SAMSON Technology Demonstrator CWIX2014 Trial Report

Glen Henderson
Cord3 Innovation

Alan Clason, Daniel Charlebois, Bruce Carruthers and Darcy Simmelink
Cord3 Innovation

Prepared By:
Cord3 Innovation
464 Besserer Street
Ottawa, ON K1S 5N4

DRDC Project Number: 3775-2005-15bs (CST C4ISR 3)
PWGSC Contract Number: W87714-08FE01/Bel
Contract Technical Authority: Daniel Charlebois, Defence Scientist.

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

Contract Report
DRDC-RDDC-2016-C103
July 2014

IMPORTANT INFORMATIVE STATEMENTS:

Aviation Management Interoperability for Emergency Response and Recovery Project, # CSSP-2014-CP-2005, was supported by the Canadian Safety and Security Program which is led by Defence Research and Development Canada's Centre for Security Science (DRDC-CSS), in partnership with Emergency Management British Columbia (EMBC). The project was led by DRDC-CSS.

Canadian Safety and Security Program is a federally-funded program to strengthen Canada's ability to anticipate, prevent/mitigate, prepare for, respond to, and recover from natural disasters, serious accidents, crime and terrorism through the convergence of science and technology with policy, operations and intelligence.

Defence Research and Development Canada

SAMSON Technology Demonstrator CWIX2014 Trial Report

SD006

Bell Canada
160 Elgin Street
17th Floor
Ottawa, Ontario
K1S 5N4

Draft
July 2014

Confidentiality

This document is UNCLASSIFIED.

Disclaimer

The contents of this report do not constitute original work on behalf of the author. A number of sections of the report are comprised of material contributed by multiple authors, most notably members of the Bell Canada / Cord3 Innovation SAMSON team.

Authors

Bell Development Team	Role
Glen Henderson	Lead System Architect

Bell Q&A Team	Role
Alan Clason	Testing Specialist

Review

DRDC SAMSON Team	Role
Bruce Carruthers	Technical Advisor
Daniel Charlebois	Scientific Authority
Darcy Simmelink	Project Manager

Table of Contents

1.0 Introduction

This report documents the results from the demonstration of a new capability for data centric security architectures for the exchange of information between organizational domains. This capability, Cross Organization Information Exchange (COIE), was demonstrated at the Coalition Warfigher Interoperability Experiment (CWIX2014) that was hosted at the NATO facilities at Bydgoszcz, Poland with participation from various partner nations. This report provides a description of:

- The capability that was delivered for experimentation at CWIX2014;
- The hardware, software and network configuration under which the testing was conducted;
- A summary of the testing methodology;
- A summary of the observed results;
- A discussion of the impact of this experiment both in terms of the success of the capability from a technology perspective and its relevance to NATO partners; and
- An identification of future work for data centric security architectures that builds upon the successes of this experiment.

The intent of the COIE experiment was to gather evidence to prove the viability of a new mechanism to exchange information between organizational domains. The challenge presented in existing information architectures is that organizations typically select their own preferred security-labelling format. The lack of a common approach to defining inter-organization security labels and associating information assets with those labels significantly hinders the ability to exchange information in a controlled manner. In contrast, when a common labelling mechanism is in place for the exchange of data assets, information can be shared between organizations (and handled and processed) according to mutually agreed to memoranda of understanding (MOU). The goal of this experiment was to create, deploy and test an implementation of a cross organization information exchange solution that will support the use of MOUs between organizations through a common labelling standard for information in transit.

The COIE activities within the overall CWIX2014 experiment were conducted between June 2 and June 13, 2014. The architecture for the experiment allowed participating partner nations to send information assets from their respective organizational domains to a central collecting domain located at the NATO Poland facility. In this way, the experiment followed a “hub and spoke” model with participating partner nations as the generator of information assets and the central Poland facility as the recipient of those assets. Supporting evidence, in the form of log files, was collected at both the source and destination domains to show the transformation of security label information and to ensure that the integrity of the security metadata was maintained through each step of the testing process.

In summary, the programmatic and technical objectives for the COIE CWIX2014 experiment were:

1. To prove the viability of a cross organization information exchange mechanism where:

    a. Security metadata from assets sent from a source domain are used to create a transport security label;
    b. Information assets and their associated transport security labels are placed in a container for transmission between domains; and
    c. At the receiving domain, the security metadata on the transport container is validated against the security metadata on the original asset to ensure that it was not modified in transit.

