides a Software Development Kit (SDK) for SCCM. It enables automation of Configuration Manager via scripts and provides the ability to add features and extensions to the tool.</p> <p>This SDK outlines the various libraries that are available for developers to use with examples in C#, VBScript, and C++. Using WMI to access SCCM is also an option.</p> <p>It is expected that “large applications will likely be written in C#”. For use with languages that are not .NET Framework based, Microsoft recommends using the VBScript samples as examples for accessing SCCM via WMI.</p>
Compliance with Recognized Standards	SCCM is arguably the industry standard when it comes to configuration management of Windows hosts. As a Microsoft product, it interfaces well with other Microsoft products, and can be used to handle a variety of other applications and operating systems.
Support Availability	<p>SCCM is widely used and there are various avenues for both official and unofficial support.</p> <p>Governments will frequently have one or more support contracts with Microsoft in order to facilitate more-prompt responses to support queries.</p>

Use or disclosure of this data is subject to the restriction on the title page of this document.

Attribute	Product Evaluation/Assessment
Functionality	<p>SCCM is software that enables management of large groups of computers running a variety of operating systems. It provides a centralized mechanism for patch management, software distribution and deployment, remote control, maintaining hardware and software inventories, and in more recent versions, network access protection.</p> <p>In relation to ARMOUR, SCCM would be a data source that would provide information about hosts on the network ARMOUR is monitoring. Additionally, it is being considered as an effector that ARMOUR could trigger to deploy patches or disable software, for example.</p>
Licensing Cost	<p>It is expected that SCCM will exist on the networks on which ARMOUR will be deployed. At this time, it is not seen as a requirement for ARMOUR, and as such, licenses are not expected to be required as part of ARMOUR deployment.</p> <p>This evaluation may change as the investigation into SCCM and ARMOUR continues.</p>
Integration Ease	<p>Integration will require implementation of a data source connector to gather data from SCCM and possibly an effector connector to allow ARMOUR to trigger actions within SCCM. At this time, it is unknown if additional components will need to be implemented within SCCM in order to support this functionality.</p>
Scalability	<p>SCCM can handle a large number of clients, with single instances of various server roles handling between 4,000 clients (for a single distribution point, with a single site having up to 250 distribution points), up to System Health Validator points supporting up to 100,000 clients.</p>
Security Features	<p>SCCM integrates with Active Directory in order to control access to the application.</p>
Trusted Source	<p>Microsoft is one of the most well-known companies in the world. Their software is used by governments and organizations around the world.</p>
Ubiquity	<p>SCCM runs on Microsoft Windows, and can be used to manage computers running Windows, Windows Embedded, Mac OS X, Linux, Unix, and mobile operating systems including Windows Phone, Symbian, iOS and Android. Note that management of some of these systems require additional subscriptions (for example, Windows Intune is required to manage Windows Phone 8 and later, iOS 5.0 and later, and Android 2.3 and later).</p>
Extensibility	<p>The SCCM Software Development Kit (SDK) is available from Microsoft. It enables automation of Configuration Manager via scripts and provides the ability to add features and extensions to the tool.</p>
Source Characteristics	<p>Microsoft is among the world's largest and most valuable companies.</p>
IP Related Issues	<p>As SCCM-ARMOUR interaction would be as a data source/effector, there are no IP-related issues anticipated.</p>

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.4 ARMOUR Subsystems

This section discusses the design approach for each of the ARMOUR subsystems and key modules identified in the ARMOUR STS. The module names match those specified in the STS to maintain consistency. Some modules are combined to reduce complexity and overlaps in functionality.

3.4.1 Data Source Connectors

The ARMOUR IF SOA framework provides secure, standards compliant web service interfaces to the data sources. It uses service orientation best practices and principles to provide abstraction of the underlying data sources and implementations. Consumer applications and services do not directly access data, but rather access it through an intermediary or proxy whose role is to route the request to the actual data source or stored data generated by the data sources. ARMOUR IF provides out of the box connectors for relevant data sources, including Syslog (RFC3164) and Simple Network Management Protocol (SNMP). Table IV provides a non-extensive product-agnostic list of data sources under consideration for use in the ARMOUR System and the sections that follow present the approach to processing each data source type.

TABLE IV ARMOUR Data Sources⁵

Infrastructure Data Sources	Security Data Sources	Non-Infrastructure Data Sources	Operations Data Sources
Network Topology Discovery (e.g., SNMP)	Security Logs (e.g., Antivirus, Intrusion Detection Systems)	Commercial Vulnerability Feeds	Operations Data (simulated for demo)
Host Data Collectors (e.g., Windows Management Instrumentation)	Security Component configuration (e.g., Firewall, Intrusion Prevention System)	Threat Data	Asset Priority/ Mission Survivability
	Security Event and Incident Management	Software update lists	Organizational Information
	Operating System Security Event notifications	Geographic Data	

ARMOUR Data Source Connectors are provided OSGi Blueprints that are implemented with Apache Camel EIPs. A loosely coupled transformation function is included within the DSC that converts the data from the Data Source's Data Model to the ARMOUR Data Model.

⁵ The Phase 3 ARMOUR Data Model focuses on Phase 3 Data Sources and does not specify any data sources beyond Phase 3 including geographic data.

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.4.1.1 Infrastructure Data

ARMOUR provides interfaces to several Infrastructure Data Sources through custom built Data Source Connectors implemented in OSGi Blueprints. Collection of infrastructure data can be direct (push), scheduled, or on demand and may be through web services, direct file transfer or other means depending on the source. Once the data source data has been received by the ARMOUR DSC, the data is transformed (or normalized) to the ARMOUR model using the Transformation API specific to the data source. This data is used by ARMOUR to build a realistic, detailed, and dynamic model of the network including detailed host information, and is the basis to support Risk Treatment.

Infrastructure Data includes the following types of information:

1. Routing configurations;
2. Network topology;
3. Safeguard Data (Firewall configurations. IDS/IPS policies, etc.);
4. Hosts prone to frequent changes (e.g., laptop); and
5. Host Configuration/Identification information from all hosts (servers, workstations, end-user devices, network appliances and security appliances) including:
 - a. Host name;
 - b. IP address(es);
 - c. Media Access Control (MAC) address(es);
 - d. Host configuration;
 - e. Virtual vs Bare metal (non-virtual);
 - f. Installed software including:
 - i. Manufacturer;
 - ii. Product;
 - iii. Version;
 - iv. Sub-version;
 - v. Patches; and
 - vi. Software Vulnerabilities.

A Data Source Connector is implemented for the topology data through a Simple Object Access Protocol (SOAP) web service to allow ARMOUR to collect the topology data. After collection, the data is transformed from the Data Source Data Model to the ARMOUR Data model.

Host Configuration information describes the detailed configuration of individual assets as well as the vulnerabilities present within the individual asset. A number of Open Source/commercial IT inventory reporting software tools are available which can collect software as well as hardware information for Windows/Linux/Mac OS or SNMP enabled systems.

Use or disclosure of this data is subject to the restriction on the title page of this document.

For Windows-based hosts, collection of host information can be accomplished using the Windows Management Instrumentation (WMI) protocol in combination with Remote Procedure Call (RPC). In this case, the host information data source connects directly to the WMI service on the target host using RPC and gathers data remotely. Alternatively, information can be collected using WMI by installing agents on individual assets. WMI gathers software, hardware and registry data. WMI is available on Windows 2000 and higher Operating Systems.

For Linux/Mac OS, information is gathered by connecting a server (host information data source) to the host through Secure Shell (SSH) and uploading an agent to the host machine using Secure Copy Protocol (SCP) / Secure File Transfer Protocol (SFTP). Once uploaded, the scanning agent runs and creates a temporary file with the gathered information. Operating systems provide the agent with a number of utilities to enable it to gather the full host information. Once complete, the file can be transferred back to the server using SCP or SCP. After the transfer, the temporary file on the host machine is deleted.

A customized parser at the host information data source is required to parse the collected information before sending to the ARMOUR system. Based on available tools and their requirements, collection of host information will require investigation into potential methods. One or more Decision Analysis and Resolution (DAR) exercises should be performed to determine the best approach. For example, further investigation into the benefits of approaching host data collection using an agent based solution rather than an agentless solution can be facilitated by a DAR. A DAR can also be used to determine the best fit method(s) to query hosts for their information (Simple Network Management Protocol (SNMP), WMI, a combination of several protocols, etc.).

Automated collection of current vulnerability data from all hosts on the network can be accomplished using a Vulnerability Scanner. Some common features of these scanners are:

- a. Network based vulnerability scanners;
- b. Large number of vulnerability test cases which can be updated regularly. These test cases based on NVD;
- c. Exportable scan results in XML, CSV or proprietary format. Can be stored in a DB local to the scanner;
- d. Vulnerability supported for all type of OSs (Linux, Window, MAC, etc.) and for all types of devices (i.e., Work Stations, Servers, Routers, Virtual devices etc.); and
- e. Supported on Windows platform or Linux.

A DAR exercise will be performed during Phase 3 to identify the best fit Vulnerability Scanner Data Source. Evaluation and weighting criteria such as Platform Support, Open Source availability and output format will be identified for the DAR and the potential candidates will be evaluated against these criteria to determine the best fit solution.

The network Vulnerability Scanner is likely to detect vulnerabilities that can be exploited remotely more effectively than those for which local access is required. For ARMOUR's purposes (i.e., analysis of the results which help determine the host security posture value) the need to consider the reliability of a given vulnerability information is important (especially when

Use or disclosure of this data is subject to the restriction on the title page of this document.

local exploit are reported) in order to avoid selecting the wrong course of action. The DAR exercise should therefore include the scanner's competency (the type of test used) in accurately determining the existence of different types of software vulnerabilities (even when the software and versions are accurately identified).

A Data Source Connector will import the vulnerability scan data, based on a defined task. The ARMOUR Data Model must provide support for the storage and manipulation of collected infrastructure data. Some specific points for this data connector are:

- a. Customized parser depending upon the format of the vulnerability scan report;
- b. Since discovered vulnerabilities are specific to a host, the vulnerability will be linked with a Host identified during network topology discovery or during host configuration scanning;
- c. The ARMOUR Data Model will be updated to support this information. Entities such as Scan_Report, Vulnerability_Instance linked with Vulnerability_Information will be evaluated for inclusion; and
- d. This new Data Model will be linked with the existing ARMOUR Host Data Model.

3.4.1.2 Security Data

Security data is collected from various data sources available within the monitored network. Data Source Connectors developed for specific technologies (and specific Data Models) ingest and transform the data from the Data Source Data Model to the ARMOUR Data Model for storage in the ARMOUR database. ARMOUR should qualify the effectiveness and accuracy of the security data and determine how to account for any lack of accuracy so that the information can best be used by the ARMOUR Incident Analyzer module to resolve network event information into identifiable security incidents.

The following list identifies the types of Data Sources that ARMOUR will collect information from through developed Data Source Connectors:

- a. Network IDS/IPS;
- b. Endpoint Protection devices (Antivirus, Host based IDS/IPS);
- c. Firewalls;
- d. Universal log management appliance;
- e. Gateways; and
- f. Additional Security Devices to be defined upon clarification of DRENet equipment.

An ARMOUR data source connector was developed for SNORT. SNORT uses a signature-based approach to monitor network data flows, perform protocol analysis and apply rules against the data to detect cyber threats. When it detects attacks or probes, it generates alerts and provides several ways to log the alert messages.

Use or disclosure of this data is subject to the restriction on the title page of this document.

Once additional Security Data Source technologies available to DRENet are identified, a DAR will be performed to determine the Data Source Connectors required for development. Based on the devices available, an evaluation of the functionality and APIs will guide the development of various Data Source Connectors.

For example, sources of security data include configuration and policy data from devices such as firewalls and an Intrusion Prevention System (IPS). Mechanisms to retrieve firewall policies (e.g., from configuration files, access lists or third-party firewall management software) will be investigated to find the most suitable approach. IPS policy will also be used as input data source. IPS policies include type of network services provided, application supported on particular ports and use of inappropriate resources. Extraction of IPS policies may require a vendor specific tool as commercial or open source tools may not be available.

3.4.1.3 Non-Infrastructure Data

Non-infrastructure data is information not available directly from the network infrastructure, but that can be fed to the system by other means and can be used within the computational services modules to help generate and analyze attack graphs. Non-infrastructure data sources include vulnerability feeds and exploit data and are integrated via data source connectors developed for specific products/feeds. The collection of these sources can be scheduled or collected on demand. Vulnerability feeds are data feeds that can import the latest vulnerabilities from an identified source. These vulnerabilities mainly includes databases of security checklists, security related software flaws, misconfigurations, product names, and impact metrics. Each newly published vulnerability is represented in a standard format that includes a vulnerability Identification (ID), CVEs ID (if it exists), references to scanners plugin IDs, Common Vulnerability Scoring System (CVSS) score, description, links to relevant sources that describe the vulnerability and solutions. In addition, each vulnerability is modelled to include exploitation preconditions and effects, which can then be used in attack simulation.

There are two main public sources of vulnerability data

- a. National Vulnerability Database (NVD - <http://nvd.nist.gov/>); and
- b. Common Vulnerabilities and Exposures (<https://cve.mitre.org>).

These two sources are inter-linked. The Mitre CVE list only contains the standard identifier number with status indicator, a brief description, and references to related vulnerability reports and advisories and do not contain information such as risk, impact, fix information, or detailed technical information. NVD feeds in information from CVE and adds on to provide enhanced information such as fix information, severity scores, and impact ratings. NVD also provides advanced searching features such as by individual CVE-ID; by OS; by vendor name, product name, and/or version number; and by vulnerability type, severity, related exploit range, and impact.

Use or disclosure of this data is subject to the restriction on the title page of this document.

NVD data contains detailed information regarding vulnerabilities including:

- 1) Vulnerable products/software;
- 2) CVE ID;
- 3) Vulnerability published date;
- 4) CVSS Base Metrics
 - a. Exploitability Metrics;
 - i. Access Vector (Local, Adjacent Network, Network);
 - ii. Access Complexity (Low, Medium High); and
 - iii. Authentication (None, Single, Multiple).
 - b. Impact Metrics.
 - i. Confidentiality Impact (None, Partial, Complete);
 - ii. Integrity Impact (None, Partial, Complete); and
 - iii. Availability Impact (None, Partial, Complete).
- 5) Common Weakness Enumeration (CWE) – Type of weakness;
- 6) NVD may also provide a vulnerable configuration tree by CPE (to be confirmed); and
- 7) Reference information.

The only option to get the CVE list is download it in various format like HTML, Text, or CSV. The NVD provides various data feeds like SCAP data feed, XML data feed, RSS data feed that can be received live with a public internet connection, however, with a disconnected system like ARMOUR, download to an advantaged (public internet connected) computer followed by manual transfer to the ARMOUR system is likely required.

The ARMOUR Data Model will be extended to support the collection of the required Vulnerability information. An ARMOUR Data Model entity will be used for each extracted CVE from the data feed. The vulnerabilities list data source connector will be responsible for getting the CVE list, parsing out each CVE, and transforming it to ARMOUR Data Model.

Exploit data provides ARMOUR with information regarding the existence of known exploits for particular vulnerabilities and the details of the exploit such as the maturity and relative availability. Exploit data is readily available from the Exploit Database (www.exploit-db.com). Exploit Database is a CVE compatible database and CVE numbers are assigned to the individual exploit entries in the database, making it easy to associate vulnerabilities and exploits. Other exploit databases exist (commercial products) such as Core Impact or Canvass that include proprietary exploits. The design should clearly define the scope of exploit that need should be included in the proposed system to achieve an adequate level of accuracy or suggest that further analysis be performed in subsequent version to assess how to factor in possible gaps.

Once the format for ARMOUR non-infrastructure data has been fully defined, integration of the Data Source data to ARMOUR can be easily facilitated through the development of Data Source Connectors.

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.4.1.4 Operations Data

Operations data is collected to uniquely identify operations and to identify the relative importance of hosts, applications and services in their support of missions. It is comprised of intangible data (data not available by polling network devices) and based on specifics to the deployment of the network, and not necessarily data on the current security picture of the network. It is identified here as an individual data source as it is an important input to the ARMOUR System to ensure mission survivability from the network perspective. Operations data includes, but is not limited to, information such as mission objectives, asset prioritization for mission survivability, dependencies and threats. Generally, operations data is provided to the ARMOUR operators by the network's Operational Authority (OA) and is typically based on deployment considerations (i.e. threat environment, critical resources or groups of resources require to perform specific objectives, etc.). The data provided by the OA is input into ARMOUR by the operators and used in the analysis, presentation and decision capabilities of ARMOUR.

Operational Priority data is input to the system by an operator (or a collection of operators) with detailed knowledge of the organization, mission, network and ARMOUR system and/or through data collection from a data source. Automated methods of Operational Priority data collection using mission knowledge from sources such as the Order of Battle (ORBAT) will be evaluated during the detailed design and could be used to supplement the manually entered priority data. Evaluation of identified methods will be performed using a DAR and will include (but not limited to) criteria that weighs the availability of a data source, the content of the data provided by the data source and its applicability to be leveraged by ARMOUR. Automated collection of Operational data from identified data sources will be facilitated by one or more Data Source Connectors. Some of the challenges identified in the STS as areas for further investigation related to the Operations and Infrastructure Analyzer apply to the automated collection of Operational Data.

Data comprising the Operational Priority data includes:

- a. Unique operation identification;
- b. Metric to rate operational priority;
- c. Operation to operation dependencies;
- d. Operation to application services dependencies;
- e. Application services to application services dependencies;
- f. Application services to host services dependencies;
- g. Host services to host services dependencies; and
- h. Host services to host system dependencies;

Conjunctive (AND) and disjunctive (OR) dependencies will be identified for dependency relationships. Operational data collected by the ARMOUR system will be used in the Operations and Infrastructure Analyzer, and Risk Treatment to help identify priority assets and generate a forward directed hyper graph modelling the system dependencies.

Use or disclosure of this data is subject to the restriction on the title page of this document.

In order to accommodate this information, the ARMOUR Data Model will be extended to support Operational Priority information as described. Entities and relationships will be added and/or updated to provide support for this data.

Once the format for ARMOUR operations data has been fully defined, support can easily be added through connectors within the integration framework. Interfaces within the framework can also be created to facilitate the generation of operation definitions to supplement or operate in the place of the data feed, should it not be available.

3.4.2 Data Storage

The approach to developing the ARMOUR database is to leverage the data abstraction features and database-agnostic design of the integration framework to provide compatibility with all major database products, including MySQL, PostgreSQL, Oracle, MS SQLServer and others.

The data source connectors gather information regarding the state of the network such as data on events, information on operations and services, external vulnerability notices, connectivity, etc. Once this information is collected, it is sent to the database for storage and provisioning to various ARMOUR subsystems. ARMOUR IF provides the database transport interface via the ARMOUR IF Data Service which provides generic methods to persist, find, findAll, query, remove and purge via an API.

The data storage subsystem is comprised of the following modules:

- 1) Database;
 - a. Data Store;
 - b. Database Management Service (DBMS);
- 2) Knowledgebase; and
- 3) File Storage.

The database technology provided by ARMOUR is PostgreSQL and includes the Data Store and DBMS sub-modules. PostgreSQL is an Object-Relational DBMS (ORDBMS) and approaches data with an object-relational model that is capable of handling complex routines and rules. A flexible API helps provide development support for the Relational Database Management System including Java/Java Database Connectivity.

The ARMOUR Data Storage subsystem needs to include the ability to access CND knowledge for future reference, for example, a Wiki knowledgebase could be included in the ARMOUR Database and used by operators. The capability would enable operators to generate, store and access knowledge pertaining to the system and networks. The knowledgebase provides a living CND body of knowledge that can be referenced for previous incident information (i.e. operator generated notes, fixes, workarounds, COA reference information, etc.). The knowledgebase should be a web-based repository accessible by the users from the Data Presentation views. A file storage module supplements the knowledge base and allows for the centralized storage of all types of files. This can include Standard Operating Procedures, incident response procedures, policies, patch update files, threat and intelligence data, etc. The file storage is available to the operators through the Data Presentation views.

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.4.3 Data Presentation

The data presentation layer of the ARMOUR System is provided by the Ozone Widget Framework as described in subsection 3.1.4. Graphical and tabular widgets provide the operators with a GUI to interface with the system to perform operations particular to their role. The following list identifies the Widgets provided with the ARMOUR Solution. As the ARMOUR System evolves, Widgets may be added or removed to meet design considerations and Stakeholder feedback.

TABLE V Data Presentation Widgets

View	Graphical/Tabular	Primary Deployment Phase
Command View	Graphical	Phase 3
Operational View	Graphical	Phase 3
Network Topology View (Infrastructure)	Graphical	Phase 2
Incident Analysis View	Graphical	Phase 5
Attack Path (Graph) View	Graphical	Phase 3
COA View	Graphical	Phase 4
Operational Data Input View	Graphical	Phase 3
Alert Toolbar	Graphical	Phase 2
Verification Policy Input View	Graphical	Phase 2
Workflow ticket System	Graphical	Phase 5
OWF Administrator	Graphical	Phase 2
LDAP Administrator	Graphical	Phase 2
PG Admin (PostgreSQL)	Graphical	Phase 2
ARMOUR IF Admin	Graphical	Phase 2
Host Summary List (Inventory)	Tabular	Phase 2
Host Details List	Tabular	Phase 2
Host History	Tabular	Phase 3
Link Detail List (Reachability)	Tabular	Phase 3
Host Configuration List	Tabular	Phase 3
Incident Details	Tabular	Phase 2
Vulnerability Details	Tabular	Phase 3
Report Template	Graphical	Phase 2
Previously Run Reports	Tabular	Phase 2

Use or disclosure of this data is subject to the restriction on the title page of this document.

Each Widget is detailed in the following subsections. Additional widgets may be developed as needed for user interface capabilities that are needed to support other functionality, roles and use cases.

As described in the ARMOUR Detailed Design Document (GD Canada document No. 741349), six user roles are defined for ARMOUR:

1. Proactive Security Analyst;
2. Reactive Security Analyst;
3. Network Operator;
4. Systems/Configuration Manager;
5. Administrator; and
6. Commander.

Each user of the ARMOUR System is assigned to one of these roles. Each role is provided access to a specific subset of widgets in order to perform their duties. The assignment of widgets to user roles is articulated in the following table ('X' denotes widgets assigned to user role required to perform their duties. 'O' denotes widgets that the user has the option to display).

TABLE VI Assignment of Widgets to User Roles (2 sheets)

Widget	Proactive Security Analyst	Reactive Security Analyst	Network Operator	Systems/ Configuration Manager	Administrator	Commander
Command View	O	O	X	X		X
Operational View	X	X	X	X		X
Network Topology View	X	X	X	X		X
Incident Analysis View	O	X	O			
Attack Path View	X	X	O			
COA View	X	X	X			
Operational Data Input Window	O			X		X
Alert Toolbar	X	X	X	X	X	X
Verification Policy Editor				X		
Work Flow Ticket System	X	X	X			X
OWF Administrator					X	
LDAP Administrator					X	
PG Admin (PostgreSQL)					X	
ARMOUR IF Admin					X	

Use or disclosure of this data is subject to the restriction on the title page of this document.

TABLE VI Assignment of Widgets to User Roles (2 sheets)

Widget	Proactive Security Analyst	Reactive Security Analyst	Network Operator	Systems/ Configuration Manager	Administrator	Commander
Host Summary List	X	X	X	X		X
Host Detail List	X	X	X	X		X
Host History	X	X	X			
Link Detail List (Reachability)	X	X	X	X		X
Host Configuration List	X	X	X	X		X
Incident Summary List	X	X	X			
Incident Details	X	X	X			
Vulnerability Details	X	X	X			
Report Template	X	X	X	X		X
Previously Run Reports	X	X	X	X		X

3.4.3.1 Data Presentation Interfaces

As with all subsystems, the Data Presentation subsystem interfaces with the Integration Framework as the message bus. The Integration Framework provides the medium for subsystem to subsystem and module to module communication.

The Data Presentation subsystem interfaces with the Data Storage subsystem via the Integration Framework to:

- Receive and display Infrastructure, Non-Infrastructure, Security and Operational Data;
- Provide input data to be persisted to the database (e.g., data verification policies);
- Query the database for specific data (e.g., reports);
- Retrieve computed/analyzed data persisted by the computational services modules; and
- Utilize the knowledgebase/file storage modules to maintain an up-to-date repository of historical events, policies/procedures and reference files.

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.4.3.2 Data Presentation View Capabilities – Graphical Displays

Graphical displays provide the user with a visual representation of the network and CND events correlated to the nodes. The design intent of the graphical displays is to support rapid indications of critical information that illustrate areas of immediate concern to the operator. The graphical displays include support for:

- a. Both two and three dimensional graphics displays with respective two and three axis manipulation of the objects within the display;
- b. Common user feature manipulation of display attributes
 - (1) Resizing;
 - (2) Undock windows;
 - (3) Rearrange window locations;
 - (4) Cascading windows; and
 - (5) Tiling windows.
- c. Illustration of both aggregate information (e.g. operational and operational service level views) and lower level detail (e.g. individual host data).

The graphical display's also support graphical data entry by the operator and includes support for:

- a. Operator selection or de-selection of icons, checkboxes, radio buttons, data drop-downs and menu items;
- b. Operator requests for additional information based on selections;
- c. Operator drag-and-drop or repositioning of icons; and
- d. Operator creation of links or relationships between icons.

3.4.3.2.1 Object Representation

Icons within the graphical windows are used to represent objects and data overlays. Icons are distinct and intuitive representations of operations, services and infrastructure as well as data overlays.

Data overlay icons will be used to represent:

- a. Uncertain status (e.g., hosts identified but corresponding software information is not available);
- b. Security metric and degree of dependency values;
- c. Presence of vulnerabilities;
- d. Likely occurrence of compromise;
- e. Sequence within an attack path;
- f. Attack value (e.g., criticality of a host within an attack path); and
- g. Course of action implication (e.g., effect of selecting or deselecting a particular course of action).

Use or disclosure of this data is subject to the restriction on the title page of this document.

Icons include additional visual details/attributes regarding the component they represent such as color, size, shadow and flashing, and provide the operator with a visual indication of various characteristics of the object (i.e. asset compromised, high value asset, operational criticality, etc.).

Relationships between objects are displayed through the use of connecting lines or arrows. Similarly to icons, relationships can include additional visual attributes to identify the characteristics of the relationship between objects such as line thickness, color and style.