2. To support a set of security labelling formats on the original information assets including:

    a. Property-based security attribute files (e.g. Titus Document Classification);
    b. XML Security Policy Information File (SPIF); and
    c. NCIA Security Label Attributes (SLAB) format.

3. To exchange information using unmodified client and server software while supporting the capability to encapsulate information assets into a transport container in transit.

4. To support COIE using email as the information exchange method.

5. To gather supporting information in the form of log files to demonstrate the effectiveness of the proposed solution.

In addition to being hosted by and evaluated by the Canadian contingent, this experiment included the participation of following nations and organizations:

- The United States of America
- The United Kingdom
- Germany
- Finland
- The Netherlands
- NATO / NCIA

The assistance of these partners in conducting the CWIX2014 experiment is greatly appreciated.

2.0 Technical Specification

For the CWIX2014 experiment, the technical requirement for the COIE solution was to:

1. Create separate information domains for each partner nation with each domain having:

    a. Unique user accounts to be used during the testing process;
    b. A windowing environment in which the user has access to an email application; and
    c. A mail server to which the user submits email messages for delivery to the target user at the target information domain.

2. Deploy Policy Enforcement Points (PEPs) where:

    a. Information assets being sent to a recipient were placed into a transport container for transit; and
    b. Information assets being received were extracted from their transport container and validated against the original asset security metadata.

3. Leverage the CWIX2014 networking architecture to ensure that messages were able to be transmitted to their destination.

These requirements were met through the adherence to the following architectural specification and information flow.

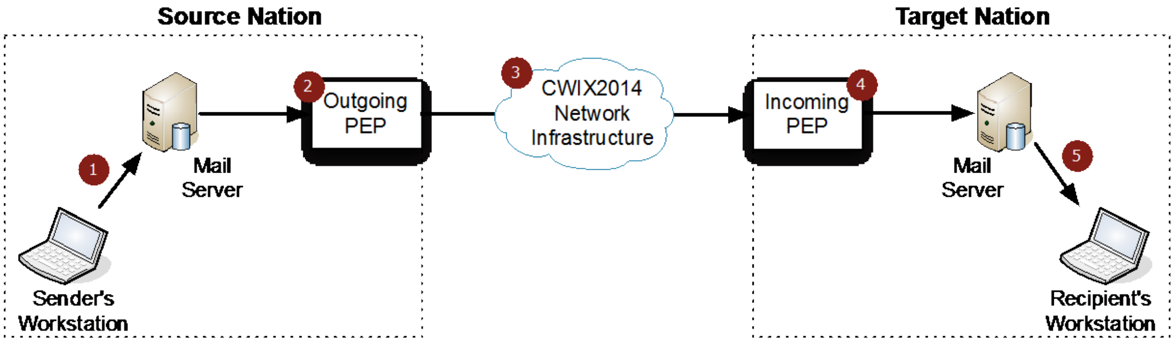

Figure 1: CWIX2014 Solution Architecture

1. The user at the sender’s workstation submits an email with a file attachment to the local mail server for delivery to the recipient at the target nation domain. The attached file contained a security label using one of the three supported label types.

2. The source nation's mail server redirects email traffic through the ***outgoing PEP*** service where the entire message is placed in a transport container for transit between domains. The security metadata in the security label on the transport container is taken from the security metadata on the information asset.

3. The message is relayed through the CWIX2014 network infrastructure to the target nation's information domain and is received at the ***incoming PEP*** service.

4. The incoming PEP extracts the original message from the transport container and compares the security metadata on the transport container's label with the security metadata on the information asset. Assuming the integrity of the message has been verified, the message is sent to the target mail server for delivery to the intended recipient.

5. The target recipient retrieves the message.

This information flow represents the architectural view of the COIE solution at CWIX2014.

The CWIX2014 testing methodology followed a *hub and spoke* model where there were many partners acting as source nations sending information assets to a single receiving target organization.