3.4.3.3 Data Presentation View Capabilities – Tabular Displays

A tabular display is a forum for presenting ARMOUR data in a spread sheet like format (columns and rows) to the operator. Tabular displays compliment the graphical displays in the sense that they present information that is complimentary and representative of the graphical displays. For example, a force directed graph would be too busy if all the attributes associated with a node were displayed. Should an operator want to view detailed attributes of the nodes within the current view without cluttering the topology, a tabular view could be initiated that would provide the requested attributes/details in a different window (widget).

Tabular Displays provide access to the following information:

- a. Pre-defined tables for commonly accessed data tables and fields;
- b. Custom tables based on user selection of data tables and fields;
- c. Common user feature manipulation of tabular data (e.g. data sorting, data filtering, data selection, data marking, etc.); and
- d. Data reporting through pre-defined and custom list generation.

Similar to graphical data entry, the GUI also supports tabular data entry by the operator. Tabular entry of data supports operator entry, modification and deletion of alphanumeric information.

In order to provide pre-defined and custom tabular display capabilities, ARMOUR provides a generic list class that can handle a configuration file for each request that queries the database and displays the information in the requested format.

Access to pre-defined tables or attributes for populating custom tables, as well as read/write/modify permissions will be restricted based on the user role's access to the Data Model objects/attributes.

3.4.3.3.1 Tabular Details

The following list identifies the list of tabular details supported by ARMOUR:

- a. Host Details: Detailed information about a host object including unique host identifiers, software installed, software services running, Internet Protocol (IP) Addresses, protocols and active ports;
- b. Vulnerability Details: Detailed information about specific vulnerabilities. This includes common details as available from the National Vulnerability Database (NVD) and additional details from unique sources such as North Atlantic Treaty Organization (NATO) Cyber Defence Data Exchange and Collaboration Infrastructure (CDXI);
- c. Exploit Details: Detailed information concerning the relative availability and maturity of exploits corresponding to known vulnerabilities;
- d. Course of Action Lists and Selection: Detailed information about the set of actions included in each of the prioritized courses of action resulting from either the proactive and reactive security analysis functions. This table supports viewing, selection and execution of the course of action, as well as a corresponding graphical view of the impact of selecting or deselecting the course(s) of action;

3.4.3.4 ARMOUR TD Views

The following subsections describe the views provided by the ARMOUR TD in order to achieve all of the functionality required to execute the mission and support the OODA loop. These views support both Graphical and Tabular Displays. Some views are discussed together as they represent a more complete capability when grouped together.

3.4.3.4.1 Infrastructure View

The Infrastructure View is the primary interface to view the network topology. This view is provided and ARMOUR provides both graphical and tabular widgets to represent the infrastructure. The infrastructure view is supported by the following widgets:

1. Network Topology View (Topology);
2. Host Summary List (Inventory);
3. Host Detail List (Node Detail);
4. Host Configuration List;
5. Host History;
6. Link Detail List (Reachability);
7. Incident Summary List (Incident);
8. Incident Details;
9. Alert Toolbar; and
10. Vulnerability Details.

Use or disclosure of this data is subject to the restriction on the title page of this document.

The Infrastructure View is one of the views that supports the Observe Phase of the OODA Loop.

3.4.3.4.1.1 Topology, Inventory and Node Details Widgets

The graphical widget, titled the “Topology” widget, provides the operator with a visual representation of the physical network topology - the network devices and the links between the hosts. Overlays will be used in this view to provide visual indication of the node’s status (e.g. Vulnerable, Compromised, etc.). The Topology widget can also display the reachability graph connectivity determined by the reachability analyzer, using graphic overlays on the topology GUI. Details of the reachability graph are defined in the Link Detail List..

The “Inventory” widget provides a list (tabular) of the hosts discovered by the system and represented in the topology widget. This list provides some high level information about each host including:

1. The hostname;
2. A brief description;
3. Whether or not it is a managed host;
4. The classification; and
5. The IP Address

Included in this view is the option to edit the data contained in the database tables. The widget provides some aggregation of data into summary form by identifying the total number of assets contained within the table. Additional capabilities, such as tooltips that display aggregate data in summary form will be included for each column that is applicable (e.g. Classification, Managed).

Providing more detailed information is the “Node Detail” list widget. This widget displays the detailed information collected on the host. This includes all the information contained in the Inventory list as well as additional information pertinent to the host configuration.

Figure 12 depicts the ARMOUR deployment showing all three supporting Infrastructure views.

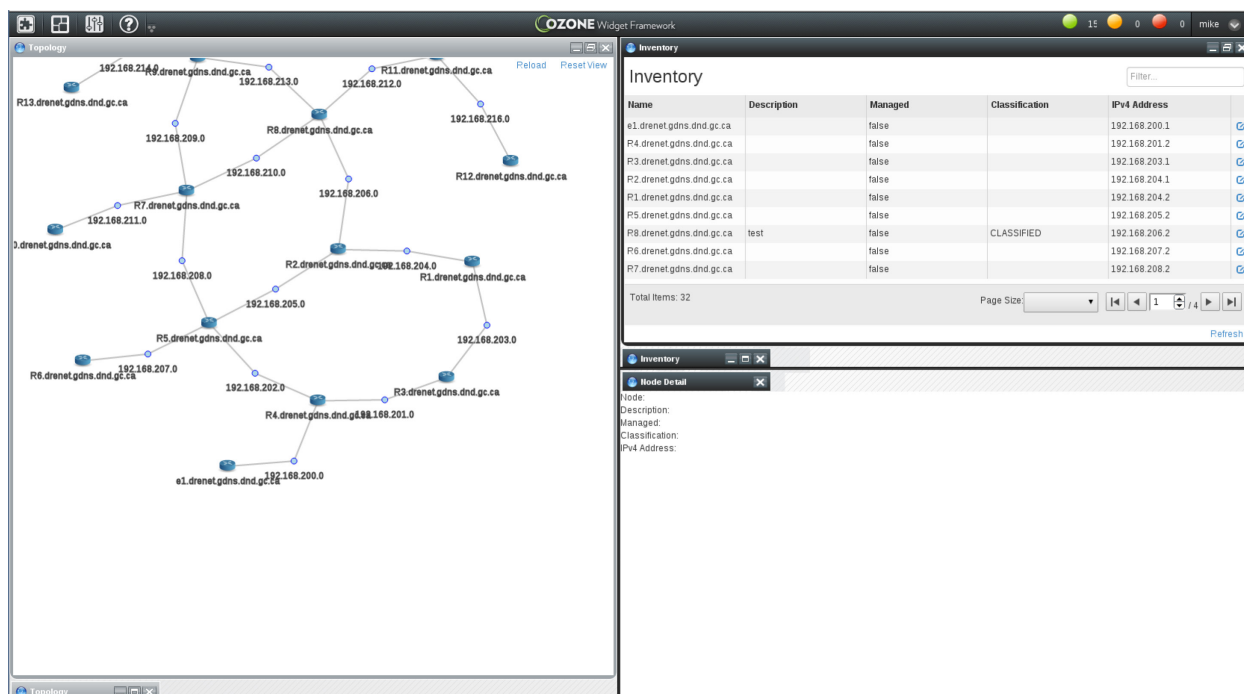


FIGURE 12. Infrastructure View

3.4.3.4.1.2 Host Configuration List Widget

The Host Configuration list widget provides detailed information on software, hardware and registry information (as described in subsection 3.4.1.1). When an operator selects a host from the Topology widget or the Inventory widget, the Host Configuration details are automatically populated in the Host Configuration list widget. This document will be updated with screen captures in subsequent releases.

3.4.3.4.1.3 Host History List Widget

The Host History list widget includes historic information pertinent to a selected host. Historic information may include previous vulnerabilities, installed patches, resolved incidents, up-time and more.

3.4.3.4.1.4 Link Details List (Reachability) Widget

The Link Detail List provides the reachability details that describe the logical links between nodes within the network. This list identifies connectivity between source and destination hosts including port and protocol information. Additional details on this widget will be provided in future versions of this document.

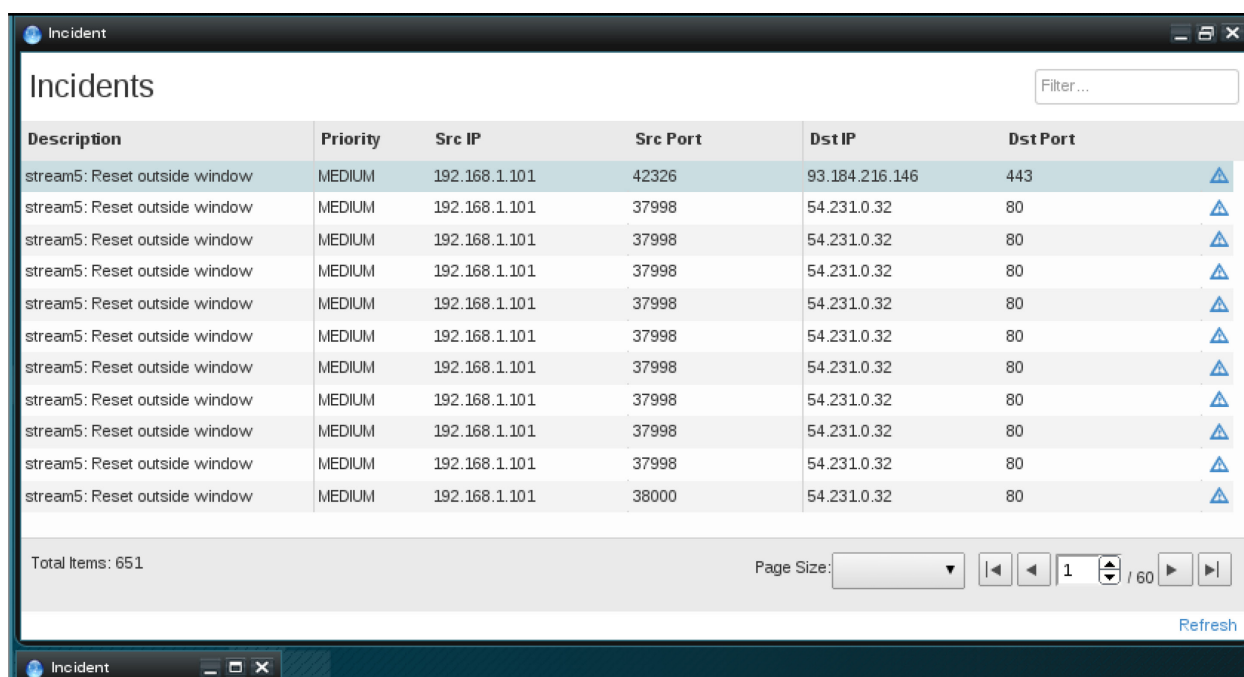
Use or disclosure of this data is subject to the restriction on the title page of this document.

3.4.3.4.1.5 Incident, Incident Details and Alert Toolbar

The Incident Summary List widget displays the summary list of security incidents that are contained within the ARMOUR database table. From this widget, Operators can view high level details regarding any of the observed security incidents.

Supplemental to the Incident Summary list is the Incident Detail list widget. This widget provides more detailed information on the highlighted incident from the Incident Summary list. The Incident Summary list and Incident Details list are linked views and as such, selecting an object in the Summary list will automatically populate the Details list with information pertaining to that incident.

Figure 13 depicts the Incident Summary widget and Figure 14 depicts the Incident Details widget.



Description	Priority	Src IP	Src Port	Dst IP	Dst Port	
stream5: Reset outside window	MEDIUM	192.168.1.101	42326	93.184.216.146	443	▲
stream5: Reset outside window	MEDIUM	192.168.1.101	37998	54.231.0.32	80	▲
stream5: Reset outside window	MEDIUM	192.168.1.101	37998	54.231.0.32	80	▲
stream5: Reset outside window	MEDIUM	192.168.1.101	37998	54.231.0.32	80	▲
stream5: Reset outside window	MEDIUM	192.168.1.101	37998	54.231.0.32	80	▲
stream5: Reset outside window	MEDIUM	192.168.1.101	37998	54.231.0.32	80	▲
stream5: Reset outside window	MEDIUM	192.168.1.101	37998	54.231.0.32	80	▲
stream5: Reset outside window	MEDIUM	192.168.1.101	37998	54.231.0.32	80	▲
stream5: Reset outside window	MEDIUM	192.168.1.101	37998	54.231.0.32	80	▲
stream5: Reset outside window	MEDIUM	192.168.1.101	37998	54.231.0.32	80	▲
stream5: Reset outside window	MEDIUM	192.168.1.101	38000	54.231.0.32	80	▲

Total Items: 651

Page Size: | 1 / 60

FIGURE 13. Incident Summary Widget

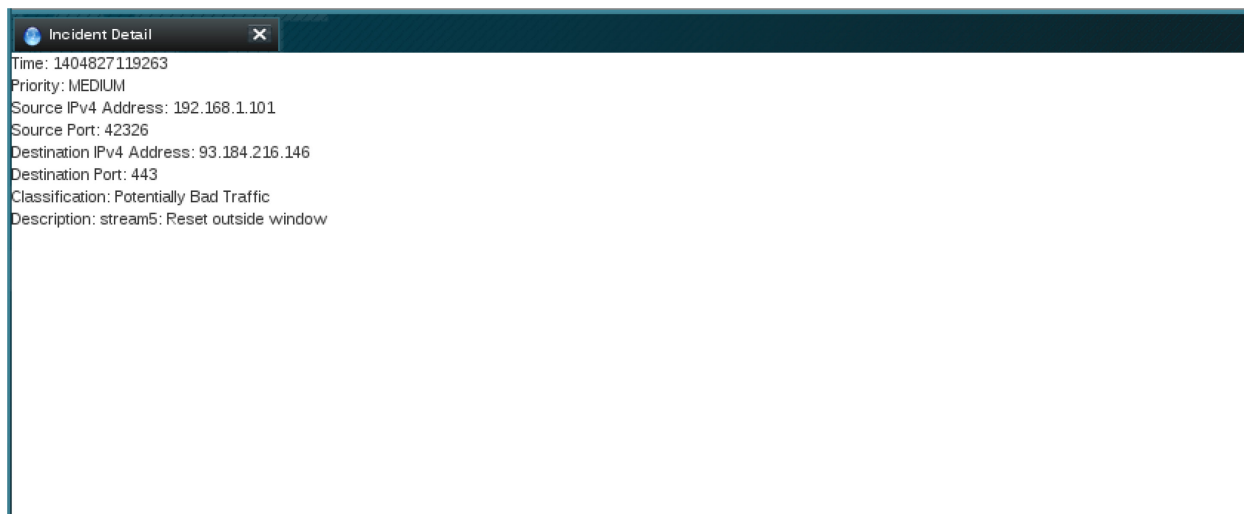


FIGURE 14. Incident Details Widget

When security events are observed, an Alert toolbar located in the banner bar of the OWF framework provides a visual indication to the user that a security event of interest has occurred. The Alert toolbar provides three levels of severity indication:

1. Low Severity Events – Events of low severity are represented by the Green sphere in the alert toolbar.
2. Medium Severity Events - Events of medium severity are represented by the Amber sphere in the alert toolbar.
3. High Severity Events - Events of high severity are represented by the Red sphere in the alert toolbar.

Each indicator includes a counter next to it to identify how many events of that severity have been observed. The user is able to click on the sphere and bring up a list of all events that have been grouped as that severity.

The Alert toolbar is depicted in Figure 15



FIGURE 15. Alert Toolbar

3.4.3.4.1.6 Vulnerability Details Widget

The Vulnerability Details widget provides detailed information on vulnerabilities and exploits present on selected hosts. When an operator selects a host from the Topology widget or the Inventory widget, the vulnerability and exploit details are automatically populated in the Vulnerability Detail widget. This document will be updated with screen captures in subsequent releases.

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.4.3.4.2 Operational View

The Operational View module of the ARMOUR System provides interfaces for the creation, management, and monitoring of missions. This interface fully integrates operational views within the network architecture interface, as well as allowing them to be viewed and managed separately.

The ARMOUR System allows operations to be viewed and managed with variable levels of granularity. Once an operation has been established, its assets, dependencies and status can be viewed at any time. This includes vulnerability, exposure, and risk data, as well as connectivity and defence posture information. Operations can also be grouped into larger organizational units to produce one or more command structures. These command structures provide the means to gauge the overall status and security posture of the entire command. Operations are searchable and filterable to facilitate display of only the resources that are most relevant to the operator at any given time. Additionally, display of specific attributes and details may be toggled on or off at the discretion of the operator.

Widgets included in the Operational View are:

1. Command View
2. Operations View
3. Operational Data Input View

The Operational View is one of the views that support the Observe Phase of the OODA Loop.

As Stakeholder feedback is received, the design of the widgets supporting the Operational View is likely to evolve and drive changes to the GUI design.

Infrastructure view and operational view (with security event notification) should be accessible from one dashboard. The infrastructure view should have a filter to only look at assets associated with specific missions. Default views should be defined if no current mission data exists or is specified.

3.4.3.4.2.1 Command View

The "Command View" widget shows a high level summary of mission resources and their security status. It provides the operator with a view of the defensive posture from the Operational Command perspective. Cyber security relevant information is displayed to indicate the status of operational icons. Drilling down into the structures navigates the user to more granular information provided by the Operations View. Figure 16 depicts a wireframe that will be used to scope the look and feel of the developed view.

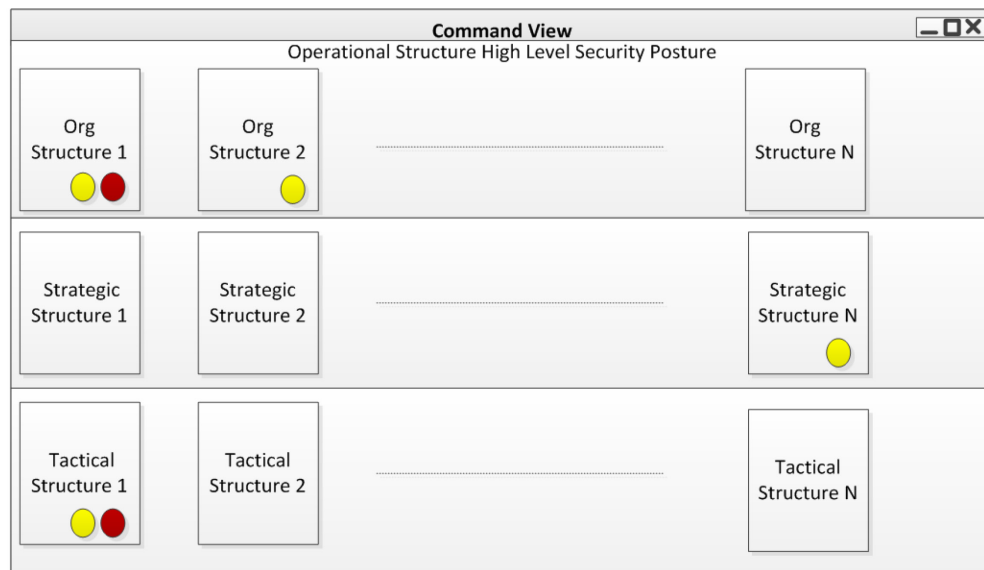


FIGURE 16. Sample Command View Wireframe

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.4.3.4.2.2 Operations View

The Operations View provides a lower level view of Operational cyber security status to the operator. At the highest level of abstraction, the Command View provides Organization wide compartments. Within the Operations view, Mission structure hierarchy can be navigated to the next level of detail. Security status indicators are displayed in aggregate at the higher element. In addition to the security status indicators, Operational Criticality indicators can be used to indicate the relative priority of particular elements. For example, the blue shaded Company in Figure 17 has been identified as a critical element.

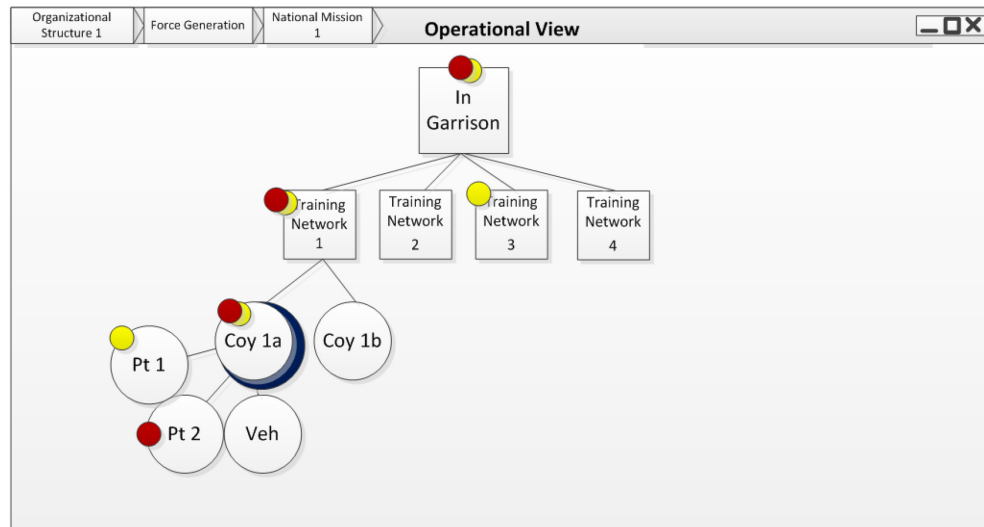


FIGURE 17. Sample Operations View Wireframe

The lowest level of the Operations View is the abstracted infrastructure (not shown in Figure 17). At the lowest level, the Operator can identify infrastructure elements to display the results of the Attack Path Analysis.

3.4.3.4.2.3 Operational Data Input View

The Operational Data Input widget provides the interface for the user to apply Operational Data to the network. Through manual data entry within the widget, the user is able to:

1. Create, modify, delete and save missions;
2. Assign, modify, delete and save assets to missions; and
3. Add, modify, delete and save dependency relationships.

Figure 18 provides a mock-up of the Operational Data Input widget.

Mission Identification & Operational Data

Mission Data

Mission ID (Operational Identifier):

Operational Priority:

ADD DELETE SAVE CANCEL

Define Mission Assets

Mission

Mission ID

Asset

Asset UID:

Asset Description:

ADD DELETE SAVE CANCEL

Define Dependency Data

Mission

Mission ID

Asset

Asset UID:

Asset Description:

Operation to Operation

Operation 1

Operation 2

Operation to Application Service

Operation

Application Service

Application Services to Application Services

Application Services to Host Services

Host Services to Host Services

Host Services to Host System

ADD DELETE SAVE CANCEL

FIGURE 18. Sample Operational Data Input View Wireframe

3.4.3.4.3 Security Action Status View

The Security Action Status View provides the ability to virtually implement and analyze proposed changes to the network. It also provides the ability to view the impact to operational and infrastructure views affected by pending COA requests.

Use or disclosure of this data is subject to the restriction on the title page of this document.

This view is initiated by selecting the speculative network model within the Infrastructure View. Within this model, the operator has complete flexibility to implement or rollback changes to any available network property, including:

- a. Marking vulnerabilities as fixed or ignored;
- b. Changing routes or physical network connectivity; and
- c. Changing firewall rules.

While the ARMOUR back-end supports the implementation of simulated data, simulation of states and data, and the analysis of the impact will be defined and developed in Phase 5. Additional details will be documented in future revisions of this document.

The Security Action Status View is one of the views that support the Orient Phase of the OODA Loop.

3.4.3.4.4 Attack Path View

The Attack Path View provides the operator with the ability to create and modify threat definitions, prepare and execute attack simulations, and view their results graphically.

Generation of the attack graph is accomplished through selection of a threat origin and attack target from within any view that can display individual or abstracted hosts (e.g., Topology, Inventory, Operational View). This origin defines the starting point for the attack as well as any of the attacker attributes that are relevant to the simulation. These origins can be created or modified from within this interface at any time.

Once an attack has been generated, it can be viewed within the attack explorer as a AND/OR directed graph. This graph features colour-coded and weighted links that visually describe not only the paths taken, but the likelihood of those paths being taken. For each step in the attack, a detailed description is given that includes the host, status, exposure, criticality, associated vulnerability and calculated risk. An overview for the entire graph features a count and list of attack steps / vulnerabilities exploited, unique vulnerability types, ports or services used, and effected hosts. It is also possible to view lists of routes and access policies that have allowed the attacks to occur at the network level.

Future revisions of this document will include screen captures of the widget.

The Attack Path View is one of the views that support the Orient Phase of the OODA Loop.

3.4.3.4.5 COA View

The COA view provides the means to create and edit COAs, enact them either through semi-automated or manual response, and monitor their status.

Use or disclosure of this data is subject to the restriction on the title page of this document.

As the primary interface to the COA library, the COA view will be responsible for the creation and maintenance of all COA definitions. Specifically, the operator will be able to set:

- a. External to internal mappings;
- b. Implementation cost value;
- c. Implementation risk value;
- d. Internal commands; and
- e. Associated rollback COA.

As part of the semi-automated response, the COA View will also present lists of COAs to an operator for approval. In making their decision, the operator will be provided with all assigned costs, and be able to cross reference each of the COAs with data from the architectural view as well as historical response data from the knowledge base. At this time, the operator may also choose a COA from the database for implementation if those suggested are unsatisfactory. After the COAs have been approved (either manually (semi-automated) or fully automated), the COA View will provide the means to monitor the status of their implementation in real time. If at any time there is a problem or change of plan, the operator will be able to select any active COA for cancellation, as well as implement the associated rollback COA.

The COA View is one of the views that support the Orient Phase of the OODA Loop. It is also the sole provider for the Decide and Act Phase view. Future revisions of this document will include additional detail on the COA View.

3.4.3.4.6 Incident Analysis View

The Incident Analysis View is provided by technologies that are integrated within the ARMOUR IF. These technologies create a visualization layer for incoming network security events that is flexible, responsive, and attractive in support of incident analysis activities. It provides the means to display and analyze all of the incoming security event data and includes features such as:

- a. Sortable, searchable lists of all incoming security events;
- b. Customizable charts and graphs capable of long-term historical correlation; and
- c. Short-, medium-, and long-term trending displays.

The Incident Analysis View is one of the views that support the Observe and Orient Phases of the OODA Loop.

3.4.3.4.7 Supporting Views

ARMOUR provides several additional supporting views that support system capabilities and are described in the following subsections.

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.4.3.4.7.1 Verification Policy Input

ARMOUR provides the ability for the user to generate, modify, edit and remove data verification policies through the Verification Policy Input window. In this widget, the user is presented with a list of current verification policies being applied. A user can add new or editing existing policies through the Edit Rule window by selecting the appropriate action within the window. Figure 19 depicts this widget.

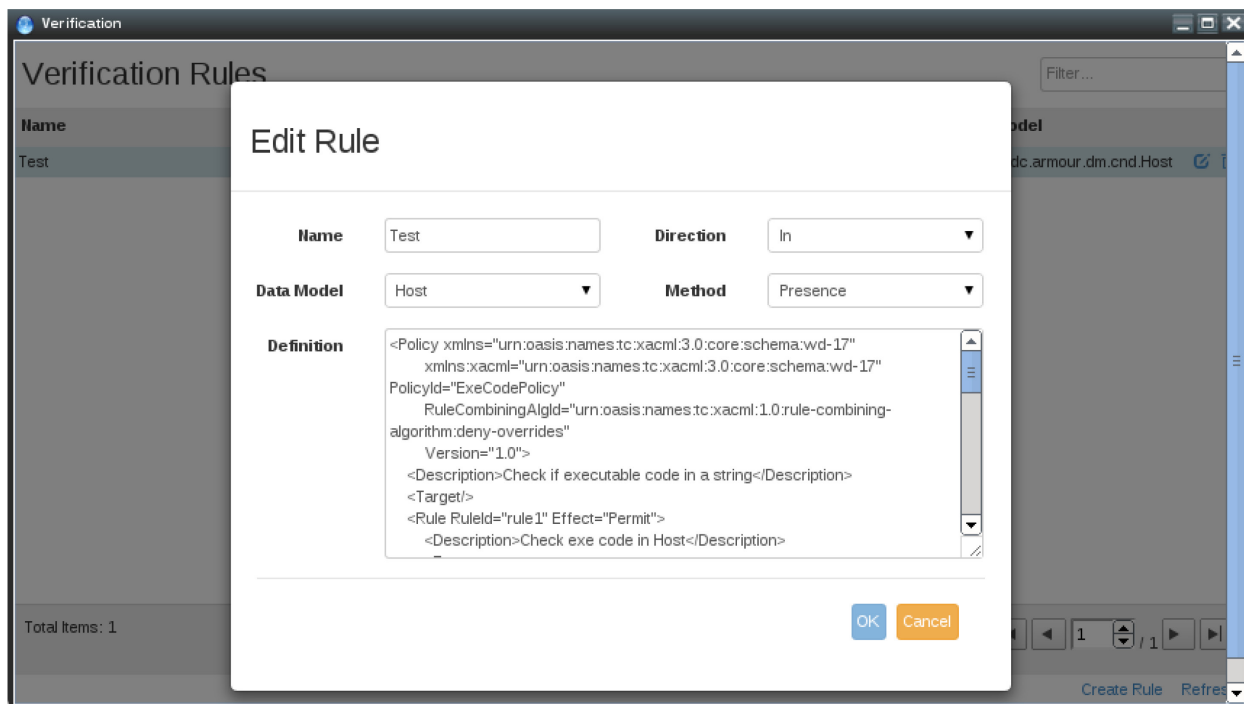


FIGURE 19. Verification Policy Input Widget

3.4.3.4.7.2 Work Flow Ticket System

Details on the Work Flow Ticket System will be included in future revisions of this document.

3.4.3.4.7.3 Administrator Views

Administration GUIs for OWF are provided as part of the OWF suite. LDAP, PostgreSQL (PG Admin) and ARMOUR IF administration is performed using the third-party consoles. Figure 20 depicts the OWF Administration page.

The screenshot displays the OWF Administration View, which is divided into four main panels. Each panel contains a table of data and a search bar at the top. The top of the window shows the 'OZONE Widget Framework' title bar and system tray icons.

Users Panel

Full Name	Last Sign In	Groups	Widgets	Dashboards	Stacks
daniel	07-25-2014 02:39	3	13	1	0
DEFAULT_USER		0	0	0	0
efim	07-24-2014 13:04	2	10	2	0
mhm	07-24-2014 09:20	2	9	2	0
mattin	07-24-2014 12:56	2	11	2	0
mike	07-24-2014 09:14	2	11	2	0
roger	07-24-2014 13:30	2	11	2	0

Page 1 of 1 | Displaying 1-7 of 7

Groups Panel

Group Name	Users	Widgets	Stacks
Commander	1	9	0
NetworkOperator	1	11	0
OWF Administrators	1	11	0
OWF Users	6	0	0
ProactiveAnalyst	1	11	0
ReactiveAnalyst	1	11	0
SystemAdministrator	1	13	0
SystemConfigurationManager	1	10	0

Page 1 of 1 | Displaying 1-8 of 8

Widgets Panel

Title	URL	Users	Groups
AlertHigh	http://10.0.0.1:9292/#/alert?type=ERROR	6	6
AlertLow	http://10.0.0.1:9292/#/alert?type=LOW	6	6
AlertMed	http://10.0.0.1:9292/#/alert?type=WARN	6	6
Dashboard Editor	admin/DashboardEdit.gsp	1	2
Group Dashboards	admin/GroupDashboardManagement.gsp	1	2
Group Editor	admin/GroupEdit.gsp	1	2
Groups	admin/GroupManagement.gsp	1	2
Incident	http://10.0.0.1:9292/#/incident	3	3
Incident Detail	http://10.0.0.1:9292/#/incident/detail	3	3
Inventory	http://10.0.0.1:9292/#/inventory	5	5
Node Detail	http://10.0.0.1:9292/#/inventory/node	5	5
Previously Run Reports	http://10.0.0.1:9292/#/report/runs	5	5

Page 1 of 1 | Displaying 1-22 of 22

Group Dashboards Panel

Dashboard Title	Groups	Widgets
Administration	2	4
NetworkOperator	1	7
ProactiveAnalyst	1	7
ReactiveAnalyst	1	7
SystemConfigurationManager	1	6
Commander	1	5
Commander (Sandbox)	1	2
SystemConfigurationManager (Sandbox)	1	2
NetworkOperator (Sandbox)	1	2
ReactiveAnalyst (Sandbox)	1	2
ProactiveAnalyst (Sandbox)	1	2

Page 1 of 1 | Displaying 1-11 of 11

FIGURE 20. OWF Administration View

Use or disclosure of this data is subject to the restriction on the title page of this document.

Figure 21 depicts the LDAP Administration view. This view is provided by the third party tool - 389DS Management Console.

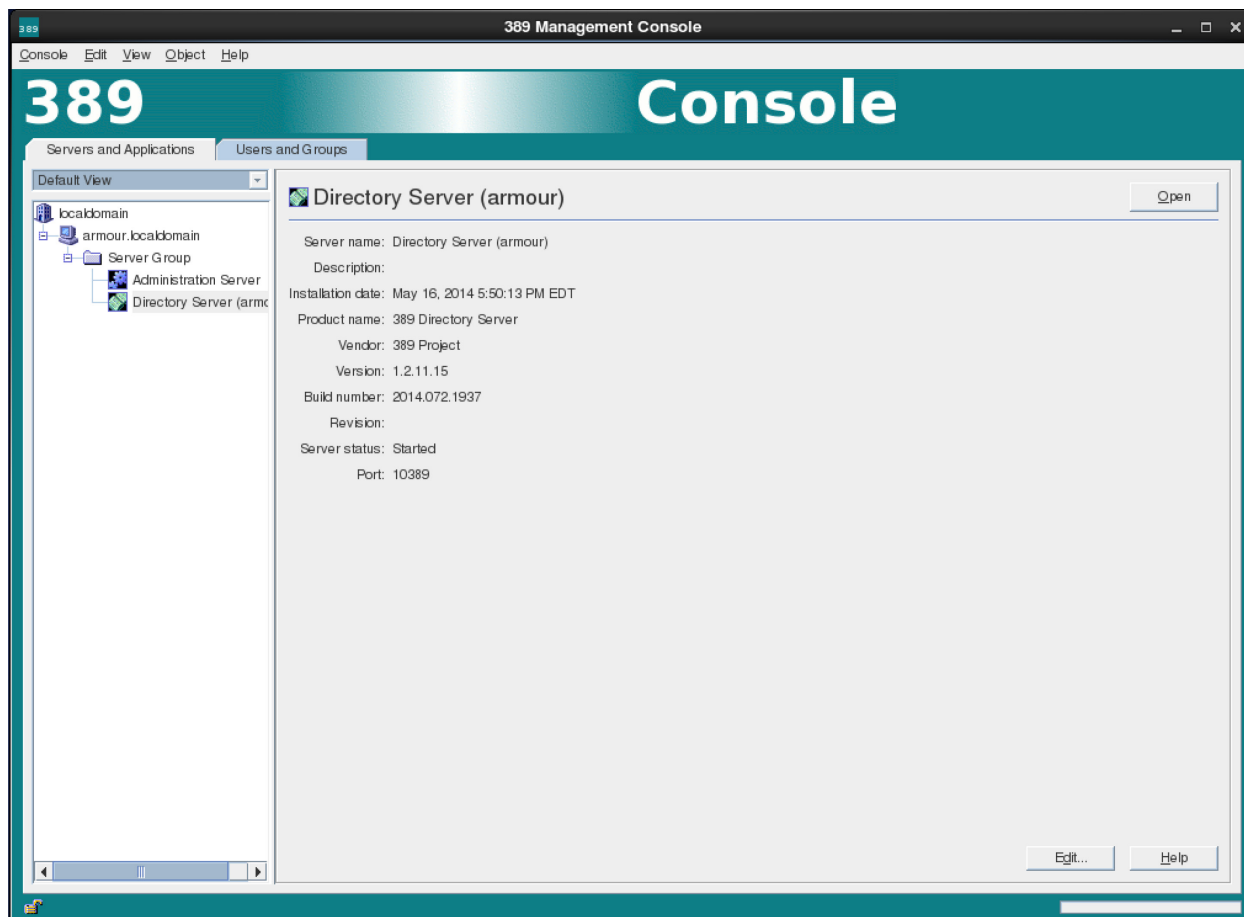


FIGURE 21. LDAP Administration View - D389 Management Console

Use or disclosure of this data is subject to the restriction on the title page of this document.

Figure 22 depicts the PostgreSQL Database Administration View. This view is provided by the third-party PG Admin tool.

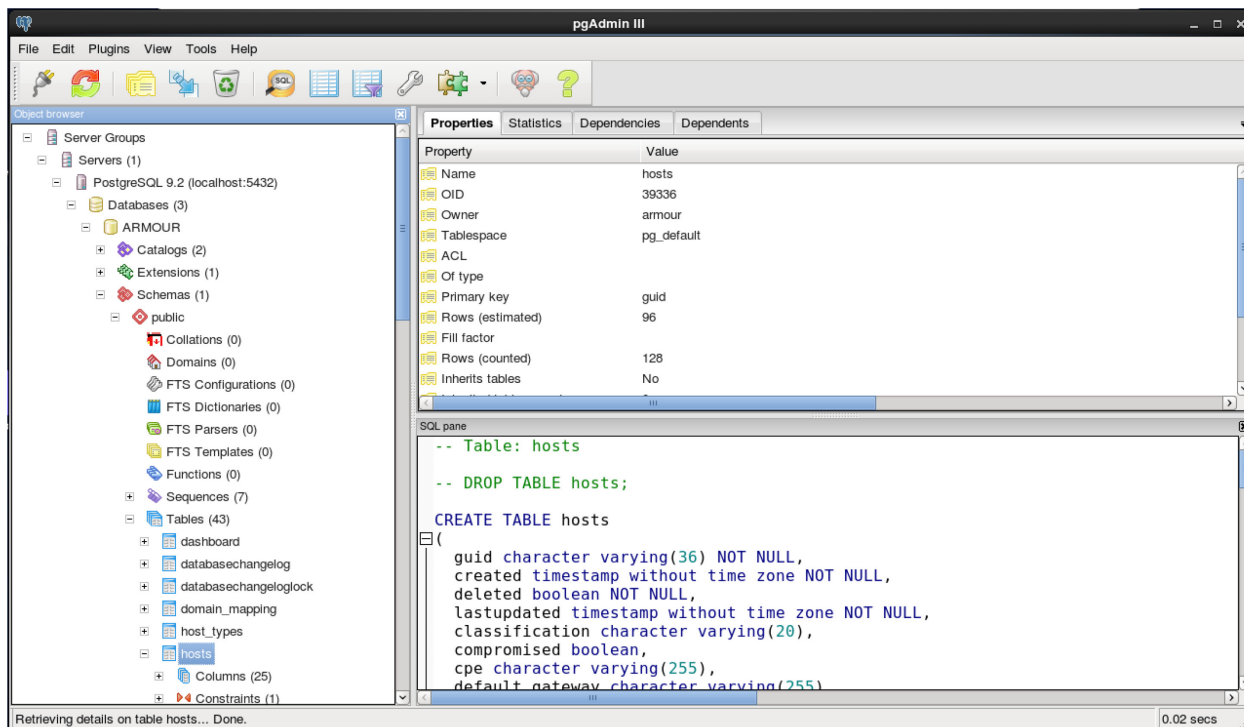


FIGURE 22. PostgreSQL Administration – PG Admin

Use or disclosure of this data is subject to the restriction on the title page of this document.

Figure 23 depicts the ARMOUR IF Administration view. This view is provided by the third-party tool HawtIO.

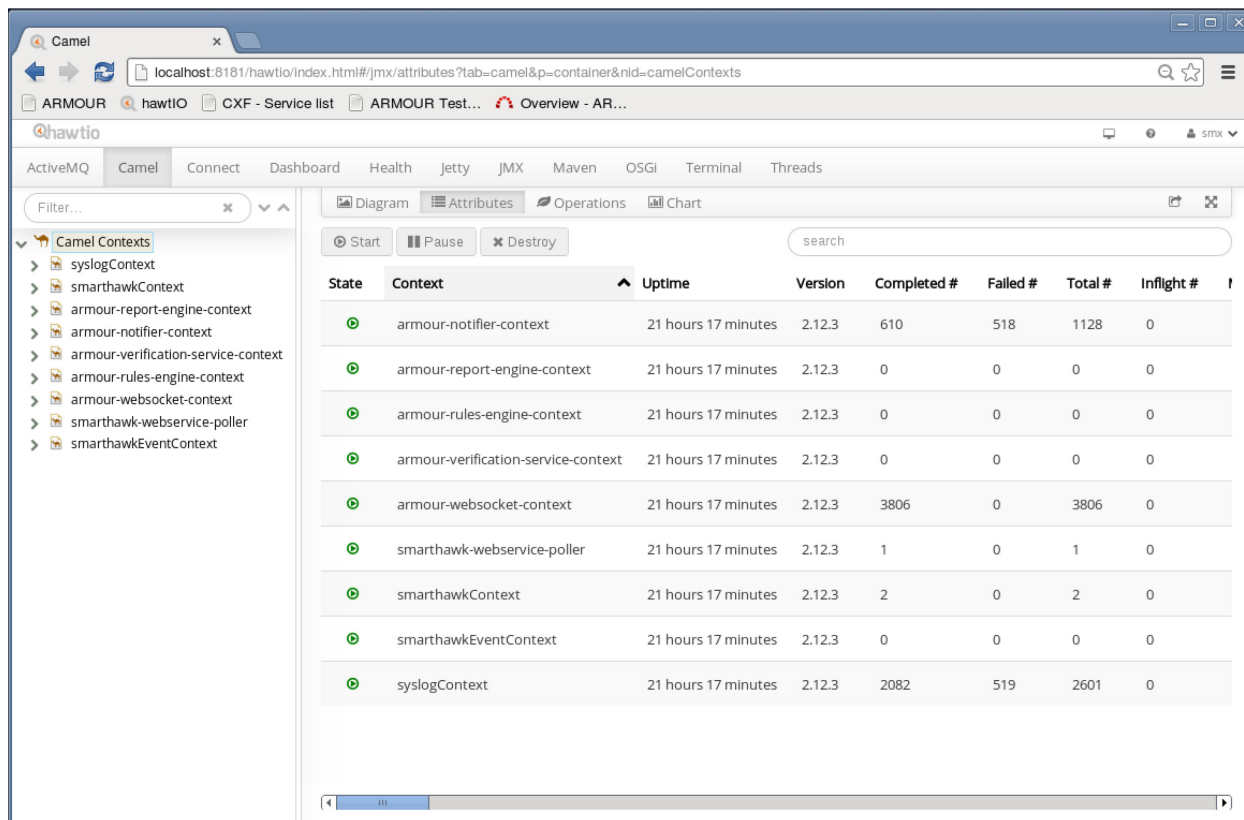


FIGURE 23. ARMOUR IF Administration View – HawtIO

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.4.3.4.7.4 Report Views

ARMOUR provides the ability to generate and run reports based on user-defined templates. A List of templates is presented to the user through the Report Template List widget. From this widget, the user can run a report, modify an existing template, create a new template or delete an existing template. Figure 24 depicts the user selecting a report template from the template list and invoking the template editor window.

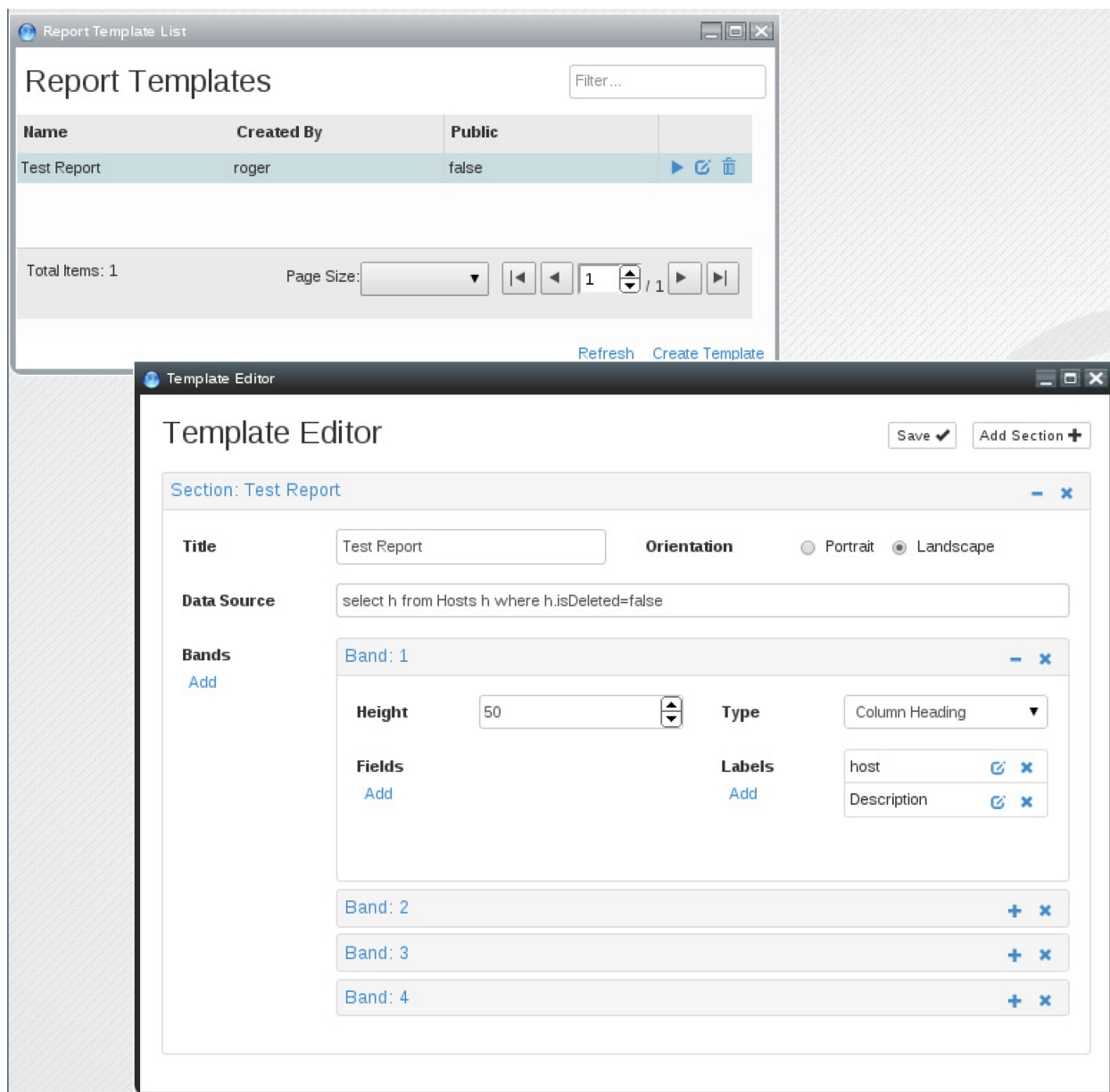


FIGURE 24. Report Template Edit Example

Use or disclosure of this data is subject to the restriction on the title page of this document.

Previously run reports can be retrieved and viewed through the Previously Run Report widget. Here the user can view reports that have already been generated and view the report data. The Previously Run Report widget is depicted in Figure 25.

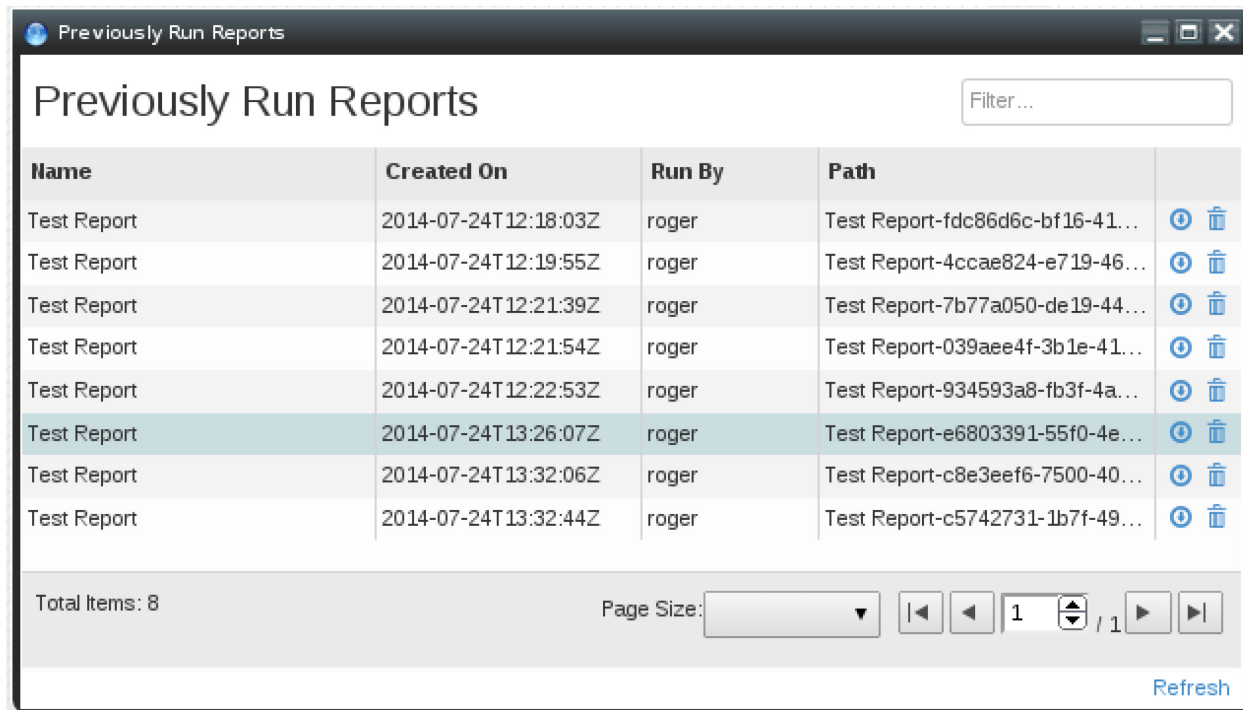


FIGURE 25. Previously Run Report Widget.

3.4.4 Computational Services

This subsection defines the approach to each of the technology modules identified in the STS. Some of these modules have been combined to reduce complexity and overlaps in functionality.

3.4.4.1 Data Normalization

Data Normalization is the process of converting input data from the format of a particular data source to the ARMOUR Data Model format. Conversely, Data Normalization is also performed for data egressing the ARMOUR system where ARMOUR data in the ARMOUR Data Model format is converted to the Data Model of the target technology. The conversion is performed for each Data Source by the Transformation service. The Transformation service uses the Data Source API and knowledge of the ARMOUR Data Model to convert the incoming data type to an ARMOUR Data Model object.

Normalization details for each data source are documented in the Detailed Design Document (GD Canada document No. 741349)

3.4.4.2 Cross Source Correlation

The purpose of the Cross Source Correlation (CSC) module is to manage the significant amount of information that is gathered by the ARMOUR system. In particular, it will combine redundant raw data (possibly gathered from a variety of data sources) into a single normalized representation. This includes producing a single representation of:

- a. Network infrastructure information;
- b. Host identification information;
- c. Host configuration information;
- d. Software vulnerability information;
- e. Security safeguard information;
- f. Operational information;
- g. Network/Infrastructure events information; and
- h. Security events information.

The CSC module has been identified in the STS (ARMOUR TDP Contract W7714-115274-SV Annex B) as an area requiring further investigation. In the STS, four challenges have been identified when performing this correlation, in particular when attempting to arrive at a single representation of the network infrastructure:

- a. Lack of a unique identifier for host-level assets;
- b. Impact of Network Address Translation;
- c. Impact of Dynamic Host Configuration Protocol; and
- d. Unavailability of Media Access Control-level address.

These challenges are being addressed as part of the ARMOUR Research and Development (R&D) efforts, which are discussed in the ARMOUR Algorithm R&D Report (GD Canada document No. 741925). The CSC work is documented in Appendix A.

In order to perform this task, data fusion techniques will be used to resolve conflicts and address missing information. Determining what data is best to use when resolving conflicts is often non-trivial, depending on a number of factors including trustworthiness of the source and age of the data. As such, this has been identified as an additional area for research, which will also be documented in Appendix A of the ARMOUR R&D Report.

The CSC module will interface with ARMOUR IF in order to retrieve information from the Data Storage module. Both normalized raw data and existing data will be extracted from the database, evaluated within the CSC module, fused, and returned to the Data Storage module (via ARMOUR IF) for storage. Note that in order to facilitate performance requirements, the CSC module may communicate directly with the database. The need for this will be assessed as the ARMOUR implementation evolves and is tested with larger inputs.

Use or disclosure of this data is subject to the restriction on the title page of this document.

3.4.4.3 Common Infrastructure Abstraction

Modern enterprise networks consist of a complex and diverse array of devices, software and configuration. The processing, generation and visualization of attack graphs for such networks is challenging due to the size of these networks. Common Infrastructure Abstraction (CIA) seeks to address the above challenges via aggregation by grouping sets of devices with similar attributes and reachability.