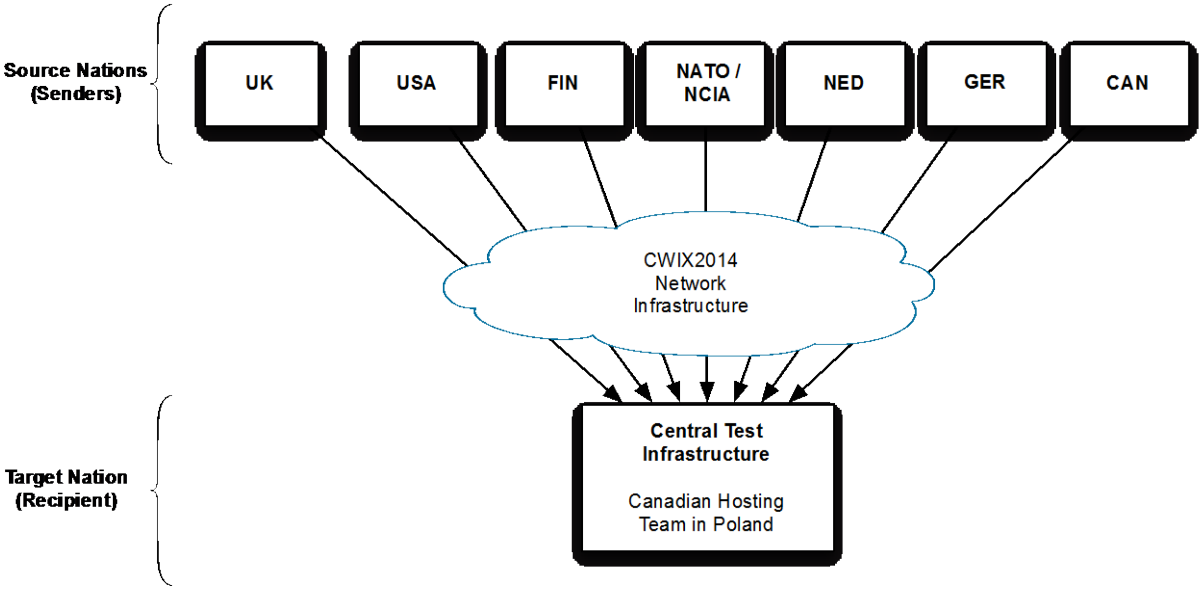

Figure 2: CWIX2014 Network Architecture

As a result, there were many outgoing PEPs in the overall architecture but only a single incoming PEP at the central test infrastructure. This single incoming PEP processed all messages delivered to the central test infrastructure hosted by the Canadian contingent at the NATO facilities in Poland.

2.1 Hardware Configuration

There were three deployment strategies for physical environments in the overall CWIX2014 test architecture.

2.1.1 *The Central Test Infrastructure*

This infrastructure was set up and maintained by the Canadian contingent in Poland. It consisted of one HP Proliant DL360 blade server and two Dell Studio 1747 laptops. The blade server hosted the virtualization platform within which the virtual machine for the exercise was hosted. One laptop was used to access and manage the virtualization software at the blade server. The second laptop was used to access the virtual workstation on which the testing procedures were executed and collect evidence in support of the exercise. The Central Test Infrastructure also used a network switch to connect the laptops and blade server to the CWIX2014 networking environment.

2.1.2 *The Canadian Source Infrastructure*

The Canadian contingent at the Canadian Warfare Centre (CWC) acted as one of the source nations. The Canadian Source Infrastructure executed the test procedures to demonstrate the COIE capability; that is, email messages were sent from this environment to the Central Test Infrastructure in Poland. The Canadian Source Infrastructure consisted of one HP Proliant DL360 blade server that hosted the virtualization platform within which the virtual machine for the exercise was hosted. The CWC provided a secure workstation that was used to access the virtual workstation on which the testing procedures were executed and evidence collected in support of the exercise.

2.1.3 *Other Nation Infrastructures*

All other partner nations provided their own hardware including: a virtualization platform to host the virtual machine for the exercise and a physical workstation that was used to access the virtual workstation on which the testing procedures were executed and collect evidence in support of the exercise.

2.2 Network Infrastructure

The connectivity between partner nations' infrastructures was the responsibility of the CWIX2014 technical support staff. The CWIX2014 support team provided network configuration details for each of the partner nations and ensured connectivity between infrastructures.

2.3 CWIX2014 Virtual Machine

For the CWIX2014 exercise, all environments used an instantiation of a VMWare ESXi 5.1 compatible virtual machine that had been specifically configured for the experiment. In this way, the virtual machine served as a template for all exercise deployments. The virtual machine was provided to partner nations (both as a DVD and via an FTP service) as a file that could be imported into their respective environments. This virtual machine was configured with the following software elements:

- Operating System: CentOS 6.5 (64-bit)
- Gnome windowing environment with the Thunderbird (version 24.5) email client
- TigerVNC (version 1.1) and XRDP (version 0.6) for remote sessions
- Postfix (version 2.6.6) and Dovecot (version 2.0.9) for email services

When imported into the target environment, local test team representatives were required to set the following information within the instantiated virtual machine:

- Network Configuration Details (provided by the CWIX2014 technical support team):
  - IP Address
  - Netmask
  - Gateway address
  - Fully qualified domain name
- Test Configuration Details (provided by the Canadian COIE test team)
  - IP address of the Central Test Environment Incoming PEP
  - Domain name of the Central Test Environment

The test configuration details were not known until the start of the exercise and could not be pre-configured in the virtual machine template.