Grouping of devices into CIA groups has three key benefits. First, it allows attack graph processing to scale to large networks. Existing research showed that without CIA in some form, attack graph algorithms are not viable for larger enterprise networks due to the long processing time required. Early research into attack graphs focused on enumerating end-hosts individually and discovered performance and scalability challenges even for small networks. Second, for large enterprise networks, attack graph visualization becomes more complex and difficult to interact with for a human user due to the sheer number of devices. Use of network abstraction ensures that attack graphs and associated network topology information is easier to understand by the human mind. Third, modern enterprise networks may have end systems with a number of similar vulnerabilities. Without abstraction, multiple attack paths may be generated for these systems. This may distort the security posture of the network in question. Use of abstraction for systems with similar vulnerabilities will present a more realistic risk assessment of the network.

CIA has been identified in the STS (ARMOUR TDP Contract W7714-115274-SV Annex B) as an area requiring further investigation, where two challenges have been identified when addressing common infrastructure abstraction:

- a. Identification of hosts with similar software configuration; and
- b. Determination of root cause.

Aspects of these challenges are discussed briefly in this section. Detailed results from research in this area will be presented in Appendix C of the ARMOUR Algorithm R&D Report (GD Canada document No. 741925). These results will also be used to guide the detailed design of this module.

Criteria need to be defined to identify similar hosts for automated grouping and abstraction. One approach to infrastructure abstraction utilizes the following steps:

- a. Group hosts based on common Reachability (e.g. only group hosts in a subnet)
- b. Group hosts based on similar vulnerabilities
- c. Divide the reachability group further based on host configuration.

When enabled, the system will support two methods for automated CIA grouping:

- a. Grouping based on routing prefix (per subnet) reachability
- b. Grouping based on reachability and host application configuration profile

Grouping based on host application configuration profile assumes that there is strong commonality in installed software on many hosts in a subnet. This may be a viable assumption for larger enterprise environments as the same set of base applications are often rolled out across all hosts and only a few select users would be allowed to install additional custom software.

Use or disclosure of this data is subject to the restriction on the title page of this document.

To avoid hard-coding criteria for host application grouping an administrator-configured baseline software set will be required. A default configuration will be available for administrators who wish to benefit from this grouping but do not wish to manage the software set.

An example software set could include:

- The same operating system with a specified version and patch level;
- Uniform base applications such as: Web Browser, Java Runtime Environment, Networking driver support; and
- A set of additional application software which may vary in configuration and version number, e.g., Adobe, Office, Skype, Flash, etc.

In the example (a simplified scenario presented for illustration) shown in Figure 26, Host A includes Office 2007 while Host B includes Office 2010. Only Host D includes Skype. Thus, if grouping is based on identical software and versions then none of these hosts can be aggregated to a single abstracted node. A relaxation criterion will be required to facilitate grouping of similar hosts with very similar application configuration profiles. To ensure that this criterion is not hard-coded, it will be based on a white list of approved applications and tools. If an end system contains any of the applications and tools in the white list, then they can be aggregated as part of the group represented by the single abstracted node. This will allow improved aggregation, leading to better abstraction. For example, if the white list in Figure 26 contains both versions of Office as well as Skype, then Hosts A, B and D can be aggregated and represented as one abstracted node. However, since Host C has Flash software and the white list does not include this application, it cannot be aggregated with other hosts under the single abstracted node.

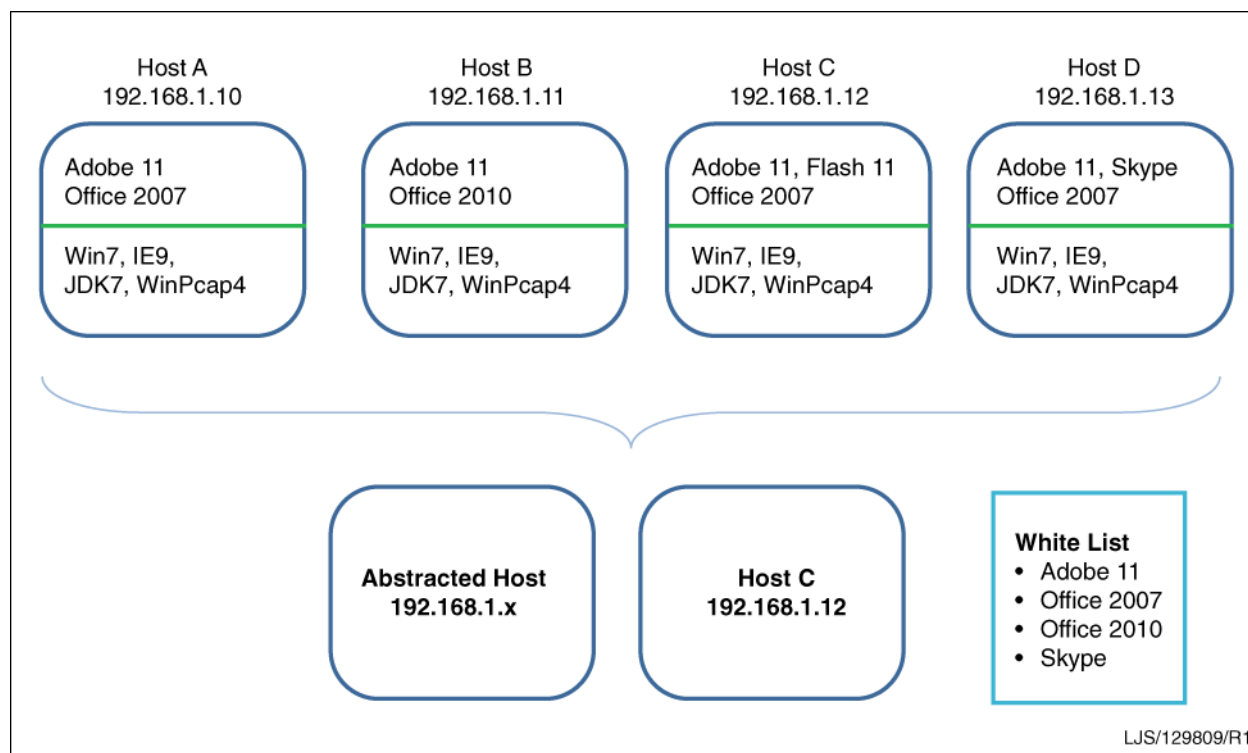


FIGURE 26. Common Infrastructure Abstraction Example

Use or disclosure of this data is subject to the restriction on the title page of this document.

The White list will be configurable by the IT Administrator. Specific software packages should be added to the list only after understanding the security implication of adding that particular package. In addition, days or months after its addition to the white list, it may become apparent that a particular software package has a security vulnerability which necessitates its removal from that white list.

Apart from the above, three additional elements deserve further mention:

- a. A new naming criterion is required in order to represent the abstracted node – this criterion must be cohesive with the addressing scheme in order to ensure consistency from a reachability perspective. The mapping of individual nodes to the abstracted node will be maintained and stored.
- b. While analyzing a security incident, an abstracted end system may be identified as the root cause. A second level of analysis is required to pin-point a particular end system from within the group of devices in the abstracted end system. This is needed since all “similar end systems” are not identical. However, the secondary process can use a more systematic brute force approach to scan through each end systems in the abstract set to identify the particular host of interest. In general for the purposes of this project, the number of hosts mapped to an abstracted host will be less than 1000 – a constraint based on the assumption that multiple subnets will not be grouped into 1 device. Alternatively, another approach is to rerun the attack graph generator on the set of the identified abstracted node but with CIA disabled.
- c. A desirable feature is to allow an administrator the option of enabling or disabling the CIA functionality as not all attack graph generation modules may work with CIA enabled.

3.4.4.4 Data Analysis and Action

3.4.4.4.1 Reachability Analyzer

The Reachability Analyzer amalgamates network topology information with access control (firewall rules) information to generate a reachability graph. This reachability information is then made available to the Common Infrastructure Abstraction (CIA) module for further refinement and to the Attack Graph Generator (either reactive or proactive) as input for the computations.

Determining reachability information requires the following:

- a. Discover network map including connectivity information which covers the core network as well as Virtual Private Network (VPN) tunnels;
- b. Obtain and analyze firewall rules which restrict traffic; and
- c. Determine reachability paths through the network, taking into account the existence of Network Address Translation (NAT) and Port and Address Translation (PAT).

In the ARMOUR STS (ARMOUR TDP Contract W7714-115274-SV Annex B), the Reachability Analyzer has been identified as a research area. The over-arching challenge is to select an approach for generating a reachability graph suitable for use by upstream modules in ARMOUR.

Use or disclosure of this data is subject to the restriction on the title page of this document.

This challenge is compounded by additional challenges related to data that the Reachability Analyzer is dependent on. There are several challenges identified in the STS that will be addressed through the R&D activities:

- a. Connectivity information is distributed across assets;
- b. Connectivity information correlation is dependent on unique identification of hosts;
and
- c. Connectivity information is dynamically changing due to DHCP.

The details as to how these challenges are addressed is captured in Appendix B of the ARMOUR Algorithm R&D report (GD Canada document No. 741925). The remainder of this section presents high-level design of this module.

The Reachability Analyzer provides visibility and intelligence of network topology, device configurations and access policy compliance. It interfaces with the Data Storage subsystem to collect configuration information from all network devices and store results, the Data Normalization and Correlation subsystem to obtain correlated/abstracted data and the Data Presentation subsystem to display the network topology (including drill-down and search capabilities).

The network mapping solution does not directly perform any scanning to determine network connectivity. Instead, it gathers data from a wide array of sources including network layer devices, infrastructure management tools, enterprise management tools, and, of course, network scanners. It uses this information to create a single coherent, virtual model of the network while avoiding many of the pitfalls that scanning alone can lead to. The breadth of this information also allows the network model to contain critical details that are often not considered at this level. Information contained within the model includes:

- a. Routing rules;
- b. Access policies;
- c. NAT policies;
- d. Available services;
- e. Device configurations;
- f. Safeguard configurations; and
- g. Host configurations.

The end result of the collection and analysis of all of these data sources is a single reachability graph (see Figure 27). This graph contains all of the information necessary to describe the connectivity between network resources in terms of both physical and routable connectivity, but also for individual protocols and ports. The operator can set up automated tasks on a periodic basis to ensure the network model is updated dynamically as the network changes.

Use or disclosure of this data is subject to the restriction on the title page of this document.

In general, the entire reachability analyzer will operate through automated retrieval of network data. However, a manual over-ride capability will exist to enable the user to add or delete or edit the final reachability information as desired. This is required as in certain cases the discovered information may be incomplete or incorrect due to “blockages” like domain separations or addressing controls.

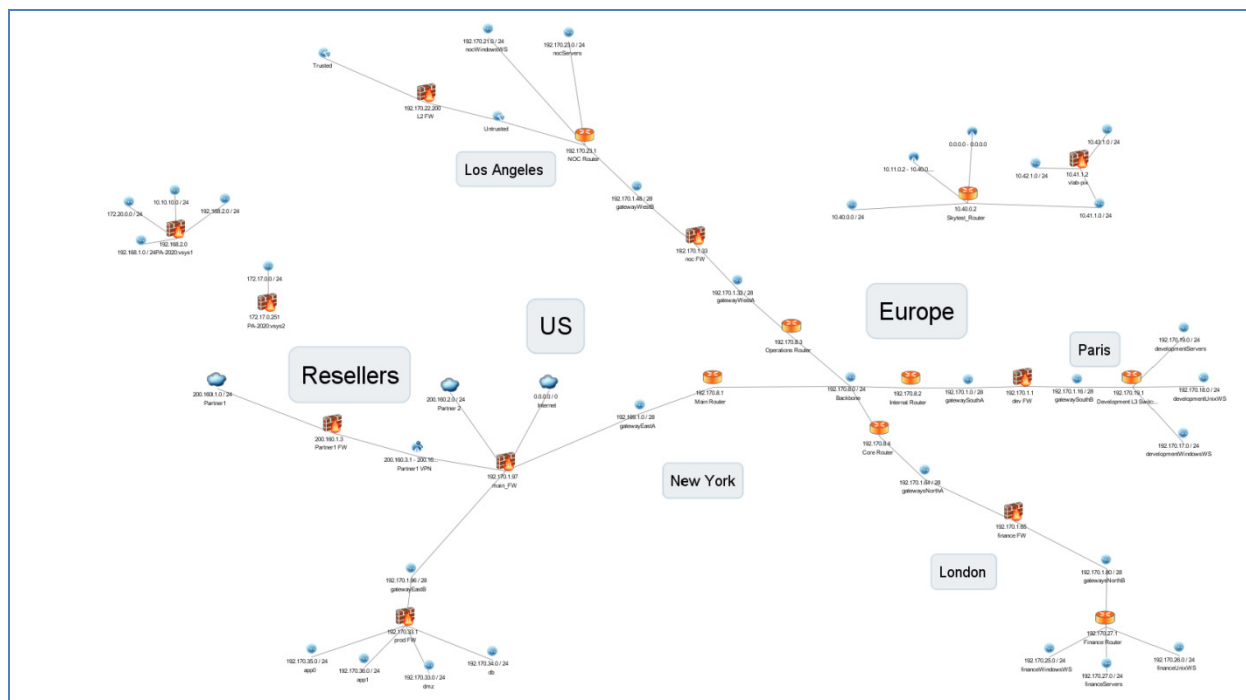


FIGURE 27. Reachability Graph Example

3.4.4.4.2 Operations and Infrastructure Analyzer

The OIA is primarily a broker between data connectors, analysis modules and services and develops priority information from security metrics, operations and resource dependencies. It identifies risks within the network and allows this to be displayed in varying levels of granularity to the user. It provides information to other ARMOUR services to allow automated remedial responses to be initiated and executed based on specified risk criterion.

The OIA supports the following data manipulation functionality:

- The gathering of pertinent data;
- The aggregation of pertinent data;
- The prioritization and reduction of pertinent data to produce an appropriate level of detail for the user;
- The ability to analyze existing operational scenarios;
- The ability to capture historical data for forensic analysis;
- The ability to model scenarios;

Use or disclosure of this data is subject to the restriction on the title page of this document.

- g. The ability to provide suitable data for automated adaptation and response using rules and inference;
- h. The ability to map dependencies between all pertinent resources in the network; and
- i. The ability to identify redundancies and single points of failure in the network.

The OIA has dependencies on attacker models and the security posture metric which must be aggregated and manipulated as part of the calculation of the operation priority metric.

The OIA operates in different layers or stack of functionality. The TBD diagram captures the high-level architecture of information through the OIA functionality stack and is composed three functional blocks:

1. The OIA maps the dependencies of all services and applications in the network exposing points of failure and producing a network topology.
2. When dependency information is coupled with security metrics this functionality now provides an attack/vulnerability based situational awareness.
3. When the previous aggregated data is combined with OIA operations information provides the capability to produce and prioritize system responses whether executed manually by a system administrator or through an automated rule based/heuristic mechanism.

Figure 28 shows the information stack and flow of information through the OIA. The aggregated data produced at the top of the OIA stack provides in-feed to both proactive and reactive capabilities. The response from these processes implements changes the configuration items at the lowest layer of the OIA stack which effectively produces a feedback mechanism where the user can visually inspect changes into the system. The data associated with each layer of the OIA stack can be current, historical or simulation based producing a very flexible analysis tool and this is show entering the stack from the left – this data is captured in a data store.

The OIA Network Resource Dependency Module (ONRDM) provides the operation Dependence Metric (DPM) which feeds into the ARMOUR attack/vulnerability functionality. This dependency metric could be calculated in advance and stored in the datastore or could be calculated at runtime based on filter criteria.

The attack/vulnerability situational awareness capability is provided by the Attack/Vulnerability Situational Analysis Module (AVSAM) that produces the SPM. This metric is flowed into the OIA Operations Information Module (OOIM) which aggregates the SPM and the Operational Priority Values (OPV) contributed by the operations information to produce a detailed, prioritized view of the topology with vulnerability data.

Use or disclosure of this data is subject to the restriction on the title page of this document.

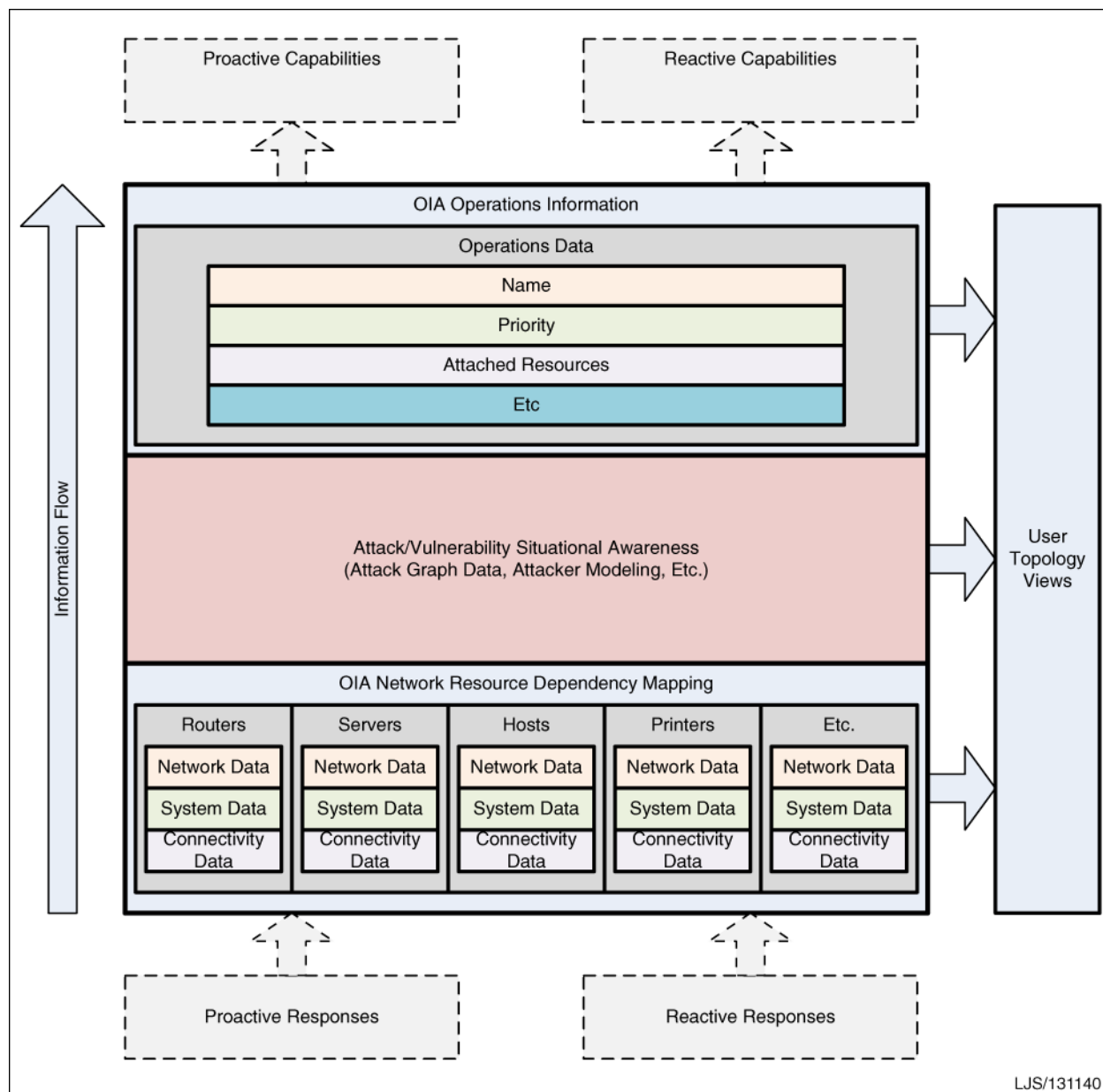


FIGURE 28. OIA Information Stack and Information Flow

Each module within the OIA contains a significant amount of functionality, data management and manipulation and provides different views into the data as processed within that module. The exact nature of the processing within each module is not yet known and will be expanded upon as the research and development activities evolve.

Use or disclosure of this data is subject to the restriction on the title page of this document.

a. OIA Network Resource Dependency Mapping Module (ONRDMM)

The ONRDM module requires a significant level of functional detail in the information that is captured to map all the potential dependencies pertinent to network hosts. This module produces the host dependency metric. Due to the detailed level of information required host based resource scanners are being considered.

A DAR will be conducted to determine the suitability of the available tool(s). The following criteria have been identified to support this activity:

- Open-source (preferably with GPLv2, MIT, BSD license),
- Support for *nix and MS Windows operating systems,
- Host-based (active) scanning support,
- Server-based (passive) scanning support,
- Fast in data acquisition,
- Flexible in which data is acquired,
- Schedulable based on time, priority and on-demand scanning,
- Data format suitable for injection into a data store,
- Capable of scanning registry/machine configuration, security settings and patch information,
- Capable of scanning all network parameters including IP, MAC, ports/protocols, services, Etc.
- Capable of identifying all applications, revisions including operating system,
- Capable of management information base (MIB) scan.

The connectivity data of the ONRDM will be provided by the reachability analyzer.

b. Attack/Vulnerability Situational Analysis Module

The AVSAM provides the capability to analyze attacker models and attack graphs to produce an attack/vulnerability situational analysis capability. This module will produce the security posture metrics required by the OOIM. The AVSAM functionality is currently in the R&D phase; this R&D is captured in Annex G - Proactive CND of the ARMOUR R&D Report, GD Canada document number 741925. The AVSAM architecture and design will be elaborated and evolve as R&D continues.

c. OIA Operations Information Module (OOIM)

The OIA operations information module captures the operations information and the network resources associated with the operations. The OOIM provides a view of the priority of the resources and when combined with the host dependency metric and security posture metric produces a comprehensive view of which resources are the highest risk of exploitation, single points of failure, data choke points, etc.

Use or disclosure of this data is subject to the restriction on the title page of this document.

The operations and infrastructure analyzer is a research area and is currently in the active research phase. OIA functionality is dependent on the proactive computer network defence R&D and the reachability analyzer R&D. The modular design of the OIA has a number of advantages:

1. Minimizes the impact of the coupling between the dependencies in the different research areas. This allows functionality stubs to be created while a module is being developed.
2. Allows modules to be upgraded and modified with minimal impact to the module in the upstream.
3. Allows module functionality to be tested in isolation.

Allows data views to be created and layered on the specific module functionality or aggregated to provide super views.

3.4.4.4.3 Proactive and Reactive Attack Graph Generator

The Attack Graph Generator module is part of the Risk Treatment context discussed in subsection 3.1.3 previously. It is the first phase in the process, and comprises the analysis of the information collected about the network infrastructure, host configurations, software and services, and vulnerabilities, and the generation of an attack graph that models the possible attack paths between an identified “attacker location” and the network resources based on the analysis.

The internal network model on which attacks are assessed will include the most up-to-date information on network resources and operational data including:

- a. Available network interfaces;
- b. Routing rules;
- c. Reachability/Connectivity information;
- d. Network Address Translation (NAT) policies;
- e. Access policies and applied regulations;
- f. Available Services;
- g. Operation Impact definitions;
- h. Host priority information; and
- i. Known Vulnerabilities.

This module will extract the required data from the ARMOUR database via ARMOUR IF. Computation of the attack graph will be done internally to the component, and the resultant graph will be stored in the database again via ARMOUR IF, made available for other modules to process, including the Attack Graph Analyzer module for attack asset identification and the OIA module for security posture evaluation.

Use or disclosure of this data is subject to the restriction on the title page of this document.

Since the module is not involved in data gathering, normalization, or correlation, whether it is being used in a proactive or reactive flow, or with real, simulated, or historic data, is immaterial. As a result, the Attack Graph Generator can be used to help assess the current security posture, simulate the security posture given application of changes to the network, and as part of the reactive workflow given detection of an incident.

An example as to why running the Attack Graph Generator with simulated data is desirable, consider the evaluation of COAs. In order to assess a COA, the impact to the security posture must be known. To know this, the Security Posture Metric must be computed by the OIA module. This requires an attack graph that reflects the network with the COA applied. This must be simulated. Further discussion into the Course of Action Analyzer module is provided in subsection 3.4.4.4.6 and will be updated in future revisions of this document (primary development phase is phase 4).

The open source research tool MulVAL has been identified as the likely piece of software that will provide the functionality required for the Attack Graph Generator module. From initial analysis, it seems to satisfy the Attack Graph Generator requirements presented in the SRS (GD Canada document No. 740930). Further, it is a tool that both GD Canada and DRDC have worked with previously, and are thus familiar with how it operates. That said, during the initial stages of Phase 3, a survey of other possible tools, such as Cauldron, will be performed, and if deemed necessary, a Decision Analysis Resolution (DAR) exercise will be completed in order to ensure the appropriate tool is being chosen. The primary criteria for such a DAR will include whether or not the product is open source, whether or not the product can produce AND/OR directed attack graphs (or directed forward hypergraphs), and an assessment as to how well it would integrate with the ARMOUR system.

3.4.4.4.4 Proactive and Reactive Attack Graph Analyzer

The Attack Graph Analyzer module is the second phase of the Risk Treatment process. It uses the attack graph generated by the Attack Graph module and outputs a list of attack assets (referencing the attack graph), along with a metric indicating the value of the asset to an attacker attempting to disrupt the system. Required for this computation is an attack graph, and metrics information including likelihood of success of technical attacks, likelihood of success of non-technical attacks, and the maturity of exploits.

Data required for this module will be obtained from the ARMOUR database via ARMOUR IF. This module will then compute the attack dependence metrics for each asset in the attack graph, and output back to the database, via ARMOUR IF, the list of attack assets and metrics relative to the attack graph used as input.

The Attack Graph Analyzer module provides context information to the end user regarding what aspects of the attack graph are most valuable to an attacker, and thus should likely be prioritized when applying COAs.

It is anticipated that the DRDC-developed tool AssetRank will be used to provide the functionality within the Attack Graph Analyzer module. Based on the initial analysis, it satisfies the majority of the SRS requirements (GD Canada document No. 740930). As with the Attack

Use or disclosure of this data is subject to the restriction on the title page of this document.

Graph Generator, an investigation into other tools that may provide this functionality will occur, and a DAR will be performed if necessary. It is anticipated that there are not many, if any, tools that would provide a metric as described in the requirements, and meeting said requirements will be one of the primary criteria for evaluation. Others will include the ability to work with AND/OR directed attack graphs or directed forward hypergraphs, and an assessment as to how easy it is expected to be to integrate with ARMOUR.

3.4.4.4.5 Incident Analyzer

Incident analysis leverages the power of complex event processing to provide a flexible and agile solution for accurately resolving network security events into identifiable security incidents. The ARMOUR system uses a cyber operations centre solution framework to integrate and correlate cyber events from multiple resources to support incident analysis and response. As discussed previously, security event and log data from DREnet security data sources, including IDS, firewalls, routers, Intrusion Prevention System (IPS), SIEMs, etc., is processed using a universal log management solution so that all of the event and log information is normalized into a standard format such as CEF. This data is then forwarded to the incident analysis module and built-in rules engine for processing.

The rules engine validates the legitimacy of events through dictionary mapping and advanced analytics and passes valid events on for Attack Path and COA Analytics. It provides complex event processing of streaming events, providing the visualization layer a near-real-time view of high-priority events, pattern recognition, and temporal events. It is exposed to the Enterprise Service Bus (ESB) as a Java component or as a service (via secure socket connection, or encrypted over Java Messaging System (JMS)) and integrated as a message flow. The rules engine will leverage the abstracted and reduced event data provided by the cross source correlator. Details of this will be defined in the Detailed Design Document (GD Canada document No. 741349).

The rules engine provides much finer grained facilities for processing security data via a powerful, user-customizable rules syntax. Its raw rules engine provides the capability to compare host or network identifiers to watch lists, while its analytic rules engine can track event statistics as well as provide robust state-based detection and categorization.

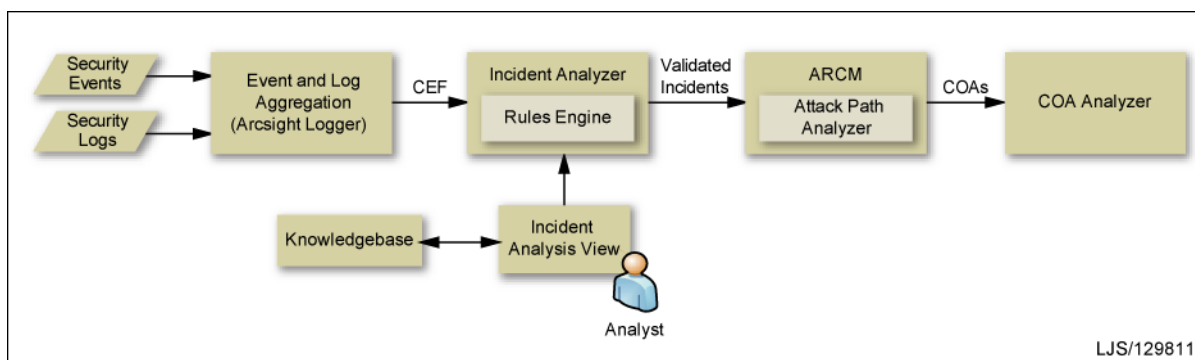


FIGURE 29. Incident Analysis Flow

Use or disclosure of this data is subject to the restriction on the title page of this document.

Correctly determining the legitimacy of events is a difficult process and requires an approach that can combine multiple methodologies. The rules engine combines dictionary mapping of events to incidents, along with more advanced analytic processing, including:

- a. Stateful analysis where each incident definition may require that a series of events takes place in a defined order, within a limited amount of time;
- b. Historical analysis where events are compared against a baseline or sliding window of events; and
- c. Signature analysis where each incident defines a set of events and a threshold and is triggered once the number of captured events in the set reaches the threshold.

By comparing incoming events to each other, as well as pre-established definitions, the incident analyzer greatly reduces the incidence of both false positive and false negative results. Note that addressing high false positive rates and high false negative rates has been identified as an area requiring further investigation in the ARMOUR STS (ARMOUR TDP Contract W7714-115274-SV Annex B) Appendix E of the ARMOUR R&D report (GD Canada document No. 741925) will contain the research results that explain how these challenges could, and will, be addressed in ARMOUR. Once incidents have been validated, the likely compromised hosts are updated in the database and passed to Risk Treatment for inclusion in analysis. There they can be correlated with the abstracted infrastructure model to identify the critical resource information associated with the incident and impacts to operations. Risk Treatment is responsible for computing the impact of incidents to the security posture of the system and recommending COAs.

The incident analysis capability also includes an integrated, open-source semantic knowledgebase (Wiki – see Data Storage subsection) to allow analysts to develop a living CND body of knowledge with references to previous incident data, standard operating procedures, incident response procedures, applicable policies, threat and intelligence data and COA reference information.

3.4.4.4.6 Course of Action Analyzer

The ARMOUR System uses a multi-module approach to meet the challenge of generating and analyzing COAs for proactive and reactive defence. This design provides the most flexibility in the generation of applicable COAs as well as their evaluation and selection for execution.

In this design, the responsibility for generating COA sets may be distributed among more than one tool. The benefit of this design choice is that it allows the ARMOUR System to play to the relative strength of each tool, as well as ensure future relevancy. A design that relies on only one data source will likely produce lower quality results than a source capable of interfacing directly with patch management repositories.

These tools will be responsible for evaluating the issue to be resolved and the architecture of the network to generate a set of COA recommendations that maximally increases the security of the network as a whole. This requires an evaluation of the security posture of the network assuming the COA was applied. That is, the OIA module will be called upon to compute the Security Posture Metric in order to ensure the recommended COAs maximally increase the security of the

Use or disclosure of this data is subject to the restriction on the title page of this document.

system. This may require computing a new attack graph, which means the Attack Graph Generator module must be triggered, with simulated data provided as input.

One tool that can generate COA sets is COADS, as described in the context of Risk Treatment in subsection 3.1.3. In the current implementation of COADS, the evaluation is based on the amount of “rank” removed from the graph (based on the output from AssetRank, which is being considered for the Attack Graph Analyzer module). This functionality has been modified previously to enable COADS to use other methods of evaluation, and as a result, it is expected to be relatively easy to modify COADS to use other metrics to evaluate COAs.

Specific determining factors when evaluating COAs will include, but will not be limited to:

- a. Vulnerability risk metrics;
- b. Attack risk metrics;
- c. Exposure metrics;
- d. Operational implications;
- e. Network architecture;
- f. Assigned cost;
- g. Availability of approved patches; and
- h. Availability of known software mitigation techniques.

This data will be obtained from the ARMOUR database via ARMOUR IF at the same time as the attack graph and attack asset information. While not used when generating COA sets, the attack asset information is valuable to the end user when evaluating the COAs presented for application in the semi-automated response mode.

As part of the Course of Action Analyzer module, Course of Action Selection has been identified in the STS (ARMOUR TDP Contract W7714-115274-SV Annex B) as an aspect of ARMOUR requiring further investigation. Specifically, three challenges related to the assignment of cost metrics, a key component when determining what courses of action should be considered, have been identified:

- a. Initial cost metric values, specifically automatic generation versus manual collection;
- b. Operator adjustment of cost metrics, in particular to allow correction to the automatically generated costs; and
- c. Application of adjusted cost metrics to future events, providing another input into the automatic generation of cost metrics.

Research in this area, and into these challenges, will be performed during Phase 3 of ARMOUR. The results of the investigation will be captured in Appendix F of the ARMOUR Algorithm R&D Report (GD Canada document No. 741925). Once COAs have been successfully generated, the COA Analyzer is responsible for normalizing, correlating, evaluating, and initiating the execution of COA as depicted in Figure 30.

Use or disclosure of this data is subject to the restriction on the title page of this document.

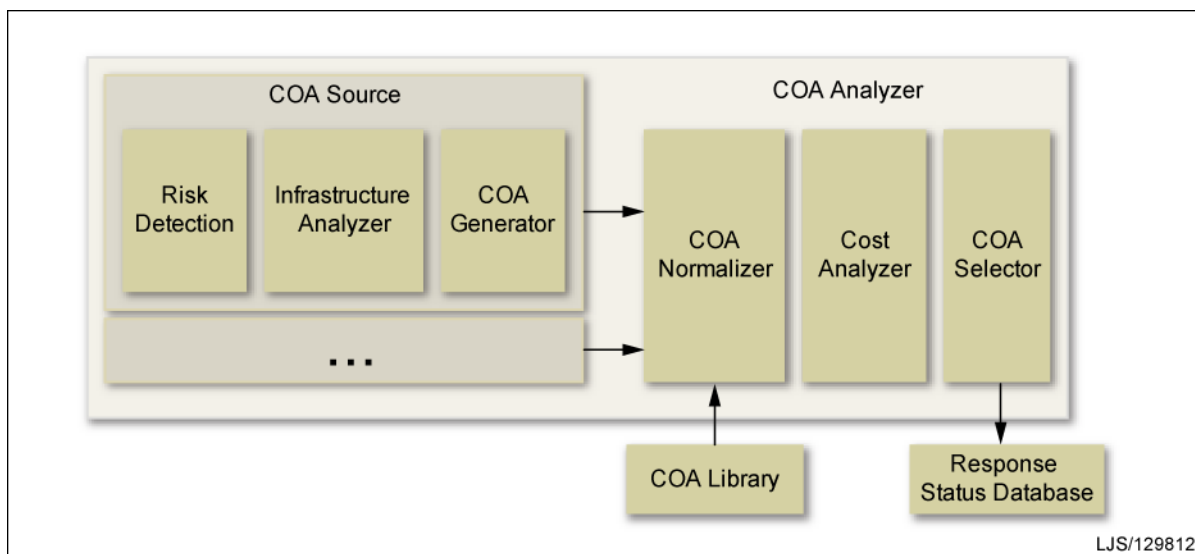


FIGURE 30. COA Analyzer

The core of the COA Analyzer is the approved COA library. This library contains the complete set of all COAs that have been approved for deployment, their component actions, associated costs, risks, associated rollback COAs, and their mappings to the COAs generated by other products. In this case, the component actions describe, in terms of the effector API, the low-level actions that must be carried out to achieve the desired action. This is necessary so that a high-level COA, such as quarantine host, may be translated into a concrete set of actions, such as disable port, disable accounts, and initiate forensic workflow. The COA library is updated continuously to support the management or creation of COAs. In the event that an incoming COA cannot be mapped to a counterpart in the library, it is logged and presented to the operator for classification. This may involve mapping to an existing COA, creation of a new COA, or blacklisting the COA from further consideration.

Once the incoming COAs have been normalized, their cost and risk metadata are analyzed. The total risk for each COA is calculated as a measure of its implementation cost added with the result of the cost associated with its rollback COA scaled against its risk. Each of the COAs is ranked in descending order, relative to its total cost. If the total cost of the top-ranked COA is below a predefined threshold, it is approved for automated response and passed directly to the Effector Connector Framework via its API to be executed. Otherwise the list of COAs must begin the approval process.

The COA approval process requires that the entire list be presented via the COA View. The operator may select one or more COAs from the available list, or reject the list entirely. The operator may also choose this opportunity to manually select a COA that has not been presented. Once COA selections have been made, a response is generated and sent to the Effector Connector Framework (refer to Figure 35) via its API, to be executed.

-
- b. In the normalization phase, each of the COAs in the list is matched against the dictionary provided by the COA library to yield their approved, standardized counterpart. This is the translated mapping:
 - i. Quarantine by switch port (x);
 - ii. Quarantine by firewall (y); and
 - iii. Apply configuration (c).
 - c. The total cost is calculated for each COA in the list and they are ranked by their relative costs. The applied costs and new ranking for this example are as follows:
 - i. (75) Apply configuration (c);
 - ii. (100) Quarantine by firewall (y); and
 - iii. (125) Quarantine by switch port (x).
 - d. The COA selector compares the top-ranked COA against a pre-determined threshold to determine whether the COA can be automatically executed. This is the point at which semi-automated and automated response handling diverge. In this example, the cost is above the pre-determined threshold of 50, so the response is not tagged for automated execution and must be sent to an Analyst for approval.
 - e. The analyst may select zero or more COAs from the list to be executed.
 - f. The selected COAs are logged to the response status database.
 - g. The selected COAs are executed via the Effector API, which provides a list of possible effect commands. Because the response has not been tagged for automated execution, the commands are set for workflow execution.
 - h. The operator monitors the execution of all COAs and their status from the COA View. If there is a problem, the operator will be responsible for cancelling the COA or executing the associated roll-back COA.

Use or disclosure of this data is subject to the restriction on the title page of this document.

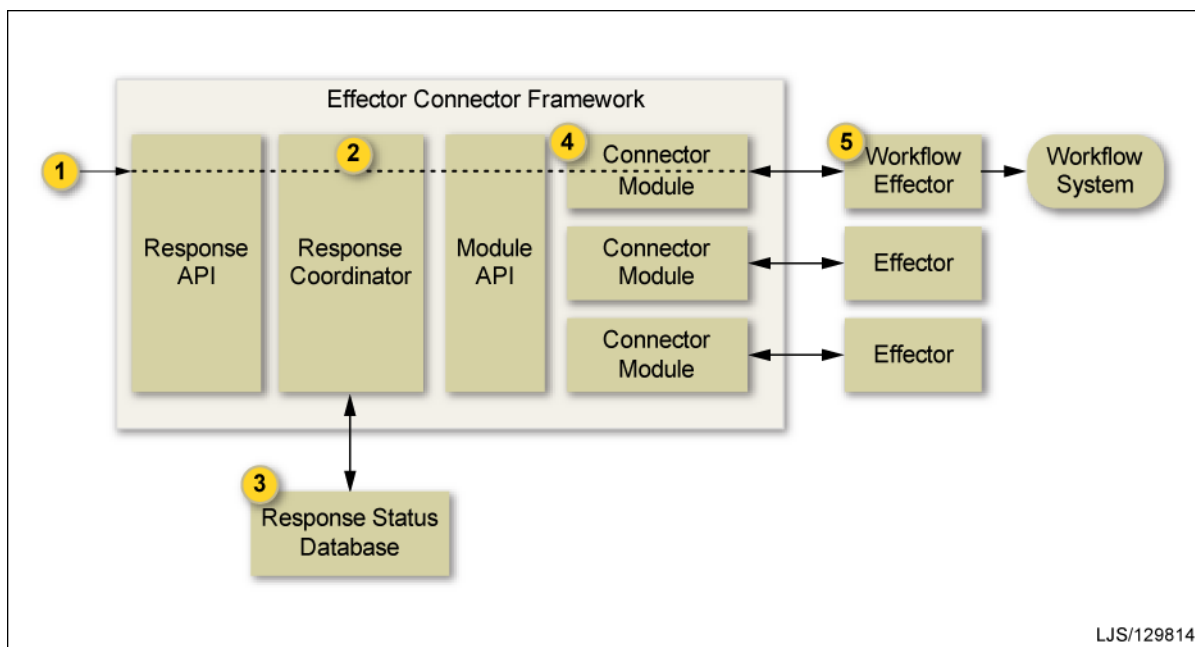


FIGURE 32. Semi-Automated Response Process

- i. One or more commands are received via the effector API.
- j. The Response Coordinator translates the Response API commands into individual actions for the appropriate effector connector modules. It is also responsible for scheduling these commands and maintaining the current status of each. In the event of a failure or cancellation, it is responsible for safely ending the execution of the commands.
- k. The individual module commands are logged and their status is tracked.
- l. Each connector module engages their specific connector to execute the required command and translates any status returned.
- m. Each effector performs the required action. In this case the selected commands are queued within the workflow system.

The Data Model for the ARMOUR system includes support for maintaining historic data for up to one year. Historic data is maintained by the data storage subsystem and includes information on the COA transaction that was implemented as part of a change (response status database). Within the COA library, a catalog of transactions are maintained and called upon when a COA is identified. Additionally, each transaction within the library contains a counter transaction intended to reverse the outcome of the transaction. When a transaction is effected, information about the transaction is stored in the database for historical purposes.

When an operator decides that an effected change is no longer desired, a COA rollback is executed. In this situation, the historic data regarding the particular COA (stored within the database) is accessed to identify the initial transaction and presented within the COA view to the operator. The COA library is used to lookup the counter action for the transaction. The counter action is selected for execution and the action is logged to the response status database.

Use or disclosure of this data is subject to the restriction on the title page of this document.

Situations may arise when an effected COA cannot be rolled back due to dependencies instantiated as part of the initial COA implementation (i.e. a patch that cannot be reverted). In this case, the operator will be required to manually revert the COA either through a restore operation or other means.

Similarly to a COA, the counter action is executed via the Effector API and the operator can monitor the status from the COA view.

3.4.4.4.8 Automated Response

Automated response is the process by which COAs are generated in response to an event and executed directly without human interaction. This process shares the same basic two stage approach as semi-automated response, but the control flow and results differ significantly. Similar to above, a notional scenario and the flow of events that would occur to support automated response are presented below (Figure 33 and Figure 34).

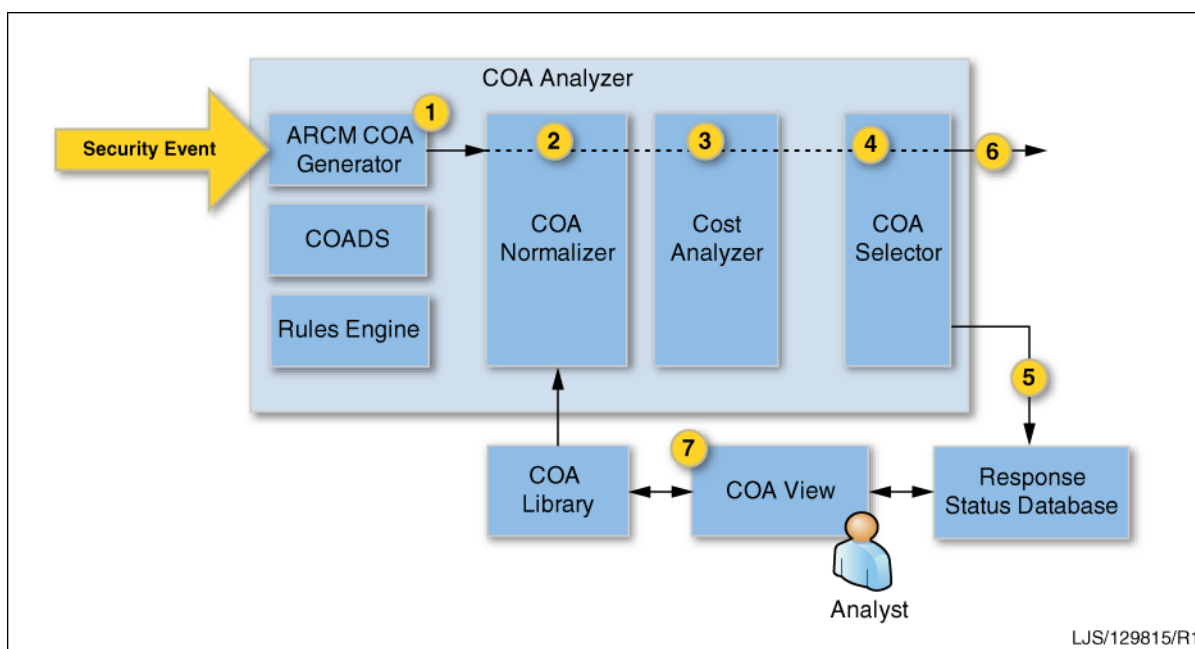


FIGURE 33. Analysis Process for Automated Response

- a. The response process begins when a list of possible COAs is generated by any of the approved COA sources. In this case, the Risk Treatment solution has acted in response to a simulated attack on a web server. The simulation has shown that an attacker could exploit a known and exposed vulnerability on the web server to gain access to the DMZ. The list of COAs for this example is as follows:
 - i. Remove the web server from the network by disabling switch port x;
 - ii. Create firewall rule y to limit access to the web server; and
 - iii. Apply patch to the web server to remove the vulnerability.

Use or disclosure of this data is subject to the restriction on the title page of this document.

- b. In the normalization phase, each of the COAs in the list is matched against the dictionary provided by the COA library to yield their approved, standardized counterpart. This is the translated mapping:
 - i. Quarantine by switch port (x);
 - ii. Quarantine by firewall (y); and
 - iii. Patch vulnerability (p).
- c. The total cost is calculated for each COA in the list and they are ranked by their relative costs. The applied costs and new ranking for this example are as follows:
 - i. (35) Patch vulnerability (p);
 - ii. (100) Quarantine by firewall (y); and
 - iii. (125) Quarantine by switch port (x).
- d. The COA selector compares the top-ranked COA against a pre-determined threshold to determine whether the COA can be automatically executed. This is the point at which semi-automated and automated response handling diverge. In this example, the cost is below the pre-determined threshold of 50, so the response is tagged for automated execution.
- e. The selected COAs are logged to the response status database.
- f. The selected COAs are executed via the Effector API, which provides a list of possible effect commands.
- g. The operator monitors the execution of all COAs and their status from the COA View. If there is a problem, the operator will be responsible for cancelling the COA or executing the roll-back COA associated with it.

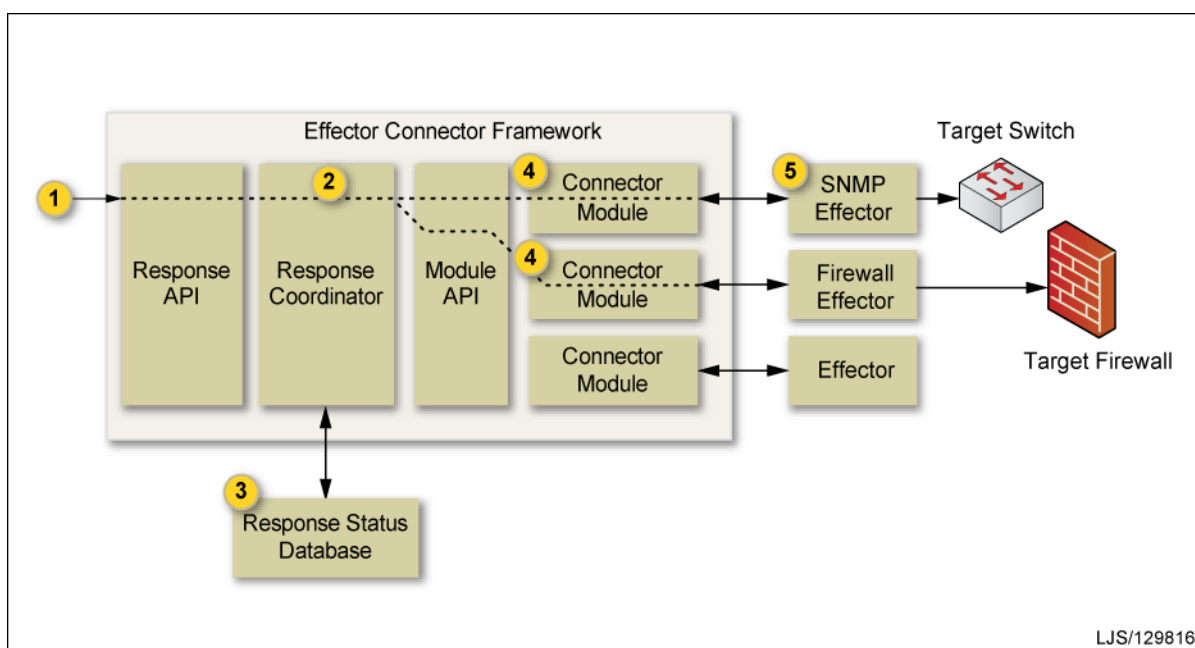


FIGURE 34. Automated Response Process

Use or disclosure of this data is subject to the restriction on the title page of this document.

- h. One or more commands are received via the Effector API.
- i. The Response Coordinator translates the Response API commands into individual commands for the appropriate effector connector modules. It is also responsible for scheduling these commands and maintaining the current status of each. In the event of a failure or cancellation it is responsible for safely ending the execution of the commands.
- j. The individual module commands are logged and their status is tracked.
- k. Each connector module engages their specific connector to execute the required command and translates any status returned.
- l. Each effector performs the required action. In this case the SNMP Effector disables the switch port being used by the effected host, and the Firewall Effector adds a rule to the gateway firewall to disallow any traffic from the effected host.

Rollback of automated COAs is implemented similarly to the semi-automated response.

3.4.5 Effector Connectors

The ARMOUR System uses service orientation best practices and principles to provide abstraction of the underlying effector technologies. The design allows semi-automated and automated response mechanisms as the means to execute COAs. This involves building web services to be deployed within ARMOUR IF that are responsible for orchestrating communication, requesting data, and returning a response in the process of executing a COA.

The approach for integrating with a diverse set of network effectors is to develop an Effector Connector Framework (see Figure 35). This framework will provide a unified API for commanding, controlling, and receiving status updates from the various supported effectors.

To facilitate support for the widest variety of effectors, the Effector Connector Framework API will provide two standardized interfaces. The first defines an internal module interface from which all effectors will be implemented. These include standards based effectors that already exist within the ARMOUR IF, such as SNMP and workflow, as well as proprietary effectors. Separation of effector control implementations into individual modules ensures the framework remains resilient while allowing effector functionality and support to quickly evolve. The external facing response API will be implemented as a secure, ReSTful web service. This is the interface through which the COA Analyzer will be able to initiate the execution of specific COAs. This approach allows complex commands, sometimes involving two or more source technologies, to be encompassed within a single action, regardless of the transport, source API, or interface.

The logic that joins the two interfaces, the Response Coordinator, is responsible for the coordination of execution in the event that an action requires multiple modules to be used in conjunction or for multiple steps be accomplished in order. This requires that the coordinator be able to verify the completion of each step and report its status to the database and the COA View. In the event of uncertainty in the network status, execution failure, or manual cancellation by the operator, the Response Coordinator aborts the current response in a safe and orderly manner.

Use or disclosure of this data is subject to the restriction on the title page of this document.

All interactions with the Effector Connector Framework provide full message encryption and strong authentication and non-repudiation requirements. Additionally, a full response flow log is maintained for every response that is initiated.