Once deployed and configured, the following network services were enabled on each source nation virtual machine. Some services were bound to the local loopback interface whereas others were bound to the network interface.

Network Service	Loopback Interface port	Network interface port
Secure Shell (SSHD)		22/tcp
Domain Name Service	53/tcp 53/udp	
Virtual Network Computing (VNC)	5900/tcp	
Remote Desktop Protocol (XRDP)		3389/tcp
Postfix SMTP	25/tcp	
Dovecot POP3	110/tcp	
CWIX2014 Outgoing PEP		25/tcp

The Central Test Infrastructure had a similar network configuration. However, since the role of the Central Test Infrastructure was to receive email messages from partner nations its virtual machine was configured to use the CWIX2014 **Incoming PEP** (rather than the **Outgoing PEP**) on port 25 of the network interface. A detailed description of the PEP software services is provided in the following section.

2.4 PEP Software Services

In the CWIX2014 experiment, the PEP components consisted of two core elements:

1. An intercept component that accesses email messages in transit; and
2. A policy component that generates the transport container (or extracts the message from the container) from the intercepted message.

As defined previously, in both the source and target environments the PEP is hosted on port 25 and waits for email traffic to transit through that port. At the source environments, a mail exchange (MX) record ensures that traffic destined for the central test facility is first routed through the local Outgoing PEP. If the Outgoing PEP allows traffic to proceed to its destination, the email message is routed through to the Incoming PEP at the target environment. If the Incoming PEP allows traffic to proceed to its destination, the email message is sent to the target environment's mail server to await delivery. If at any time, an error is encountered, the message is bounced back to the originating mail server and a bounce message is sent to the sender.

2.4.1 *Intercept Component*

Both the Incoming and Outgoing PEPs used the same intercept technology. The intercept for all mail traffic was based on ProxSMTP (version 1.10); an open source email virus/spam filter. ProxSMTP intercepts email messages and makes them available to scanning and modification in transit. In this case, ProxSMTP was re-purposed to process the security metadata on the attachments on intercepted email traffic. ProxSMTP allows calls to be made to third party software solution to process email messages that have been intercepted. In this way, ProxSMTP was used without modification to intercept email traffic and call out to the CWIX2014 policy component for processing. For the CWIX2014 exercise, the policy component was a Python 2.7 script.

2.4.2 *Policy Component*

For the experiment, there was a policy component for processing messages as they leave a domain (outgoing) and a policy component for processing messages as they enter a domain (incoming). Specifically, partner nations were required to send email messages and the intercept used the Outgoing PEP configuration; conversely the Central Test Infrastructure used the Incoming PEP configuration. Once the ProxSMTP intercept has access to an email message, it called the PEP script to:

1. begin a log record of the transaction and
2. extract the attachment from the message.

Depending on whether the intercept is using the Outgoing or Incoming PEP, the processing of the message continues as follows:

For the **Outgoing PEP:**

1. Determine the type of security label on the message attachment.

2. Extract the security attributes from the security label, including:
    a. Policy ID;
    b. Classification;
    c. Releasability; and
    d. Caveats.

3. Create a new security label (transport security label) in XML SPIF format containing the 4 core security attributes.

4. Create a ZIP archive and place the original email message and transport security label into this archive. This archive is the transport container.

5. MIME Encode the transport container so that it can be attached to an email message.

6. Generate a new email message (as a file) based on the original message's header information and attach the encoded transport container.

7. Replace the original email message with the new message containing the transport container and return processing control to the intercept that will deliver the transport container to the target domain.

For the **Incoming PEP**:

1. Extract the security attributes from the transport container's security label, including:
    a. Policy ID;
    b. Classification;
    c. Releasability; and
    d. Caveats.

2. Extract the attachments from the original message that has been stored in the transport container.

3. Determine the type of security label on the original message attachment.

4. Extract the security attributes from the security label on the attachment in the original message and ensure that the two sets of security attributes are in accord (ensure that the security attributes on the transport container have not been modified).