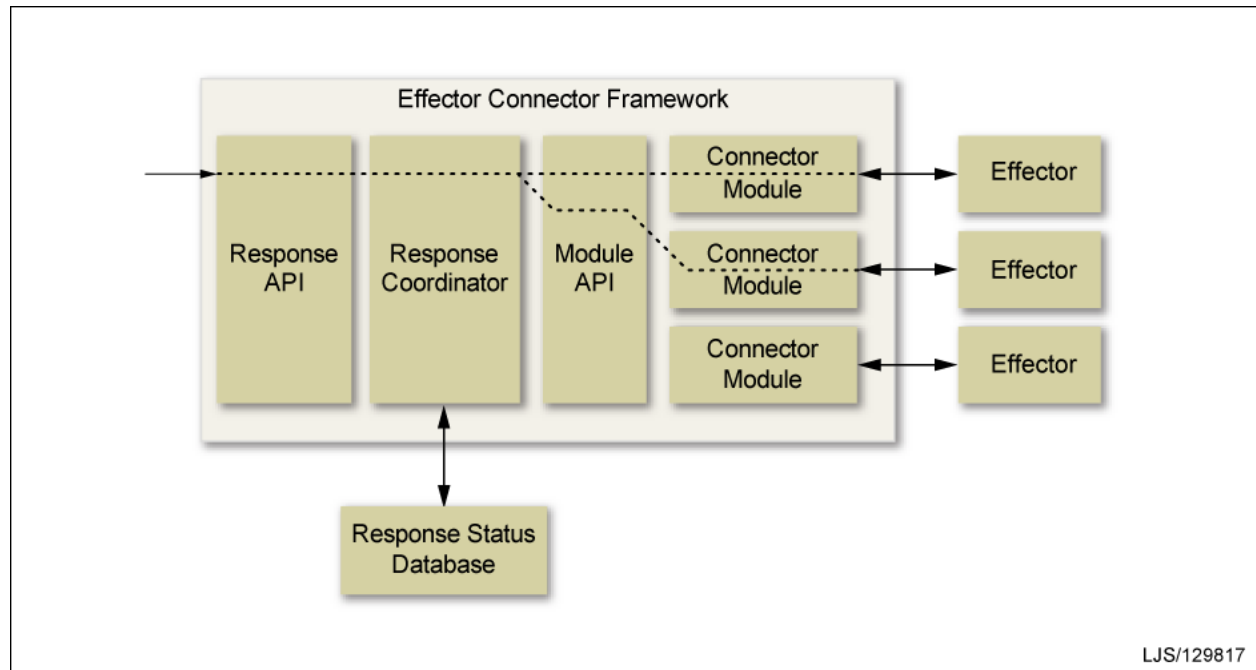


FIGURE 35. Effector Connector Framework

Use or disclosure of this data is subject to the restriction on the title page of this document.

4. ARMOUR SECURITY ARCHITECTURE

For ARMOUR, the functions required to maintain the security posture are realized through the implementation and maintenance of applicable security controls from DND IT Security Control Profile: Designated-Medium-Medium. The selection of controls assumes that the enclave is a standard security management domain with enhanced integrity, data flow enforcement and audit requirements. It consists of the standard desktop environment, virtual images of common network security applications, middleware and operating systems. Use of removable media and mobile devices is prohibited. The ARMOUR IT security control profile supports the following concepts:

- Role-based access controls to information: the access to information within the enclave is controlled. All users are screened to established trustworthiness relative to all data prior to being given an account. Users are authenticated before being permitted to access ARMOUR data to support access controls and security audit. Strict separation of duties is maintained between system administration and user functionality. Security logs are maintained and reviewed to identify behaviour that requires intervention.
- Controlled ARMOUR Enclave Perimeter: the interactions with DRENet and other domains are restricted to traffic relevant to the ARMOUR System. Use of one way communication devices to strictly enforce traffic flow requirements is supported. Controls are applied to anticipate, detect and mitigate threats at the enclave boundaries. Hardware and software are tested to ensure proper configuration.
- Protection of Physical Facilities: access to DRDC and GD Canada facilities used by the ARMOUR enclave is limited to authorized personnel. All personnel are authenticated before being permitted to enter any facility used by ARMOUR.
- Internal Resource Usage Control: inter module communications are restricted. Modules are authenticated to support access to other sub-systems and modules and security audit.

Technical security controls related to internal resource usage control, including the inter-module access control, are implemented and managed via the ARMOUR IF security services operating in a hardened operating environment (as depicted in Figure 36). The main one is directory based authenticated communication between modules. Module authentication means that any module communicating with the ARMOUR IF messaging bus requires a username and password to publish or subscribe to the bus. The authentication is verified against the LDAP server. The LDAP server therefore will contain system user identity, credential and password information. This simple mechanism enables security of information in transit.

The second, and most common authentication, is for users. ARMOUR Data Presentation modules implement user authentication against the User Directory (LDAP server). The LDAP group(s) this user belongs to determines the user interface presented (i.e., Administrator panels/tabs are only available to Admins, etc.), and more importantly in a military scenario, the information itself is verified to ensure that only a particular role can view certain pieces of data. The way this is achieved by mapping LDAP groups directly to roles. Using LDAP groups to determine a user's role membership has the advantage of centralizing the management of data

Use or disclosure of this data is subject to the restriction on the title page of this document.

viewing privileges. Additionally, this decouples System Administrators from being able to view privileged data. Administrators can administer the system without necessarily having the rights to view all data, conversely, certain users can't administer the system, but can view protected data.

For internal services, when communications between modules require security controls (i.e., the modules are stored on different virtual images or on different platforms) the session can register the required security functions (i.e., encryption, integrity checking, content verification) and the appropriate services are called in. Attributes can be defined as to when the security features are invoked based on absolutes (services are on different machines) or on policy. These policies can be defined internally or read from external sources.

Users are managed based on the attributes associated to their expected behaviour in the system. All action related to user attributes is logged in an accountable or trusted fashion. Depending on the desired security posture, identity management can support multi-factor authentication including PKI or other user token based approaches.

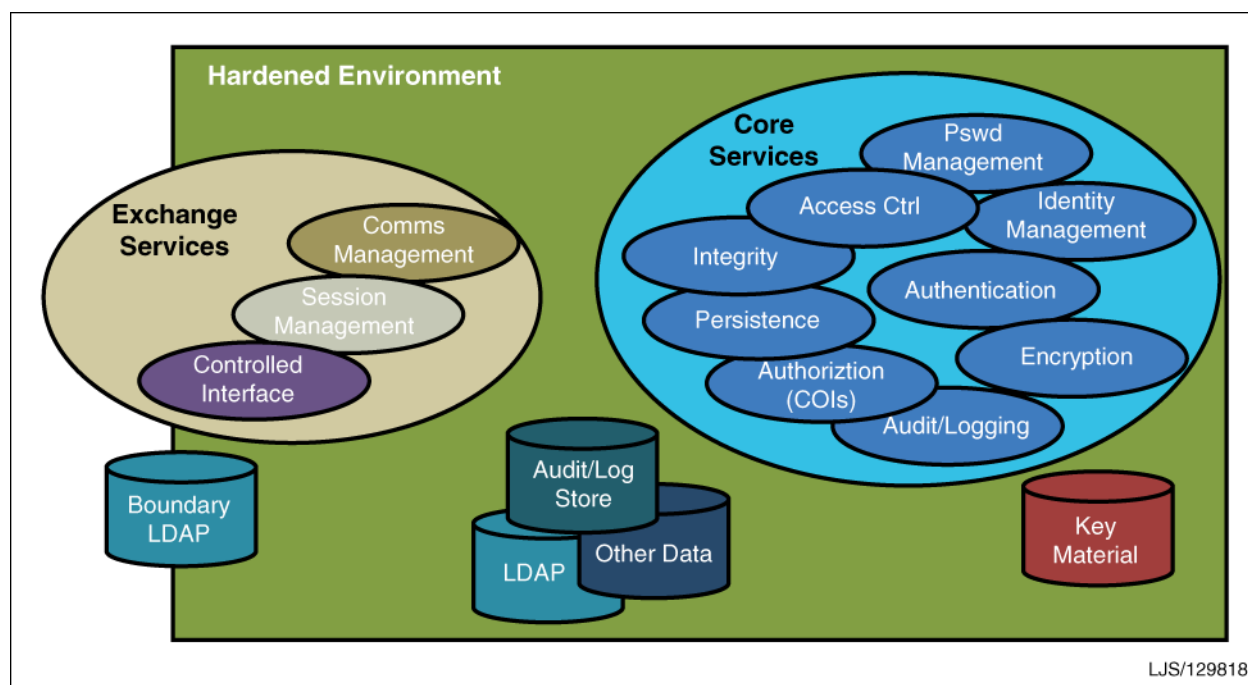


FIGURE 36. Hardened RTIF Environment

Supporting the extant security services in the ARMOUR solution is the methodology to be used for all integration and development activities. The development team will adopt the GD Canada Integrated Systems Security Engineering Process (ISSE) CSP-SYS-013 which identifies all the activities to be performed as based on the ITSG-33 security controls. Amongst the security controls that will be applied are those associated integration and secure software development.

Use or disclosure of this data is subject to the restriction on the title page of this document.

Details on the applicability of the security controls in support of certification and accreditation will be captured in SD008, Certification and Accreditation Plan.

4.1 One-Way Data Diode

Interactions to other domains can be restricted by using Data diodes, also known as a unidirectional network or unidirectional security gateway Figure 37 depicts the use of a data diode enforcing traffic flow between two networks.

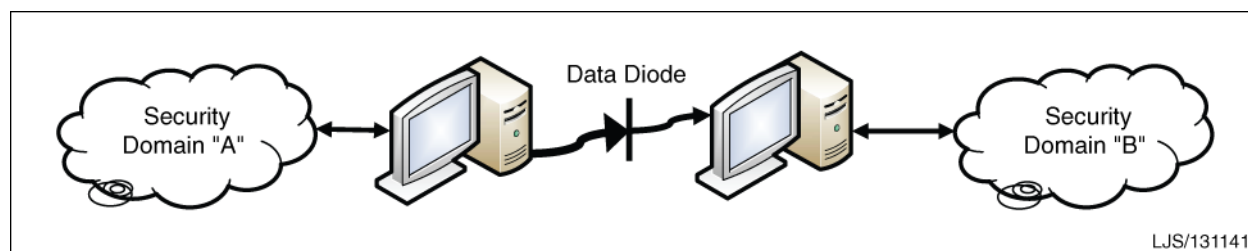


FIGURE 37. Data Diode Flow Control

As ARMOUR is deployed within its own enclave, the collection of data from Data Sources and the transmission of data to Effector technologies hosted in the infrastructure or the managed network (i.e., DRENet) is managed via a secure perimeter and interface point.

Separate one-way diodes can be used on the data source or ingestion interface and on the effector interface to transfer effector outputs to the target network.

There are primarily three variants of Data Diodes

1. The most basic form of data diode is based on fibre optical Tx/Rx where only one direction of transceivers are deployed and removed in the other direction. This technique of physical disconnection is used in most of the commercial Data Diode products

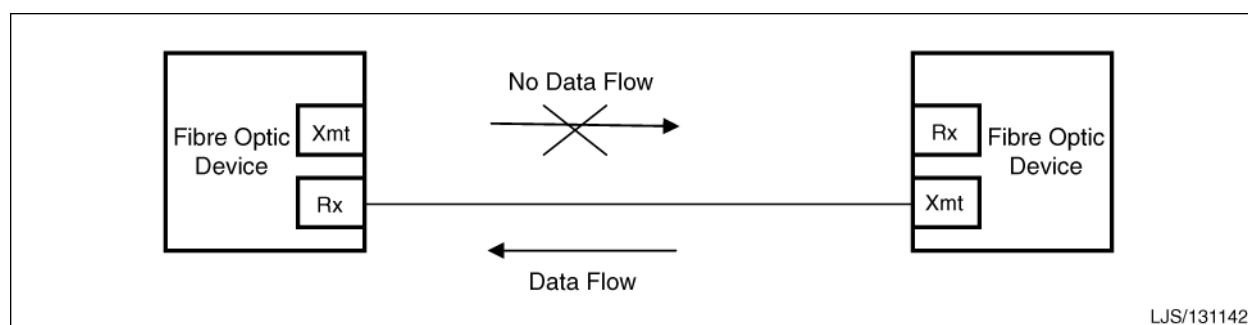


FIGURE 38. Data Diode Internals

2. Some Data Diode vendors allow the use of protocol that requires Bi-Directional links. Some commercial offerings use proprietary protocols that allow for data transfer from protocols like TCP/IP that usually require bidirectional links. This is usually done by using TCP/IP client-server proxies.

Use or disclosure of this data is subject to the restriction on the title page of this document.

3. The US Naval Research Laboratory (NRL) has their own version of Data Diode called Data Pump. Data pump allows a limited back channel to send the acknowledgments back.

Some primary vendors of Data Diodes are

1. **Nexor Data Diode** - Data Flow in Single direction. Nexor provides various software packages to use Diode for updates and monitoring, and to provide support for application traffic, file transfers and database replication.
2. **Genua vs-Diode** - Software based solution. A feedback channel in the opposite direction to send status information allows the use of bidirectional applications like FTP, SMTP etc.
3. **Fox-IT Data Diode** - Hardware based data diode. Proxy based solutions to allow the use of various bidirectional applications.
4. **OWL Computing Technologies Dual Diode** - Data diode based on their patented Dual diode technology. Provides software packages to use Dual Diode for various applications like File transfer, data transfer across different network domains etc.
5. **Waterfall Unidirectional Gateway** - Fiber optics hardware based data diode. Software packages available for specific application (i.e., File transfer, Database duplication etc.).

A Decision Analysis and Resolution (DAR) can be performed to help select the best-fit solution to fulfil the flow control requirements imposed on ARMOUR.

4.2 Preventative Measures for Software Modules

In addition to the ARMOUR IF security services or features deployed in a hardened operating environment, the ARMOUR software modules will be developed with the following measures to help protect software against cyber attacks:

- Perform code reviews and testing using appropriate tools and libraries;
- Perform vulnerability assessment testing and remediation;
- Perform penetration testing and remediation as required; and
- Following best practices for preventative measures such as those covered in Open Web Application Security Project (OWASP) Top Ten most critical web application security flaws or SANS Top 20 critical security control.

Use or disclosure of this data is subject to the restriction on the title page of this document.

5. PERFORMANCE

The ARMOUR System is designed with functionality, cost and performance as top criteria. Overall, the intent of the ARMOUR TD is to provide a CND solution with minimal latency in the collection, storage, retrieval, analysis and effectuation of data while maintaining acceptable IT resource loading (processor loading, RAM usage, etc.). Performance metrics need to be evaluated and tested for each subsystem/module and their respective interfaces to ensure acceptable end-to-end functionality. Additionally, products need to be evaluated for their capability to provide expansion while maintaining acceptable performance.

The ARMOUR TD is designed to meet the performance requirements outlined in the System Technical Specification.

If one of the capabilities is determined to lack the performance to sufficiently meet the needs of the system, considerations need to be taken to employ an alternate solution based on cost, functionality and performance.

As an example, should ARMOUR IF be determined to have excessive latency when interfacing with the data store, an alternate API that provides better performance will be evaluated. Emphasis on the selection of another interface needs to be given to capabilities that exist within current subsystems/modules in an effort to reduce the cost of introducing a new solution.

Individual subsystem/module, interface and overall system performance are further discussed in the ARMOUR TD Detailed Design Document (GD Canada document No. 741349).

Use or disclosure of this data is subject to the restriction on the title page of this document.

6. NOTES

6.1 Abbreviations

The abbreviations that follow are used in this document.

ACL	Access Control List
ADD	Architectural Design Document
AJAX	Asynchronous JavaScript and XML
ANSI-SQL	American National Standards Institute Structured Query Language
API	Application Programming Interface
ARMOUR	Automated Computer Network Defence
AVSAM	Attack/Vulnerability Situational Analysis Module
C2IEDM	Command and Control Information Exchange Data Model
CANSOFCOM	Canadian Special Operations Forces Command
CAS	Central Authentication Service
CDM	Conceptual Data Model
CDRL	Contract Data Requirements List
CEF	Common Event Format
CEP	Complex Event Processing
CERN	European Organization for Nuclear Research
CIA	Common Infrastructure Abstraction
CIAP	Consolidated Information Assurance Picture
CND	Computer Network Defence
COA	Course Of Action
COADS	Course Of Action Decision Support
CONOPs	Concept of Operations
COP	Common Operating Picture
COTS	Commercial Off-The-Shelf
CPE	Common Platform Enumeration
CSC	Cross Source Correlation
CSIR	Consolidated Security Information Repository
CSS	Cascading Style Sheets
CSV	Comma Separated Value
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
DAR	Decision Analysis and Resolution
DB	Database
DBMS	Database Management Service
DDD	Detailed Design Document
DDS	Data Distribution Standard
DID	Data Item Description
DM	Data Model

Use or disclosure of this data is subject to the restriction on the title page of this document.

DMS	Digital MainSpring
DMZ	Demilitarized Zone
DND	Department of National Defence
DNS	Domain Name Service
DPM	Dependence Metric
DRDC	Defence Research and Development Canada
DREnet	Defence Research Establishment net
EIF	Enterprise Integration Framework
EIP	Enterprise Integration Pattern
eSAC	electronic Software Authorization Checklist
ESB	Enterprise Service Bus
ETL	Extract, Transform, Load
FAA	Federal Aviation Authority (U.S.)
FTP	File Transfer Protocol
GD Canada	General Dynamics Canada Ltd
GD Team	General Dynamics Team
GFI	Government Furnished Information
GUI	Graphical User Interface
HMI	Human-Machine Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
I&A	Identification and Authentication
ID	Identification
IDK	Integration Development Kit
IDS	Intrusion Detection Systems
IF	Integration Framework
IP	Internet Protocol
IPS	Intrusion Prevention System
ISSE	Integrated Systems Security Engineering
IT	Information Technology
JC3IEDM	Joint Consultation, Command and Control Information Exchange Data Model
JMS	Java Messaging System
JPA	Java Persistence Adapter
KML	Keyhole Mark-up Language
KPI	Key Performance Indicator
KSU	Kansas State University
LDAP	Lightweight Directory Access Protocol
LHC	Large Hadron Collider

Use or disclosure of this data is subject to the restriction on the title page of this document.

MAC	Media Access Control
MIB	Management Information Base
MUSIC	Multi Sensor Integration in a Common Operating Environment
NAT	Network Address Translation
NATO	North Atlantic Treaty Organization
NIST	National Institute of Standards and Technology
NRL	Naval Research Laboratory
NSA	National Security Agency
NVD	National Vulnerability Database
OA	Operational Authority
OASIS	Open Artwork System Interchange Standard
OIA	Operations and Infrastructure Analyzer
OLE	Object Linking and Embedding
OMG	Object Management Group
ONRDM	OIA Network Resource Dependency Module
OODA	Observe, Orient, Decide and Act
OOIM	OIA Operations Information Module
OPV	Operational Priority Values
ORBAT	Order of Battle
ORDDMS	Object-Relational DBMS
ORM	Object-Relational Mapping
OSGi or OSGI	Open Services Gateway Initiative
OSS	Open Source Software
OWF	Ozone Widget Framework
PAT	Port and Address Translation
PCI DSS	Payment Card Industry Data Security Standard
PDA	Personal Digital Assistant
PDF	Portable Data Format
PKI	Public Key Infrastructure
QoS	Quality of Service
R&D	Research and Development
RCP	Remote Procedure Call
RD	Reference Document
RDBMS	Relational Database Management System
RFP	Request for Proposal
RIA	Rich Internet Application
RLI	Remediation Latency Indicator
RPC	Remote Procedure Call
RSS	Rich Site Summary
RTIF	Rapid Technology Integration Framework
SCCM	System Center Configuration Manager

Use or disclosure of this data is subject to the restriction on the title page of this document.

SCP	Secure Copy Protocol
SDO	Service Data Object
SEIM	Security Event and Incident Management
SFTP	Secure File Transfer Protocol
SIEM	Security Information and Event Management
SLA	Service Level Agreement
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SOPES	Shared Operational Picture Exchange Services
SOW	Statement of Work
SPM	Security Posture Metric
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Socket Layer
STS	System Technical Specification
TBD	To Be Determined
TCP	Transmission Control Protocol
TD	Technology Demonstration
TDP	Technology Demonstration Project
TLS	Transport Layer Security
TRL	Technology Readiness Level
UDDI	Universal Description, Discovery and Integration
UDP	User Datagram Protocol
UI	User Interface
US	United States
VLI	Vulnerability Level Indicator
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WMI	Windows Management Instrumentation
WSDL	Web Services Description Language
WSSS	Web Services Standards Support
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language

Use or disclosure of this data is subject to the restriction on the title page of this document.

This page is left blank intentionally.

Use or disclosure of this data is subject to the restriction on the title page of this document.

APPENDIX A

SRS Requirements Mapping

Use or disclosure of this data is subject to the restriction on the title page of this document.

This page is left blank intentionally.

Use or disclosure of this data is subject to the restriction on the title page of this document.

A. SRS REQUIREMENTS MAPPING

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_29	Requirement	The ARMOUR solution shall provide an integrated CND capability.	All	1
ARMR_SRS_39	Requirement	The ARMOUR solution CND capability shall automate a closed loop operational process supporting the OODA (Observe, Orient, Decide, Act) decision cycle applied to CND.	All	1
ARMR_SRS_53	Requirement	The ARMOUR solution shall include an integration framework to manage integration and communication between system components (data source connectors, database, data presentation, computational services component and effector technologies).	All	3.1.1
ARMR_SRS_56	Requirement	The ARMOUR solution shall include a database component.	Phase 2	3.4.2
ARMR_SRS_57	Requirement	The ARMOUR solution database component shall provide a data store.	Phase 2	3.4.2
ARMR_SRS_58	Requirement	The ARMOUR solution database component shall provide Database Management System (DBMS).	Phase 2	3.4.2
ARMR_SRS_59	Requirement	The ARMOUR solution shall include a data presentation component.	Phase 2	3.4.3
ARMR_SRS_60	Requirement	The ARMOUR solution data presentation component shall provide a Graphical User Interface (GUI) module.	Phase 2	3.4.3
ARMR_SRS_61	Requirement	The ARMOUR solution GUI module shall support data display for the user.	Phase 2	3.4.3.3
ARMR_SRS_62	Requirement	The ARMOUR solution GUI module shall support data input from a user.	Phase 2	3.4.3.3
ARMR_SRS_63	Design Goal	It is desirable that the ARMOUR solution data presentation component provides an interface to various "effectors" (interfaces to technologies that effectuate COAs).	Phase 2	3.4.3.1
ARMR_SRS_68	Design Goal	It is desirable that the ARMOUR solution be modular at all levels allowing modules and sub-modules within modules to be removed and replaced with alternate modules (or sub-modules) providing similar functionality.	Phase 2	3.1.1
ARMR_SRS_70	Requirement	The ARMOUR solution integration framework shall support the choice of the following user interfaces: a. Thick; b. Thin; and c. Web-based.	Phase 2	3.3.1.3
ARMR_SRS_71	Requirement	The ARMOUR solution integration framework shall be a Service Oriented Architecture (SOA)	Phase 2	3.1
ARMR_SRS_72	Requirement	The ARMOUR solution SOA architecture used shall be standards-based.	Phase 2	3.1.1 3.4.1
ARMR_SRS_73	Requirement	The ARMOUR solution integration framework shall support standards-based web services.	Phase 2	3.4.1
ARMR_SRS_74	Requirement	The ARMOUR solution integration framework shall support Extensible Markup Language (XML) standards-based interfaces between system components (data source connectors, database, data presentation, computational services component and effectors).	Phase 2	3.1.1

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_75	Design Goal	It is desirable that modules within the integration framework communicate via the integration framework messaging services.	Phase 2	3.1.1
ARMR_SRS_76	Design Goal	It is desirable that modules within the computational services component not communicate directly with each other.	Phase 2	3.1.1
ARMR_SRS_79	Info	The ARMOUR solution integration framework may use proprietary interfaces between system components where required to meet performance requirements.	Phase 2	5
ARMR_SRS_80	Requirement	The ARMOUR solution integration framework shall support a modular environment.	Phase 2	3.1.1
ARMR_SRS_82	Requirement	The ARMOUR solution integration framework shall include a mechanism to alert processing modules within the computational services component when data is available.	Phase 2	3.1.1
ARMR_SRS_83	Requirement	The ARMOUR solution integration framework shall include a mechanism to identify processes that are taking too long (e.g., stalled processes).	Phase 2	3.1.1
ARMR_SRS_84	Requirement	The ARMOUR solution integration framework shall include a mechanism to recover from processes that are taking too long (e.g., stalled processes).	Phase 2	3.1.1
ARMR_SRS_85	Requirement	The ARMOUR solution integration framework shall include a mechanism to configure the criteria for identifying processes that are taking too long, such as a time limit, running process, or inspection of log events.	Phase 2	3.1.1
ARMR_SRS_86	Requirement	The ARMOUR solution integration framework shall include a mechanism to enable modules to query the progress of other modules.	Phase 2	3.1.1
ARMR_SRS_87	Requirement	The ARMOUR solution integration framework shall support a GUI.	Phase 2	3.1.1
ARMR_SRS_88	Requirement	The ARMOUR solution integration framework shall support both web client and stand-alone client GUI technologies.	Phase 2	3.3.1.3
ARMR_SRS_89	Design Goal	It is desirable that the GUI technology selected be Linux compatible.	Phase 2	3.3.1.3
ARMR_SRS_90	Design Goal	It is desirable that the GUI technology selected be Microsoft Windows 7 compatible.	Phase 2	3.3.1.3
ARMR_SRS_91	Design Goal	It is desirable that the GUI technology selected be Apple Mac OS X compatible.	Phase 2	3.3.1.3
ARMR_SRS_92	Requirement	The ARMOUR solution integration framework shall include a mechanism for report generation.	Phase 2	3.1.1
ARMR_SRS_93	Requirement	The ARMOUR solution integration framework shall include a mechanism to retrieve generated reports.	Phase 2	3.1.1
ARMR_SRS_94	Requirement	The ARMOUR solution integration framework shall include a mechanism to print reports generated.	Phase 2	B.5
ARMR_SRS_95	Requirement	The ARMOUR solution integration framework shall include a mechanism to develop new reports.	Phase 2	3.1.1
ARMR_SRS_96	Requirement	The ARMOUR solution integration framework shall include a mechanism to modify existing reports.	Phase 2	B.5

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_97	Requirement	The ARMOUR solution integration framework shall include a data verification module	Phase 2	3.1.1
ARMR_SRS_98	Requirement	The ARMOUR solution data verification module shall provide an Application Programming Interface (API) from the ARMOUR System to the effector connectors.	Phase 2	3.1.1
ARMR_SRS_99	Requirement	The ARMOUR solution data verification module shall provide an API from the data source connectors to the ARMOUR System	Phase 2	3.1.1
ARMR_SRS_100	Requirement	The ARMOUR solution data verification module shall verify the data format of externally received data against a configurable set of policies.	Phase 2	3.1.1
ARMR_SRS_101	Requirement	The ARMOUR solution data verification module shall verify the data format and content of data to be released from the ARMOUR solution against a configurable set of policies.	Phase 2	3.1.1
ARMR_SRS_102	Requirement	The ARMOUR solutions configurable set of policies applied to the data verification module shall be able to contain a minimum of 200 incoming and 200 outgoing policies.	Phase 2	3.1.1
ARMR_SRS_103	Requirement	The ARMOUR solutions configurable set of policies shall be specified using a standard policy definition language (e.g., XACML or alternative OASIS or W3C standard language).	Phase 2	3.1.1
ARMR_SRS_104	Requirement	The ARMOUR solutions incoming data verification policies shall be able to verify that no executable code is received.	Phase 2	3.1.1
ARMR_SRS_105	Requirement	The ARMOUR solutions incoming data verification policies shall be able to verify that the format of data received matches the Data Model of the database.	Phase 2	3.1.1
ARMR_SRS_106	Requirement	The ARMOUR solutions outgoing data verification policies shall be able to verify that the format of the data to be released matches the effector connector requirements.	Phase 2	3.1.1
ARMR_SRS_107	Requirement	The ARMOUR solutions outgoing data verification policies shall be able to verify that no executable code is released.	Phase 2	3.1.1
ARMR_SRS_108	Requirement	The ARMOUR solutions integration framework shall include a mechanism to create the configurable set of policies used by the data verification module.	Phase 2	3.4.3.4.7.1
ARMR_SRS_630	Requirement	The ARMOUR solutions integration framework shall include a mechanism to modify the configurable set of policies used by the data verification module.	Phase 2	3.4.3.4.7.1
ARMR_SRS_631	Requirement	The ARMOUR solutions integration framework shall include a mechanism to delete the configurable set of policies used by the data verification module.	Phase 2	3.4.3.4.7.1
ARMR_SRS_187	Requirement	The ARMOUR solutions access to the database component shall be through an API.	Phase 2	3.4.2
ARMR_SRS_188	Requirement	The ARMOUR solution API shall include, syntax for: a. Functions; b. Parameters; c. Valid values; d. Error codes; and e. Expected results.	Phase 2	3.4.2
ARMR_SRS_189	Requirement	The ARMOUR solution API shall be fully documented and maintained.	Phase 2	3.4.2
ARMR_SRS_190	Requirement	The ARMOUR solution API shall be under version control.	Phase 2	3.4.2

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_191	Requirement	The ARMOUR solution API shall supply Structured Query Language (SQL) standard interfaces.	Phase 2	3.4.2
ARMR_SRS_192	Requirement	The ARMOUR solution API shall supply web services standard interfaces.	Phase 2	3.1.1 3.4.2
ARMR_SRS_193	Requirement	The ARMOUR solution API shall be extensible.	Phase 2	3.1.1 3.4.2
ARMR_SRS_194	Requirement	The ARMOUR solution database component shall include a data store.	Phase 2	3.4.2
ARMR_SRS_195	Requirement	The ARMOUR solution database component shall include a DBMS.	Phase 2	3.4.2
ARMR_SRS_196	Requirement	The ARMOUR solution data store shall be implemented using a relational database.	Phase 2	3.4.2
ARMR_SRS_197	Requirement	The ARMOUR solution relational database shall support ANSI-SQL-2003 or later.	Phase 2	3.3.1.4
ARMR_SRS_199	Requirement	The ARMOUR solution relational database shall support Open Database Connectivity (ODBC).	Phase 2	3.3.1.4
ARMR_SRS_198	Requirement	The ARMOUR solution relational database shall support Java Database Connectivity (JDBC).	Phase 2	3.4.2
ARMR_SRS_200	Requirement	The ARMOUR solution relational database shall support web services interface standards.	Phase 2	3.4.2
ARMR_SRS_201	Requirement	The ARMOUR solution relational database shall process transactions reliably using Atomic, Consistent, Isolated, Durable (ACID) properties.	Phase 2	3.3.1.4
ARMR_SRS_202	Requirement	The ARMOUR solution relational database API calls shall be compatible with at least one of the following commercial products: a. Oracle Database; or b. Microsoft SQL Server. Acceptable versions are those current at contract award, or later.	Phase 2	3.4.2
ARMR_SRS_203	Requirement	The ARMOUR solution relational database syntax shall be compatible with at least one of the following commercial products: a. Oracle Database; or b. Microsoft SQL Server. Acceptable versions are those current at contract award, or later.	Phase 2	3.4.2
ARMR_SRS_204	Requirement	The ARMOUR solution relational database syntax shall be compatible with at least one of the following OSS products: a. PostgreSQL; or b. MySQL. Acceptable versions are those current at contract award, or later.	Phase 2	3.4.2
ARMR_SRS_205	Requirement	The ARMOUR solution relational database API calls shall be compatible with at least one of the following OSS products: a. PostgreSQL; or b. MySQL. Acceptable versions are those current at contract award, or later.	Phase 2	3.4.2
ARMR_SRS_206	Requirement	The ARMOUR solution data store shall be used to store all raw data received from the data source connectors.	Phase 2	3.4.2
ARMR_SRS_207	Requirement	The ARMOUR solution data store shall be used to store all persistent data resulting from data manipulations performed by the computational services component.	Phase 2	3.4.2

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_208	Requirement	The ARMOUR solution computational services component processing shall be able to work on a consistent view of the data during a long processing time.	Phase 2	3.4.4
ARMR_SRS_209	Requirement	The ARMOUR solution data store shall be used to store all persistent data entered from the data presentation component.	Phase 2	3.4.3.1
ARMR_SRS_210	Requirement	The ARMOUR solution Data Model shall be fully documented and maintained.	Phase 2	3.1.2
ARMR_SRS_211	Requirement	The ARMOUR solution Data Model shall be documented using Unified Modeling Language (UML).	Phase 2	3.1.4.3.1 3.1.4.4.1 3.1.4.5.1 3.1.4.7.1
ARMR_SRS_212	Requirement	The ARMOUR solution Data Model shall be under version control.	Phase 2	3.1.4
ARMR_SRS_213	Requirement	The ARMOUR solution Data Model shall support: a. Actual; b. Historic; c. Simulated; and d. Projected data.	Phase 2	3.1.4
ARMR_SRS_214	Requirement	The ARMOUR solution actual data shall include the most current information available from data sources.	Phase 2	3.4.1
ARMR_SRS_215	Requirement	The ARMOUR solution actual data shall include the most current information manually entered by a user.	Phase 2	3.4.3
ARMR_SRS_216	Requirement	The ARMOUR solution actual data shall include the most current information resulting from processing performed by the computational services component.	Phase 2	3.4.4
ARMR_SRS_224	Requirement	The ARMOUR solution DBMS shall include database installation services.	Phase 2	3.4.2
ARMR_SRS_225	Requirement	The ARMOUR solution DBMS shall include database configuration services.	Phase 2	3.3.1.4
ARMR_SRS_226	Requirement	The ARMOUR solution DBMS shall include database backup services.	Phase 2	3.3.1.4
ARMR_SRS_227	Requirement	The ARMOUR solution DBMS shall include database restore services.	Phase 2	3.3.1.4
ARMR_SRS_228	Requirement	The ARMOUR solution DBMS shall include database performance tuning services.	Phase 2	3.3.1.4
ARMR_SRS_229	Requirement	The ARMOUR solution DBMS shall include database access control configuration.	Phase 2	3.4.2
ARMR_SRS_231	Requirement	The ARMOUR solution modules within the computational services component shall be able to operate on a coherent version of the data store, without need for interaction with other modules or processes.	Phase 2	3.4.4
ARMR_SRS_232	Requirement	The ARMOUR solution modules within the computational services component shall include progress reporting on each processing function that could take more than 1 second.	Phase 2	3.1.1 3.4.4
ARMR_SRS_421	Requirement	The ARMOUR solution data presentation component shall include a GUI module.	Phase 2	3.4.3

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_422	Requirement	The ARMOUR solution GUI module shall be modular allowing display sub-modules to be removed and replaced with alternate sub-modules providing similar functionality.	Phase 2	3.1.4
ARMR_SRS_423	Requirement	The ARMOUR solutions design of the ARMOUR GUI module shall follow design guidance provided in the ARMOUR STS Text.	Phase 2	1
ARMR_SRS_424	Requirement	The ARMOUR solutions design of the ARMOUR GUI shall support software localization to enable a multi-lingual interface.	Phase 2	3.4.3.1
ARMR_SRS_425	Requirement	The ARMOUR solution GUI shall provide an interface for user selection of the desired language.	Phase 2	3.4.3.1
ARMR_SRS_427	Requirement	The ARMOUR solution GUI module shall support at least 8 concurrent users (of any type).	Phase 2	3.4.3.1
ARMR_SRS_431	Requirement	The ARMOUR solution GUI module shall support data entry.	Phase 2	3.4.3.1
ARMR_SRS_432	Requirement	The ARMOUR solution GUI module shall support 2-Dimensional graphic displays.	Phase 2	3.4.3.2
ARMR_SRS_433	Requirement	The ARMOUR solution 2-Dimensional graphic display shall provide the user the ability to zoom in and zoom out.	Phase 2	3.4.3.2
ARMR_SRS_434	Requirement	The ARMOUR solution 2-Dimensional graphic display shall be manipulable on 2 axes (pan left/right, pan up/down).	Phase 2	3.4.3.2
ARMR_SRS_435	Requirement	The ARMOUR solution GUI module shall support 3-Dimensional graphic displays.	Phase 2	3.4.3.2
ARMR_SRS_436	Requirement	The ARMOUR solution 3-Dimensional graphic display shall provide the user the ability to zoom in and zoom out.	Phase 2	3.4.3.2
ARMR_SRS_437	Requirement	The ARMOUR solution 3-Dimensional graphic display shall be manipulable on 3 axes (rotate left/right, rotate up/down).	Phase 2	3.4.3.2
ARMR_SRS_438	Requirement	The ARMOUR solution GUI module shall enable user manipulation of display attributes (e.g., resizing, undock windows, rearrange window locations, cascading windows, tiling windows).	Phase 2	3.4.3.2
ARMR_SRS_439	Requirement	The ARMOUR solution GUI module shall enable multiple linked views.	Phase 2	3.4.3.3 3.4.3.3.1 3.4.3.4.1.1
ARMR_SRS_440	Requirement	The ARMOUR solution GUI module shall support tabular data views.	Phase 2	3.4.3.3
ARMR_SRS_441	Requirement	The ARMOUR solution tabular data views shall allow data filtering (of the entire record set).	Phase 2	3.4.3.3
ARMR_SRS_442	Requirement	The ARMOUR solution tabular data views shall allow data sorting (of the entire record set).	Phase 2	3.4.3.3
ARMR_SRS_443	Requirement	The ARMOUR solution tabular data views shall allow data marking (e.g., highlighting a cell, row or column).	Phase 2	3.4.3.3
ARMR_SRS_444	Requirement	The ARMOUR solution shall mark in all linked views, the data marked in the tabular view.	Phase 2	3.4.3.3 3.4.3.3.1 3.4.3.4.1.1
ARMR_SRS_445	Requirement	The ARMOUR solution tabular data views shall allow data selection (e.g., marking a cell, row or column for subsequent data manipulation).	Phase 2	3.4.3.3

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_446	Requirement	The ARMOUR solution GUI module shall support both graphical and tabular data presentation within the same view.	Phase 2	3.4.3.3
ARMR_SRS_461	Requirement	The ARMOUR solution display (graphical and tabular) shall include methods to aggregate data sets into summary form.	Phase 2	3.4.3.2 3.4.3.4.1.1
ARMR_SRS_462	Requirement	The ARMOUR solution display (graphical and tabular) shall include methods to drill-down to subordinate levels of detail down to host level details.	Phase 2	3.4.3.2
ARMR_SRS_463	Requirement	The ARMOUR solution GUI module shall support additional display modules used to present results specific to modules contained in the Computation Services Component.	Phase 2	3.4.3.1
ARMR_SRS_464	Requirement	The ARMOUR solution shall populate additional display modules within the GUI module from data contained in the database.	Phase 2	3.4.3.1
ARMR_SRS_465	Requirement	The ARMOUR solution data presentation component shall include a reporting module.	Phase 2	3.1.1 3.4.3.4.7.4
ARMR_SRS_466	Requirement	The ARMOUR solution report generation mechanism shall generate structured views of data formatted for export in: a. Portable Document Format (PDF); b. Comma Separated Value (CSV); and c. XML file formats.	Phase 2	3.1.1 3.2
ARMR_SRS_467	Requirement	The ARMOUR solution report generation mechanism shall use configurable templates for report format and data presentation.	Phase 2	3.1.1 3.4.3.4.7.4
ARMR_SRS_468	Design Goal	It is desirable that the ARMOUR solution report generation mechanism allow for the development of new reports and customization of existing reports.	Phase 2	3.1.1 3.2
ARMR_SRS_469	Requirement	The ARMOUR solution report generation mechanism shall support software localization to enable a multi-lingual report generation.	Phase 2	3.1.1 3.2
ARMR_SRS_470	Requirement	The ARMOUR solution report generation mechanism shall support language localization through user selection of the desired report language.	Phase 2	3.1.1 3.2
ARMR_SRS_499	Requirement	The ARMOUR solution systems shall be hardened in accordance with industry best practice (e.g., unnecessary host services disabled, host services operate in least privileged accounts, etc).	Phase 2, 3, 4, 5	4
ARMR_SRS_500	Requirement	The ARMOUR solution shall be designed to prevent cross site scripting, buffer overflow and similar cyber attacks.	Phase 2, 3, 4, 5	4.2
ARMR_SRS_501	Requirement	The ARMOUR solution systems shall undergo vulnerability assessment testing.	Phase 2, 3, 4, 5	4.2
ARMR_SRS_502	Requirement	The ARMOUR solution shall address issues identified in the vulnerability assessment prior to planned demonstrations.	Phase 2, 3, 4, 5	4.2
ARMR_SRS_503	Requirement	The ARMOUR solution systems shall undergo penetration testing.	Phase 2, 3, 4, 5	4.2
ARMR_SRS_504	Requirement	The ARMOUR solution shall address issues identified in the penetration testing prior to planned demonstrations.	Phase 2, 3, 4, 5	4.2
ARMR_SRS_505	Requirement	The ARMOUR solution shall control access to the ARMOUR system via user login.	Phase 2	4

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_506	Requirement	The ARMOUR solution user login shall be based on a User Identifier (ID) and password combination.	Phase 2	4 3.2
ARMR_SRS_507	Requirement	The ARMOUR solution access shall support two factor authentication for user login.	Phase 2	4 3.2
ARMR_SRS_508	Requirement	The ARMOUR solution user logins shall create a unique Session ID.	Phase 2	4 3.2
ARMR_SRS_509	Requirement	The ARMOUR solution users shall be automatically logged out from their session after a configurable period.	Phase 2	4 3.2
ARMR_SRS_510	Requirement	The ARMOUR solution users shall be prompted to logout when exiting the system.	Phase 2	4 3.2
ARMR_SRS_511	Requirement	The ARMOUR solution users shall be identified by a unique login identifier (User ID).	Phase 2	4 3.2
ARMR_SRS_512	Design Goal	It is desirable that the ARMOUR solution administrator accounts not use default naming or setup.	Phase 2	4 3.2
ARMR_SRS_513	Design Goal	It is desirable that the ARMOUR solution Default and Guest User IDs be disabled.	Phase 2	4 3.2
ARMR_SRS_514	Requirement	The ARMOUR solution User IDs shall be authenticated using standards based best practices.	Phase 2	4 3.2
ARMR_SRS_515	Requirement	The ARMOUR solution authentication shall employ strong passwords including: a. A minimum of eight characters; b. Both letters and numbers required; c. Combination of upper and lower case required; d. At least one special character or symbol required; and e. No dictionary words.	Phase 2	4 3.2
ARMR_SRS_516	Requirement	The ARMOUR solution password parameters comprising the strong password shall be configurable.	Phase 2	4 3.2
ARMR_SRS_517	Requirement	The ARMOUR solution user passwords shall expire after a configurable period of time.	Phase 2	4 3.2
ARMR_SRS_518	Requirement	The ARMOUR solution shall require change of expired passwords on the next login attempt.	Phase 2	4 3.2
ARMR_SRS_519	Requirement	The ARMOUR solution password changes shall require user to enter the current password prior to change.	Phase 2	4
ARMR_SRS_520	Requirement	The ARMOUR solution system shall enforce forgotten passwords to be changed.	Phase 2	4
ARMR_SRS_521	Requirement	The ARMOUR solution shall log all password changes.	Phase 2	4
ARMR_SRS_522	Requirement	The ARMOUR solution system shall trigger a user lock-out after a number of failed logins attempts.	Phase 2	4
ARMR_SRS_523	Requirement	The ARMOUR solution system shall provide a method to configure the number of attempts leading to a user lock-out.	Phase 2	4
ARMR_SRS_524	Design Goal	It is desirable that the ARMOUR solution system enforce the account lock-out to be maintained for a configurable number of hours.	Phase 2	4

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_525	Requirement	The ARMOUR solution shall rely on the assigned user roles for authorizing access to system functions.	Phase 2	3.4.3
ARMR_SRS_526	Requirement	The ARMOUR solution user roles shall be configurable.	Phase 2	4
ARMR_SRS_527	Requirement	The ARMOUR solution configurable user roles shall include: a. Proactive security analyst; b. Reactive security analyst; c. Network operator; d. System/configuration manager; e. System administrator; and f. Commander.	Phase 2	3.4.3
ARMR_SRS_528	Requirement	The ARMOUR solution system functions shall be assigned, as available, to one or more user roles.	Phase 2	3.4.3
ARMR_SRS_529	Requirement	The ARMOUR solution users shall be assigned one or more user roles, thereby defining authorization to access system functions.	Phase 2	3.4.3
ARMR_SRS_530	Requirement	The ARMOUR solution shall authenticate communication from data sources to data source connectors.	Phase 2	4
ARMR_SRS_531	Design Goal	It is desirable that the ARMOUR solution encrypt communication from data sources to data source connectors.	Phase 2	4
ARMR_SRS_532	Requirement	The ARMOUR solution communication from data source connectors to the ARMOUR system shall be authenticated.	Phase 2	4
ARMR_SRS_533	Requirement	The ARMOUR solution communication from the ARMOUR system to the effector connectors shall be authenticated.	Phase 2	4
ARMR_SRS_534	Requirement	The ARMOUR solution communication from the effector connectors to the effector technologies shall be authenticated.	Phase 2	4
ARMR_SRS_535	Design Goal	It is desirable that the ARMOUR solution encrypt communications from the effector connectors to the effector technologies.	Phase 2	4
ARMR_SRS_536	Design Goal	It is desirable that the ARMOUR solution inter-module communication within the ARMOUR system be authenticated.	Phase 2	4
ARMR_SRS_537	Requirement	The ARMOUR solution inter-module communication within the ARMOUR system shall be authenticated where communication is distributed across multiple hosts comprising the ARMOUR solution.	Phase 2	4
ARMR_SRS_538	Requirement	The ARMOUR solution data source connectors shall only permit a write connection from the data source connectors to the ARMOUR database component.	Phase 2	4.1
ARMR_SRS_539	Requirement	The ARMOUR solution communication link to the database component from the data source connectors shall employ DND standards based practices to enforce the one-way feed (e.g., One Way Diodes).	Phase 2	4.1
ARMR_SRS_540	Requirement	The ARMOUR solution effector connectors shall only permit a write connection from the data presentation component to the effector connectors.	Phase 2	4.1
ARMR_SRS_541	Requirement	The ARMOUR solution communication link from the data presentation component to the effector connectors shall employ DND standards based practices to enforce a one-way feed (e.g., one-way diodes).	Phase 2	4.1
ARMR_SRS_542	Requirement	The ARMOUR solution integrity of the system data shall be maintained by input data validation (type, length, format, range).	Phase 2	3.1.1
ARMR_SRS_543	Requirement	The ARMOUR solution data that fails input data validation shall be rejected.	Phase 2	3.1.1

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_544	Requirement	The ARMOUR solution XML input data shall be validated based on an established XML schema.	Phase 2	3.1.1
ARMR_SRS_545	Requirement	The ARMOUR solution storage of system security data (encryption keys, certificates, passwords, etc.) shall only be stored in encrypted form.	Phase 2	4
ARMR_SRS_546	Requirement	The ARMOUR solution security-relevant activities (including inter-module communications) shall be recorded using a trusted-audit mechanism.	Phase 2	4
ARMR_SRS_547	Requirement	The ARMOUR solution system shall log all user access attempts (Login, Logout, failed Login, password changes).	Phase 2	4
ARMR_SRS_548	Requirement	The ARMOUR solution system shall provide the ability to generate administrator notifications by event type.	Phase 2	4
ARMR_SRS_549	Requirement	The ARMOUR solution system shall have administrator configurable levels for logging.	Phase 2	4
ARMR_SRS_550	Requirement	The ARMOUR solution administrator configurable logging levels shall range from error and informational logging (e.g., failed user logins) to debug and audit logging.	Phase 2	4
ARMR_SRS_551	Requirement	The ARMOUR solution logging levels shall be clearly defined and documented.	Phase 2	4
ARMR_SRS_552	Requirement	The ARMOUR solution logs shall include: a. Date/time; b. Success/failure; c. Requesting user/process; and d. Resources authorized.	Phase 2	4 3.2
ARMR_SRS_553	Requirement	The ARMOUR solution system shall provide a trusted means to review all audit logs.	Phase 2	4 3.2
ARMR_SRS_554	Requirement	The ARMOUR solution trusted-audit mechanism shall record audit logs to secure storage.	Phase 2	4 3.2
ARMR_SRS_555	Requirement	The ARMOUR solution secure storage shall only be readable by authorized users in possession of an auditor token.	Phase 2	4 3.2
ARMR_SRS_556	Requirement	The ARMOUR solution shall numerically identify all Common and Error logs.	Phase 2	4 3.2
ARMR_SRS_557	Design Goal	It is desirable that the ARMOUR solution provide a documented description of all Common and Error logs for troubleshooting purposes.	Phase 2	4 3.2
ARMR_SRS_30	Requirement	The ARMOUR solution shall support a CND capability that is able to analyze the network for attacks and identify courses of action in advance of an attack.	Proactive - Phase 3,4	1
ARMR_SRS_31	Requirement	The ARMOUR solution shall support a CND capability that is able to analyze the network for attacks and identify courses of action in response to an attack.	Proactive - Phase 3,4	1
ARMR_SRS_34	Requirement	The ARMOUR solution shall support operational security analysis and response while operating in the proactive mode.	Phase 3, 4	3.4.4.2.7 3.4.4.2.8
ARMR_SRS_40	Requirement	The ARMOUR solution shall support the collection of infrastructure information for the network being defended.	Phase 3	3.4.1.1
ARMR_SRS_42	Requirement	The ARMOUR solution shall support the collection of operational information.	Phase 3	3.4.1.4

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_44	Requirement	The ARMOUR solution observe capabilities shall correlate redundant information into a single consistent set of operational and infrastructure data.	Phase 3	3.4.4.2
ARMR_SRS_45	Requirement	The ARMOUR solution orient capabilities shall provide an attack graph to support CND decision making.	Phase 3	3.1.3 3.4.4.2.3
ARMR_SRS_54	Requirement	The ARMOUR solution shall include data source connectors to interface with infrastructure and non-infrastructure (e.g., operational and security reference) data sources.	Phase 3	3.1.1 3.4.1
ARMR_SRS_55	Requirement	The ARMOUR solution data source connectors shall include mechanisms to sources of infrastructure information required by the ARMOUR solution (e.g., network management systems, vulnerability scanners, vulnerability databases, security incident and event management systems, intrusion detection and prevention systems, enterprise architecture systems).	Proactive - Phase 3, Reactive - Phase 5	3.4.1
ARMR_SRS_64	Requirement	The ARMOUR solution shall include a computational services component.	Phase 3, 4, 5	3.1.1 3.4.4
ARMR_SRS_65	Requirement	The ARMOUR solution computational services component shall provide analysis and computation modules required by the ARMOUR solution.	Phase 3, 4, 5	3.4.4
ARMR_SRS_111	Requirement	The ARMOUR solutions data source connectors shall include mechanisms to collect network infrastructure and non-infrastructure information from data sources.	Phase 3	3.4.1
ARMR_SRS_112	Design Goal	It is desirable that data source connectors provide interfaces to the following data sources: a. Sourcefire Intrusion Prevention System; b. Symantec Endpoint Protection; c. McAfee Firewall; d. McAfee Web Gateway; e. Checkpoint Firewall; f. DB Protect; and g. NetScout.	Phase 3	3.4.1.2
ARMR_SRS_113	Requirement	The ARMOUR solutions data source connectors shall collect network infrastructure information from infrastructure based data sources.	Phase 3	3.4.1
ARMR_SRS_114	Requirement	The ARMOUR solutions network infrastructure information shall include routing configurations.	Phase 3	3.4.1.1
ARMR_SRS_115	Requirement	The ARMOUR solutions network infrastructure information shall include network topology.	Phase 3	3.4.1.1
ARMR_SRS_116	Requirement	The ARMOUR solutions network infrastructure information shall include physical topology (e.g., presence of physically separate host systems and their connectivity).	Phase 3	3.4.1.1
ARMR_SRS_117	Design Goal	It is desirable that the network infrastructure information identify hosts which are virtualized.	Phase 3	3.4.1.1
ARMR_SRS_118	Design Goal	It is desirable that the network infrastructure information identify hosts prone to frequent change (e.g., guest systems, laptops).	Phase 3	3.4.1.1
ARMR_SRS_119	Requirement	The ARMOUR solutions network infrastructure information shall include all hosts on the network.	Phase 3	3.4.1.1
ARMR_SRS_120	Requirement	The ARMOUR solutions network infrastructure information shall include host identification information.	Phase 3	3.4.1.1