5. Replace the email message with the transport container with the original email message and return processing control to the intercept that will deliver the original message target domain’s mail server.

For both PEPs, once processing is complete any working files are removed and the log file is closed. This log file provided a complete record of the actions performed by the PEPs and provides evidence in support of the testing activities.

The PEPs were subject to the following assumptions when processing email messages:

- If an email message had no attachment, the message is allowed through the PEP without modification;
- If an email message had multiple attachments, the first attachment defined the security attributes for the transport container; and
- The transport security label used the XML SPIF format.

2.5 Security Label Formats

For all information types and label formats, the PEPs assumed that:

1. The original information asset is a ZIP archive; significantly, Microsoft Office uses the ZIP archive format natively for its file format representation.

2. Security labels are in a directory called *docProps* and stored in a file called *custom.xml.* This is a location used by Microsoft to store custom XML properties.

Each of the following security label formats is supported by the CWIX2014 COIE solution when stored at the *custom.xml* location.

Property-Based Security Label

This format is used by the Titus Document Classification product.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<Properties>
<property name="CWIXPOLICYID"><vt:lpwstr>CAN 1.2.3.4</vt:lpwstr></property>
<property name="CWIXCLASSIFICATION"><vt:lpwstr>NATO SECRET</vt:lpwstr></property>
<property name="CWIXRELEASABILITY"><vt:lpwstr>FIN/USA</vt:lpwstr></property>
<property name="CWIXCAVEATS"><vt:lpwstr>SPECIAL_OPS</vt:lpwstr>
</property>
</Properties>
```

Security Policy Information File (XML-SPIF)

```
<ns0:spif xmlns:ns0='http://www.xmlspif.org/spif'>
<ns0:securityPolicyId id='N/A' name='CAN 1.2.3.4'/>
<ns0:securityClassifications><ns0:securityClassification name='NATO SECRET' />
</ns0:securityClassifications>
<ns0:securityCategoryTagSets>
<ns0:securityCategoryTagSet name='Release Categories'>
<ns0:securityCategoryTag name='Releasable To'>
<ns0:tagCategory name='FIN/USA'><ns0:markingData><ns0:code />
</ns0:markingData></ns0:tagCategory></ns0:securityCategoryTag>
</ns0:securityCategoryTagSet>
<ns0:securityCategoryTagSet name='Discretionary Handling Categories'>
<ns0:securityCategoryTag name='Administrative Markings'>
<ns0:tagCategory name='SPECIAL_OPS' />
</ns0:securityCategoryTag>
</ns0:securityCategoryTagSet>
</ns0:securityCategoryTagSets>
</ns0:spif>
```

NCIA / SLAB Security Label