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_121	Requirement	The ARMOUR solutions network infrastructure information shall be collected from all hosts including: a. Servers; b. End user devices; c. Network appliances (e.g. switches and routers); and d. Security appliances (e.g. firewalls, intrusion prevention systems, and other security devices).	Phase 3	3.4.1.1
ARMR_SRS_122	Requirement	The ARMOUR solutions host identification information shall include host name.	Phase 3	3.4.1.1
ARMR_SRS_123	Requirement	The ARMOUR solutions host identification information shall include host internet protocol address(es).	Phase 3	3.4.1.1
ARMR_SRS_124	Requirement	The ARMOUR solutions host identification information shall include host Media Access Control (MAC) address(es).	Phase 3	3.4.1.1
ARMR_SRS_125	Requirement	The ARMOUR solutions network infrastructure information shall include host configuration information.	Phase 3	3.4.1.1
ARMR_SRS_126	Requirement	The ARMOUR solutions host configuration information shall include software installed.	Phase 3	3.4.1.1
ARMR_SRS_127	Requirement	The ARMOUR solutions identification of software installed in the host configuration information shall clearly identify the software by, at minimum: a. Manufacturer; b. Product; c. Version (e.g., major release); d. Sub-version (e.g., minor release); and e. Patches (e.g., service packs).	Phase 3	3.4.1.1
ARMR_SRS_128	Design Goal	It is desirable that host configuration information identify instances of software vulnerabilities present on the host.	Phase 3	3.4.1.1
ARMR_SRS_129	Requirement	The ARMOUR solution host configuration information shall identify instances of software vulnerabilities present on the host, when the security reference data does not clearly identify vulnerability dependencies on other software products (e.g., where the vulnerability in software product 'A' is only present if software product 'B' is present as well),	Phase 3	3.4.1.4
ARMR_SRS_130	Requirement	The ARMOUR solution shall uniquely identify any software vulnerabilities using a standard vulnerability identifier.	Phase 3	3.4.1.1
ARMR_SRS_131	Requirement	The ARMOUR solution, where host configuration information includes identification of software vulnerabilities present on the host, the identification of the vulnerability shall consider local software dependencies (e.g., the vulnerability is only real if dependent software running on the same host, like an embedded library or underlying operating system dependency, is also present).	Phase 3	3.4.3.4.4
ARMR_SRS_132	Requirement	The ARMOUR solution, where host configuration information includes identification of software vulnerabilities present on the host, the identification of the vulnerability shall consider distributed software dependencies (e.g., the vulnerability is only real if dependent software running on other hosts, like a web server to application server dependency, are also present).	Phase 3	3.4.3.4.4
ARMR_SRS_133	Requirement	The ARMOUR solutions host configuration information shall include host services running at the time of information collection.	Phase 3	3.4.3.4.4
ARMR_SRS_134	Requirement	The ARMOUR solutions host configuration information shall include listening ports at the time of information collection.	Phase 3	3.4.3.4.4
ARMR_SRS_135	Requirement	The ARMOUR solutions list of listening ports shall identify the host services listening on that port.	Phase 3	3.4.4.2.1

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_136	Requirement	The ARMOUR solutions list of listening ports shall identify whether the host service is only accepting local host connections.	Phase 3	3.1.2.7.1
ARMR_SRS_137	Design Goal	It is desirable that the ARMOUR solutions host configuration information include a list of host services running within the last 24 hours prior to the information collection.	Phase 3	3.4.1.1
ARMR_SRS_138	Design Goal	It is desirable that the ARMOUR solutions list of host services running within the last 24 hours include the percentage up-time during that 24 hour period.	Phase 3	3.4.1.1
ARMR_SRS_632	Design Goal	It is desirable that the ARMOUR solutions host configuration information includes a list of listening ports active within the last 24 hours prior to the information collection.	Phase 3	3.4.1.1
ARMR_SRS_139	Design Goal	It is desirable that the ARMOUR solutions list of listening ports active within the last 24 hours include the percentage of time active during that 24 hour period.	Phase 3	3.4.1.1
ARMR_SRS_140	Requirement	The ARMOUR solutions data source connectors shall collect non-infrastructure information from external security reference data sources.	Phase 3	3.4.1
ARMR_SRS_141	Requirement	The ARMOUR solutions data source connectors shall support XML standards-based interfaces between the data source connector and external security reference data sources.	Phase 3	3.4.1.3
ARMR_SRS_142	Requirement	The ARMOUR solution external security reference data sources shall include software vulnerability information and malicious software information in a machine readable structured data format.	Phase 3	3.4.1.3
ARMR_SRS_143	Requirement	The ARMOUR solution software shall use standard vulnerability data sources to identify vulnerability information.	Phase 3	3.4.1.3
ARMR_SRS_633	Design Goal	It is desirable that the ARMOUR solution software vulnerability information use the NATO Cyber Defence Data Exchange and Collaboration Infrastructure as a vulnerability data source if available.	Phase 3	3.4.3.3.1
ARMR_SRS_145	Requirement	The ARMOUR solution software vulnerability information shall use a clearly defined and extensible knowledge base for vulnerability specification.	Phase 3	3.4.1.3
ARMR_SRS_146	Requirement	The ARMOUR solution software vulnerability information shall include metrics describing characteristics of each vulnerability.	Phase 3	3.4.1.3
ARMR_SRS_147	Requirement	The ARMOUR solution characteristics describing the software vulnerability information shall clearly identify the vulnerable software by: a. Manufacturer; b. Product; c. Version (e.g., major release); d. Sub-version (e.g., minor release); and e. Patches (e.g., service packs).	Phase 3	3.4.1.1
ARMR_SRS_148	Design Goal	It is desirable that the ARMOUR solutions characteristics describing the software vulnerability information clearly identify vulnerability dependencies on other software products.	Phase 3	3.4.1.1
ARMR_SRS_149	Design Goal	It is desirable that the ARMOUR solutions characteristics describing the software vulnerability information dependencies clearly identify local software dependencies.	Phase 3	3.4.1.1
ARMR_SRS_150	Design Goal	It is desirable that the ARMOUR solutions characteristics describing the software vulnerability information dependencies clearly identify distributed software dependencies.	Phase 3	3.3.1.6 3.4.1.1 3.4.1.4 3.4.4.2.3

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_151	Requirement	The ARMOUR solution software vulnerability information shall include information related to the availability of exploits targeting the vulnerability.	Phase 3	3.4.1.3
ARMR_SRS_152	Requirement	The ARMOUR solution Software vulnerability metrics shall include a minimum set of metrics compliant with the Common Vulnerability Scoring System (CVSS).	Phase 3	3.4.1.3
ARMR_SRS_153	Design Goal	It is desirable that the ARMOUR solutions malware information use malware attribute enumeration and characterization for malware specification.	Phase 3	3.4.1.1 3.4.1.3
ARMR_SRS_154	Design Goal	It is desirable that the ARMOUR solutions external security reference data sources include attack pattern information.	Phase 3	3.4.1.2 3.4.1.3
ARMR_SRS_155	Design Goal	It is desirable that the ARMOUR solutions attack pattern information use common attack pattern enumeration and classification for attack pattern specification.	Phase 3	3.4.1.2 3.4.1.3
ARMR_SRS_156	Requirement	The ARMOUR solution data Source Connectors shall include the ability to collect cyber security safeguard information.	Phase 3	3.4.1
ARMR_SRS_157	Requirement	The ARMOUR solution shall include firewall configurations for the network being defended.	Phase 3	3.4.1.1
ARMR_SRS_158	Design Goal	It is desirable that the ARMOUR solution security safeguard information include intrusion detection system policies.	Phase 3	3.4.1.1
ARMR_SRS_159	Requirement	The ARMOUR solution security safeguard information shall include intrusion prevention system policies.	Phase 3	3.4.1.1
ARMR_SRS_171	Requirement	The ARMOUR solution data source connectors shall include the ability to collect operational priority information.	Phase 3	3.4.1.4
ARMR_SRS_172	Design Goal	It is desirable that the ARMOUR solution operational priority information be collected by manual data entry in the GUI.	Phase 3	3.4.1.4
ARMR_SRS_173	Requirement	The ARMOUR solution operational priority information shall include a means to uniquely identify operations.	Phase 3	3.4.1.4
ARMR_SRS_174	Requirement	The ARMOUR solution operational priority information shall include a metric to rate operation priority.	Phase 3	3.4.1.4
ARMR_SRS_175	Requirement	The ARMOUR solution operational priority information shall include operational dependency information.	Phase 3	3.4.1.4
ARMR_SRS_176	Requirement	The ARMOUR solution operational dependency information shall include operations to application services dependency relationships.	Phase 3	3.4.1.4
ARMR_SRS_177	Requirement	The ARMOUR solution operational dependency information shall include operations to operations dependency relationships.	Phase 3	3.4.1.4
ARMR_SRS_178	Requirement	The ARMOUR solution operational dependency information shall include application services to application services dependency relationships.	Phase 3	3.4.1.4
ARMR_SRS_179	Requirement	The ARMOUR solution operational dependency information shall include application services to host services dependency relationships.	Phase 3	3.4.1.4
ARMR_SRS_180	Requirement	The ARMOUR solution operational dependency information shall include host services to host services dependency relationships.	Phase 3	3.4.1.4

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_181	Requirement	The ARMOUR solution operational dependency information shall include host services to host system dependency relationships.	Phase 3	3.4.1.4
ARMR_SRS_182	Requirement	The ARMOUR solution dependency relationships shall identify conjunctive (compound) dependencies.	Phase 3	3.4.1.4
ARMR_SRS_183	Requirement	The ARMOUR solution dependency relationships shall identify disjunctive (alternative) dependencies.	Phase 3	3.4.1.4
ARMR_SRS_184	Requirement	The ARMOUR solution disjunctive dependency relationships shall include a metric indicating the preference or probability of selection for each alternative.	Phase 3	3.4.1.4
ARMR_SRS_185	Requirement	The ARMOUR solution dependency relationships shall be representable as a directed forward hyper graph (alternatively, an AND/OR directed graph).	Phase 3	3.4.1.4
ARMR_SRS_234	Requirement	The ARMOUR solution computational services shall include a cross source correlation module.	Phase 3	3.4.4.2
ARMR_SRS_235	Requirement	The ARMOUR solution cross source correlation module shall fuse redundant raw data into a single normalized representation of the data.	Phase 3	3.4.4.2
ARMR_SRS_236	Requirement	The ARMOUR solution cross source correlation module shall fuse redundant: a. Network infrastructure information; b. Host identification information; c. Host configuration information; d. Software vulnerability information; e. Security safeguard information; and f. Operational information, into a single normalized representation.	Phase 3	3.4.4.2
ARMR_SRS_237	Requirement	The ARMOUR solution cross source correlation module shall fuse redundant network events information and security events information into a single normalized representation.	Phase 3	3.4.4.2
ARMR_SRS_238	Requirement	The ARMOUR solution cross source correlation module shall include methods to resolve conflicts within the raw data during data fusion.	Phase 3	3.4.4.2
ARMR_SRS_239	Requirement	The ARMOUR solution cross source correlation module shall save the fused infrastructure data as a normalized infrastructure data view.	Phase 3	3.4.4.2
ARMR_SRS_240	Design Goal	It is desirable that the ARMOUR solution cross source correlation module be designed to fuse data new data with existing data (e.g., delta changes).	Phase 3	3.4.4.2
ARMR_SRS_241	Requirement	The ARMOUR solution cross source correlation module shall save the fused infrastructure events data as a normalized infrastructure events view.	Phase 3	3.4.4.2
ARMR_SRS_243	Requirement	The ARMOUR solution computational services shall include a reachability analyzer module.	Phase 3	3.4.4.2.1
ARMR_SRS_244	Requirement	The ARMOUR solution reachability analyzer module shall generate a reachability graph.	Phase 3	3.4.4.2.1
ARMR_SRS_245	Requirement	The ARMOUR solution reachability graph shall be a view of the infrastructure data that captures all unique host-to-host connectivity by port and protocol.	Phase 3	3.4.4.2.1

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_246	Requirement	The ARMOUR solution reachability graph shall display the following network connectivity information: a. Routing tables; b. Switch configurations; c. Firewall policies; d. Network intrusion prevention systems; e. Host intrusion prevention systems; and f. Device configuration.	Phase 3	3.4.4.2.1
ARMR_SRS_247	Design Goal	It is desirable that the ARMOUR solution reachability analyzer module be designed to amend a previously generated reachability graph with delta changes in network connectivity information (e.g.: without having to re-generate from all sources of information for each change) where this would improve overall performance.	Phase 3	3.4.4.2.1
ARMR_SRS_255	Requirement	The ARMOUR solution computational services component shall include an operations and infrastructure analyzer module.	Phase 3	3.4.4.2.2
ARMR_SRS_256	Requirement	The ARMOUR solution operations and infrastructure analyzer module shall have the ability to uniquely identify operations.	Phase 3	3.4.4.2.2
ARMR_SRS_257	Requirement	The ARMOUR solution operations shall be assigned an operation priority value based on user input.	Phase 3	3.4.4.2.2
ARMR_SRS_258	Requirement	The ARMOUR solution operation priority value shall allow relative comparison of operational priority between uniquely identified operations.	Phase 3	3.4.4.2.2
ARMR_SRS_259	Requirement	The ARMOUR solution operations and infrastructure analyzer module shall assign an operation dependence metric to operations, operational services, host services, possible host-to-host-on-protocol-and-port connections, and hosts derived from the operation priority values of the uniquely identified operations.	Phase 3	3.4.4.2.2
ARMR_SRS_260	Requirement	The ARMOUR solution operation dependence metric assigned to application services, host services, possible host-to-host-on-protocol-and-port connections and hosts shall be the result of an evaluation of operational dependency relationships.	Phase 3	3.4.4.2.2
ARMR_SRS_261	Requirement	The ARMOUR solution derived operation dependence metrics shall allow relative comparison of priority to the uniquely identified operations.	Phase 3	3.4.4.2.2
ARMR_SRS_262	Requirement	The ARMOUR solution operation dependence metric applied across all: a. Operations; b. Applications services; c. Host services; and d. Hosts, shall sum to 1.	Phase 3	3.4.4.2.2
ARMR_SRS_263	Design Goal	It is desirable that the ARMOUR solution allow for a configurable number of operation-priority scenarios.	Phase 3	3.4.4.2.2
ARMR_SRS_264	Design Goal	It is desirable that the ARMOUR solutions configurable number of operation-priority scenarios be limited only by storage and processing performance.	Phase 3	3.4.4.2.2
ARMR_SRS_265	Design Goal	It is desirable that the ARMOUR solution operations and infrastructure analyzer module generate a separate operation-priority metric for each operation-priority scenario.	Phase 3	3.4.4.2.2
ARMR_SRS_266	Requirement	The ARMOUR solution shall have a mechanism to enter attacker models.	Phase 3	3.4.3.4.4 3.4.4.2.3 3.4.4.2.4
ARMR_SRS_267	Requirement	The ARMOUR solution attacker model shall include the resources an attacker has access to.	Phase 3	3.4.4.2.3

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_268	Requirement	The ARMOUR solution attacker model shall include the attacker preference for attack characteristics, including: a. Attacks that do not leave log entries; b. Multi-hop attacks vs. non-multi-hop attacks; and c. Use of vulnerabilities with selectable attributes.	Phase 3	3.4.4.2.3 3.4.4.2.4
ARMR_SRS_269	Requirement	The ARMOUR solution shall limit access to add, modify or delete attacker models to the system/configuration manager user.	Phase 3	3.4.3
ARMR_SRS_270	Design Goal	It is desirable that the ARMOUR solution does not impose a limit on the number of attacker models retained (up to available storage space).	Phase 3	3.4.4.2.3
ARMR_SRS_271	Design Goal	It is desirable that the ARMOUR solution attacker models capture the likelihood of successfully exploiting vulnerabilities.	Phase 3	3.4.4.2.3
ARMR_SRS_273	Design Goal	It is desirable that the ARMOUR solution operations and infrastructure analyzer module generate a security posture metric for operations, application services, host services, and hosts.	Phase 3	3.4.4.2.2
ARMR_SRS_274	Design Goal	It is desirable that the ARMOUR solution operations and infrastructure analyzer module generate a separate security posture metric for each attacker model.	Phase 3	3.4.4.2.2
ARMR_SRS_275	Design Goal	It is desirable that the ARMOUR solution operations and infrastructure analyzer module provide an aggregation of the security posture metric for: a. Groups of operations; b. Groups of application services; c. Groups of host services; and d. Groups of hosts.	Phase 3	3.4.4.2.2
ARMR_SRS_276	Requirement	The ARMOUR solution security posture metric shall be a single, normalized value in the range 0 to 1 where 0 is completely insecure and 1 is perfectly secure (impossible to launch a successful attack).	Phase 3	3.4.4.2.2
ARMR_SRS_277	Requirement	The ARMOUR solution security posture metric shall include parameters taking into account: a. Host operational criticality; b. Likelihood of successful exploit by attackers; c. Vulnerabilities; d. Safeguards; and e. Network topology.	Phase 3	3.4.4.2.2
ARMR_SRS_278	Requirement	The ARMOUR solution security posture metric shall be described relative to the attack graph generated by the attack graph generator module.	Phase 3	3.4.4.2.2
ARMR_SRS_279	Requirement	The ARMOUR solution method/algorithm used to generate the security posture metric shall be clearly described in the Detailed Design Document.	Phase 3	3.4.4.2.2
ARMR_SRS_280	Requirement	The ARMOUR solution parameters used to generate the security posture metric shall be well defined.	Phase 3	3.4.4.2.2
ARMR_SRS_281	Design Goal	It is desirable that the ARMOUR solution operations and infrastructure analyzer module metrics be based on operation priority (e.g., not based on attacker intent).	Phase 3	3.4.1.4
ARMR_SRS_282	Requirement	The ARMOUR solution metrics generated shall enable comparison of security postures at each level including: a. Host system; b. Operations; c. Operational services; d. Networks; and e. Infrastructure.	Phase 3	3.4.1.4
ARMR_SRS_286	Design Goal	It is desirable that the ARMOUR solutions aggregate metrics generated include parameters derived from interactions and dependencies between host systems.	Phase 3	3.4.4.2.4
ARMR_SRS_287	Design Goal	It is desirable that the ARMOUR solution metrics generated include parameters derived from infrastructure redundancy.	Phase 3	3.4.4.2.4

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_288	Design Goal	It is desirable that the ARMOUR solution metrics generated include parameters derived from performance degradation.	Phase 3	3.4.4.2.4
ARMR_SRS_289	Design Goal	It is desirable that the ARMOUR solution metrics generated include parameters derived from local attack characteristics.	Phase 3	3.4.4.2.4
ARMR_SRS_290	Design Goal	It is desirable that the ARMOUR solution metrics generated include parameters derived from remote attack characteristics.	Phase 3	3.4.4.2.4
ARMR_SRS_291	Requirement	The ARMOUR solution metrics generated shall include parameters derived from properties specified in software vulnerability information.	Phase 3	3.4.4.2
ARMR_SRS_292	Design Goal	It is desirable that the ARMOUR solution metrics generated include parameters derived from the maturity of exploits.	Phase 3	3.4.1.3 3.4.4.2.4
ARMR_SRS_293	Requirement	The ARMOUR solution host security metric generated shall represent the exploitability (likelihood of success) of technical attacks (e.g., launching malicious code).	Phase 3	3.4.4.2.4
ARMR_SRS_294	Requirement	The ARMOUR solution host security metric generated shall represent the likelihood of success of non-technical attacks that would allow an attacker to execute arbitrary code on a host.	Phase 3	3.4.4.2.4
ARMR_SRS_296	Requirement	The ARMOUR solution computational services component shall include an attack graph generator module.	Phase 3	3.4.4.2.3
ARMR_SRS_297	Requirement	The ARMOUR solution attack graph generator module shall generate an attack graph.	Phase 3	3.4.4.2.3
ARMR_SRS_298	Requirement	The ARMOUR solution attack graph shall be: a. A directed forward hyper graph; or b. An AND/OR directed graph.	Phase 3	3.4.4.2.3
ARMR_SRS_299	Requirement	The ARMOUR solution attack graph shall contain both conjunctive and disjunctive dependencies.	Phase 3	3.4.4.2.3
ARMR_SRS_300	Requirement	The ARMOUR solution attack graph shall include network conditions including: a. Network connectivity; b. Host services running; and c. Presence of vulnerabilities.	Phase 3	3.4.4.2.3
ARMR_SRS_301	Requirement	The ARMOUR solution network connectivity information shall be derived from the reachability graph.	Phase 3	3.4.4.2.1
ARMR_SRS_302	Requirement	The ARMOUR solution host services running shall be derived from the abstracted infrastructure data.	Phase 3	3.4.1.1
ARMR_SRS_303	Requirement	The ARMOUR solution presence of vulnerabilities shall be derived from the abstracted infrastructure data.	Phase 3	3.4.1.1
ARMR_SRS_304	Requirement	The ARMOUR solution attack graph generator shall examine new vulnerabilities without requiring human adaptation of the input data for the particular attack graph generator.	Phase 3	3.1.3
ARMR_SRS_305	Requirement	The ARMOUR solution presence of exploits shall be derived from the security reference information.	Phase 3	3.1.3
ARMR_SRS_307	Requirement	The ARMOUR solution attack graph generator module shall have a proactive mode.	Phase 3	3.1.3
ARMR_SRS_308	Requirement	The ARMOUR solution while in proactive mode, shall include an attack graph representing the multi-stage attack paths between user-selected attack sources and user-selected or priority hosts.	Phase 3	3.4.4.2.3

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_309	Requirement	The ARMOUR solution high priority hosts shall be automatically identified as any host having an operation dependence metric at or above a priority level specified by the user.	Phase 3	3.1.3
ARMR_SRS_315	Requirement	The ARMOUR solution computational services component shall include an attack graph analyzer module.	Phase 3	3.4.4.2.4
ARMR_SRS_316	Requirement	The ARMOUR solution attack graph analyzer module shall have a proactive mode.	Phase 3	3.4.4.2.4
ARMR_SRS_317	Requirement	The ARMOUR solution proactive mode of the attack graph analyzer module shall identify attack assets, within the Exploit Dependency Graph - Proactive, generated by the attack graph generator module, critical to the success of an attacker (called Attack Assets - Proactive).	Phase 3	3.4.4.2.4
ARMR_SRS_319	Requirement	The ARMOUR solution Attack Assets - Proactive shall be described relative to the attack graph generated by the Proactive attack graph generator module.	Phase 3	3.4.4.2.4
ARMR_SRS_324	Requirement	The ARMOUR solution method/algorithm used to identify the attack assets shall be clearly described in the Detailed Design Document.	Phase 3	3.4.4.2.5
ARMR_SRS_325	Requirement	The ARMOUR solution method/algorithm used to identify the attack assets shall account for interactions and dependencies between host systems.	Phase 3	3.1.3
ARMR_SRS_326	Requirement	The ARMOUR solution method/algorithm used to identify the attack assets shall account for infrastructure redundancy.	Phase 3	3.1.3
ARMR_SRS_327	Design Goal	It is desirable that the ARMOUR solution method/algorithm used to identify the attack assets account for performance degradation.	Phase 3	3.1.3
ARMR_SRS_328	Requirement	The ARMOUR solution method/algorithm used to identify the attack assets shall account for local attack characteristics.	Phase 3	3.1.3
ARMR_SRS_329	Requirement	The ARMOUR solution method/algorithm used to identify the attack assets shall account for remote attack characteristics.	Phase 3	3.1.3
ARMR_SRS_330	Requirement	The ARMOUR solution method/algorithm used to identify the attack assets shall account for properties specified in software vulnerability information.	Phase 3	3.1.3
ARMR_SRS_331	Requirement	The ARMOUR solution method/algorithm used to identify the attack assets shall account for parameters derived from the maturity of exploits.	Phase 3	3.4.4.2.4
ARMR_SRS_332	Requirement	The ARMOUR solution method/algorithm used to identify the attack assets shall account for exploitability (likelihood of success) of technical attacks (e.g., malware).	Phase 3	3.4.4.2.4
ARMR_SRS_333	Requirement	The ARMOUR solution method/algorithm used to identify the attack assets shall account for the likelihood of success of non-technical attacks (e.g., social engineering).	Phase 3	3.4.4.2.4
ARMR_SRS_334	Requirement	The ARMOUR solution attack graph analyzer module shall generate an attack dependence metric for each attack asset.	Phase 3	3.4.4.2.4
ARMR_SRS_336	Requirement	The ARMOUR solution attack dependence metric shall indicate the degree of dependence the attacker (with capabilities according to an attacker model) places on attack assets.	Phase 3	3.4.4.2.4

Use or disclosure of this data is subject to the restriction on the title page of this document.

ID	Statement Type	Requirement Text	Primary Dev Phase	Relevant Section(s)
ARMR_SRS_337	Requirement	The ARMOUR solution attack dependence metric shall be described relative to the attack graph generated by the attack graph generator module.	Phase 3	3.4.4.2.4
ARMR_SRS_338	Requirement	The ARMOUR solution, in the absence of data on the likelihood of successful exploit for each vulnerability, the likelihood values shall be specified by the user.	Phase 3	3.4.4.2.1
ARMR_SRS_340	Requirement	The ARMOUR solution parameters used to generate the attack dependence metric shall be well defined.	Phase 3	3.4.4.2.4
ARMR_SRS_341	Requirement	The ARMOUR solution method/algorithm used to generate the attack dependence metric shall be clearly described in the Detailed Design Document.	Phase 3	N/A
ARMR_SRS_342	Design Goal	It is desirable that the ARMOUR solution attack graph analyzer module generate an attack dependence metric for each attack asset in a path to gain the use of each uniquely identified attack asset.	Phase 3	3.4.4.2.4
ARMR_SRS_450	Requirement	The ARMOUR solution display (graphical and tabular) shall include a method to display operations information.	Phase 3	3.4.3.4.2
ARMR_SRS_451	Requirement	The ARMOUR solution display (graphical and tabular) shall include a method to include operations criticality information against the operations information.	Phase 3	3.4.3.4.2
ARMR_SRS_452	Requirement	The ARMOUR solution display (graphical and tabular) shall include a method to display infrastructure information.	Phase 3	3.4.3.4.1
ARMR_SRS_453	Requirement	The ARMOUR solution display (graphical and tabular) shall include a method to display reachability information.	Phase 3	3.4.4.2.1
ARMR_SRS_454	Requirement	The ARMOUR solution display (graphical and tabular) shall include a method to include security status indicators against the infrastructure information.	Phase 3	3.4.3.4.3
ARMR_SRS_455	Requirement	The ARMOUR solution display (graphical and tabular) shall include methods for the user to select aspects of the infrastructure to designate as attacker sources.	Phase 3	3.4.3.4.4
ARMR_SRS_456	Requirement	The ARMOUR solution display (graphical and tabular) shall include methods for the user to select aspects of the infrastructure to designate as attacker goals.	Phase 3	3.4.3.4.4
ARMR_SRS_457	Requirement	The ARMOUR solution display (graphical and tabular) shall include methods for the user to display attack paths between designated attacker sources and goals.	Phase 3	3.4.3.4.4
ARMR_SRS_460	Requirement	The ARMOUR solution security status indicators shall include: a. Likely compromised host and operations; b. Vulnerable hosts and operations; c. New hosts; d. New operations; e. Attacker sources; f. Attacker goals; and g. Operational criticality.	Phase 3	3.4.3.4.2.1 3.4.3.4.2.2

Use or disclosure of this data is subject to the restriction on the title page of this document.