```
<?xml version="1.0" encoding="UTF-8"?>
<slab:ConfidentialityLabel Id="String" ReviewDateTime="2006-12-17T10:30:47.0Z"
xsi:schemaLocation="urn:int:nato:ia:xmlsecuritylabel:xmlconfidentialitylabel:draft
..\schemas\ConfidentialityLabel-v1d12.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:slab="urn:int:nato:ia:xmlsecuritylabel:xmlconfidentialitylabel:draft">
 <slab:ConfidentialityInformation>
  <slab:PolicyIdentifier URI="oid:1.3.26....">CAN 1.2.3.4</slab:PolicyIdentifier>
  <slab:Classification URI="oid:1.3.26....">NATO SECRET</slab:Classification>
  <slab:Category Type="PERMISSIVE" URI="oid:1.3.26....">
   <slab:GenericValue>Releasable to FIN/USA</slab:GenericValue>
  </slab:Category>
  <slab:Category Type="INFORMATIVE" TagName="Administrative Markings">
    <slab:GenericValue>SPECIAL_OPS</slab:GenericValue>
  </slab:Category>
 </slab:ConfidentialityInformation>
 <slab:CreationDateTime> 2013-08-29T16:15:00 </slab:CreationDateTime>
</slab:ConfidentialityLabel>
```

3.0 Test Scenarios

The test methodology, test cases and testing results are presented in detail in section 2:*Testing* and section 4:*Test Results* of the *CWIX2014 Test Report*. A brief discussion of the test cases is provided below.

Each partner was invited to participate in two rounds of testing: structured testing and ad-hoc testing. The structured testing required partners to submit a set of pre-defined labelled documents via email. In each test case, the sender would attach a specific document, including:

- A set of test cases with a Microsoft Office document attachment with a file with a property-based security label.
- A set of test cases with a Microsoft Office document attachment with a file with an XML SPIF security label.
- A set of test cases with a Microsoft Office document attachment with a file with an NCIA/SLAB security label.
- A test case with an attachment with an invalid security label.
- A test case with an attached JPG file with a security label.
- A test case with an attached PDF file with a security label.

During the ad-hoc testing, the partners were invited to take an existing file and alter the security label security attributes. Partners were provided with a utility that allowed them to see the security label on a file, change attributes within that label or replace the security label with a label using a different format.

Upon completion of the test cycle, the results and supporting log file information was provided to the CWIX2014 COIE test team for interpretation and analysis. This information formed the basis for the information presented in the *CWIX Test Report* section 4:*Test Results* and *Appendix A: Result Sheets*.

4.0 Results

While the detailed test results are documented in the *CWIX2014 Test Report* section 4 (*Test Results*), the following observations can be made about the results from the experiment.

1. All solution elements performed as expected.
2. The approach to hardware and software deployment, configuration and operation allowed the experiment to take place without any technical difficulties.
3. Core testing was completed with all participating partner nations; ad-hoc testing was completed with all but USA participation. Due to time constraints at the partner nation, the USA contingent was not able to schedule ad-hoc testing.
4. All testing was successful: no failures were encountered during the test cycle procedures.
5. The solution was successfully demonstrated in operation to partners and VIP representatives.

4.1 Additional Observations

It was noted during testing that naming conflicts used when creating the transport container could lead to unexpected results. This issue was discovered during ancillary testing by the test team and is noted below.

> MS Office documents are ZIP archives with security attributes located in files within that ZIP structure. Therefore a DOCX file is, in actuality, a ZIP file. The approach for labeling other file types is to follow that same approach: create a ZIP archive; place the asset in the archive with a file that holds that asset's security attributes. A minor issue arose when naming the newly create ZIP archive. To keep things simple, the archive was given the same name as the original asset. This was done so that the asset would look the same to the user in Explorer regardless as to whether it was inside a ZIP archive. A file called img27.jpg, therefore would be placed inside a ZIP archive called img27.jpg and the user would be able to identify the file as containing a JPG image.
>
> This caused two related issues:
>
> 1. When creating the ZIP archive, the underlying ZIP module code kept trying to add itself to the archive (a runaway race condition…mostly due to code limitations); and
> 2. When extracting the files from the ZIP archive, the extracted asset overwrote the original ZIP archive since they both had the same name and that resulted in a corrupted ZIP.

These are minor issues that were quickly resolved but led to some deeper analysis of the problem space. It is proposed that in future standards, it be stated that the asset should be given a random file name while stored in the archive and that the original file name be either taken from the ZIP archive or, even better, from a metadata file that gets stored in the archive with the asset. If the file is renamed while it is in the archive, the original file name stored in a metadata file and the file and metadata file encrypted, the asset's name will be obfuscated and will prevent information leakage through the name of the file.

4.2 Impact and Future Considerations

The CWIX2014 experiment has provided evidence of the following.

1. The concept of using a transport container to encapsulate information assets while in transit between partner nation infrastructures is viable.

2. It is possible to allow partners to use separate security label formats so long as:

    a. Core security attributes can be extracted from the label format; and
    b. These attributes can be evaluated in the context of a MOU between nations so that received information can be protected appropriately for the sensitivity of the information.

The CWIX2014 experiment provided partner nations with a view into the work being done by the DND DCSS program for data centric information exchange between domains and infrastructures.

Participation in CWIX2014 has successfully demonstrated the feasibility of the transport container concept for exchange of information assets between domains. The scope of this demonstration, however, was limited to container syntax, transformation and encapsulation. Solution elements that were not in scope for this demonstration must be further investigated, reviewed and studied in order to fully develop this approach for information sharing between organizations, including:

- The binding mechanism to tie information security metadata to information assets;

- The trust model for ensuring that the integrity of information, and the reliability of the associated security metadata, is maintained throughout the information exchange process;

- The underlying infrastructure to encapsulate, bind and transfer information between domains in a manner that leverages the trust and capabilities of existing IT and security services;

- The service oriented architecture needed to both leverage and supply required information security capabilities to support the transport container model; and

- The policy models for bridging security metadata between organizational security domains.

It is recommended that Canada use CWIX2015 to further develop these areas of interest and to allow the results of that investigation to shape a position for Canada with respect to confidentiality labelling and cross domain systems.