Cross Domain Rule Set Verification Tools and Process Improvements

Charles McElveen, Martin Liedy	**Kelly Djahandari**	**Lance Call**
Cobham Analytic Solutions	**Northrop Grumman**	**AFRL**
Orlando, Florida	**Orlando, Florida**	**Mesa, Arizona**
Charles.McElveen@cobham.com, Martin.Liedy@cobham.com	**Kelly.Djahandari@ngc.com**	**Lance.Call@mesa.afmc.af.mil**

ABSTRACT

In the Team Mission Training Environment, the primary training goal is to promote pilot and crew station readiness. One of the key ingredients in achieving this goal is a well designed Cross Domain Rule Set that minimizes limitations on data flowing between security domains; therefore, allowing for more effective and realistic training over the Combat Air Force (CAF) Distributed Mission Operations Network (DMON). The key to developing a well designed Cross Domain Rule Set is verification of the simulation traffic processed by the Cross Domain Solution. Verification of a Rule Set is a complex and time consuming process. Complicating factors include the volume of simulation traffic and the intricacies of the Rule Set. In the past, the process of analyzing simulation traffic processed by the Cross Domain Solution has been a manually intensive process that required many hours of analysis that introduced possible margins for error. A manual process is not an acceptable method since this analysis is performed many times for an applied Rule Set to ensure that the Rule Set is functioning as required by the Designated Approving Authority.

This paper reports on solutions and tools that address this verification problem and the process improvements that have been achieved from lessons learned. This paper describes the Event Analysis Tool (EAT), an automated tool used to streamline the simulation traffic verification process, and the Air Force Research Laboratory developed visualization tool that allows users to verify that data is being filtered according to the applied Cross Domain Rule Set. These tools greatly speed up the Rule Set verification process and help to identify residual inference risks.

ABOUT THE AUTHORS

Charles McElveen, CISSP, ISSEP, is a Senior Security Engineer at Cobham Analytic Solutions with over 27 years of progressive experience supporting a variety of military and Department of Defense (DoD) programs. He is currently supporting the Distributed Mission Operations Network (DMON) Cross Domain Solution Services tasking. He received his Bachelor's Degree in Computer Science from the University of Southern Mississippi and a Master's Degree in Management/Management Information Systems from Florida Institute of Technology.

Kelly Djahandari, CISSP, is a Security Software Engineer at Northrop Grumman Information Systems and is leading a Cross Domain Solution Research and Development task order under the Distributed Mission Operations Mission Training program. Her information assurance experience includes more than 16 years of software engineering in network security research and cross domain solutions. She received a Bachelor's Degree from George Mason University and a Master's Degree from the University of Virginia.

Martin Liedy, CISSP, is a Senior Principal Engineer at Cobham Analytic Solutions with over 25 years of experience supporting a large range of military and DoD programs. He is currently supporting the DMON Cross Domain Solution Services tasking. He received his Bachelor's Degree in Computer Science and a Master's Degree in Management of Information Systems, both from University of Maryland.

Lance Call is a Software Engineer Principal at L3 Communications AFRL Mesa. He holds a BS degree in Electronics Engineering Technology from Brigham Young University. He has over 20 years of experience supporting DOD simulation and training systems. He is the AFRL's F-22 CDS testbed task lead and is the primary software developer of the DISGUI visualization tool.

Cross Domain Rule Set Verification Tools and Process Improvements

Charles McElveen, Martin Liedy
Cobham Analytic Solutions
Orlando, Florida
Charles.McElveen@cobham.com,
Martin.Liedy@cobham.com

Kelly Djahandari
Northrop Grumman
Orlando, Florida
Kelly.Djahandari@ngc.com

Lance Call
AFRL
Mesa, Arizona
Lance.Call@mesa.afmc.af.mil

INTRODUCTION

The Combat Air Force (CAF) Distributed Mission Operations (DMO) Network (DMON) is a key component of the team mission training environment and critical in achieving the primary goal of promoting pilot and crew station readiness. The DMON provides a network infrastructure for warfighters located in geographically separate locations to perform virtual-constructive team training and mission rehearsals. Though currently most distributed training events are held at a single security level, cross domain training events are becoming increasingly common as more cross domain solutions achieve approvals to operate. Cross domain training events provide a more realistic level of team training. With more simulators being added to the DMON and the CAF DMO vision to train like they fight, the need for cross domain solutions deployed on the DMON to allow routine, recurring training between different security domains continues to grow.

A cross domain solution allows team training between two or more Mission Training Centers (MTCs) operating in different security domains, a high security domain and a low security domain. The goal is to allow the MTCs to train together using their capabilities while imposing minimal restrictions on any MTC in order to provide maximum training effectiveness. The core functionality of the cross domain solution is provided by the Rule Set. The Cross Domain Rule Set protects the sensitive high security domain data by permitting only authorized data to traverse to the low security domain. A Cross Domain Rule Set that is too permissive compromises the security of the high security domain while a Rule Set that is too restrictive diminishes training effectiveness.

In early 2009, the first cross domain capability for daily team training on the DMON became a reality. This capability is provided by the DMON Cross Domain Solution (DCDS). The development of a DCDS Rule Set is an involved process requiring inputs from a diverse group of individuals ranging from data owners to end users, verification and validation by the

Rule Set developers, and approval by the Designated Approving Authority.

This paper focuses on the design, development, and verification of the Rule Set executed by the DCDS. This paper also addresses process improvements related to the verification of the Rule Set.

DMON Cross Domain Solution (DCDS)

The DCDS allows the interconnection of MTCs operating in different security domains. The DCDS is a Protection Level 3 (PL-3 as described in JAFAN 6/3) system consisting of a bi-directional Controlled Interface located at the high side security domain MTC site and a Management System located at the DMON Operations Center. The core functionality of the DCDS Controlled Interface is provided by the Rule Set which processes Distributed Interactive Simulation (DIS) Protocol Data Units (PDUs). The Rule Set ensures unauthorized program data on the high side security domain is not released to the low side security domain. The DCDS Controlled Interface will either block, guise and pass, or pass unaltered DIS PDUs from the high security domain to the low security domain in accordance with the deployed Rule Set. The DCDS Controlled Interface can also filter low-to-high DIS PDU traffic.

The DCDS Controlled Interface is managed via a Management Station which provides the user interface to the remote components. The Management Station is used to deploy the Rule Set, configure the components, archive and retrieve audit data and transaction logs, and to start/stop the components for daily operations.

CROSS DOMAIN RULE SET DEVELOPMENT

Unclassified and classified, or program specific, rules are developed for the DCDS Controlled Interface. Unclassified rules are primarily used to ensure that the controlled interface is configured per its design specification. Once this configuration is confirmed, the focus turns to the classified rule definition, development, implementation and testing.

Development of a classified Cross Domain Rule Set is a multiple phase process including conduct of rules working group meetings, analysis of program specific classification guides, input and analysis of subject matter experts, development of an English Language Rule Plan, development of a Rule Implementation Report, coding of the rule, and testing of the Rule Set.

The first step in developing a Rule Set is an initial rules working group meeting. The rules working group meeting kicks off the process of developing a Rule Set for a given security domain. The goals of the initial Rule Set working group are to identify and obtain security classification guides from the data owners and to identify subject matter experts and stake holders for future working group meetings. The security classification guides identify program information in the high side security domain that must be protected. Attendees at a Rule Set working group meeting include MTC simulator developers, Rule Set code developers, platform subject matter experts, and government representatives.

The second step in the process is a detailed analysis of the security classification guides to identify information for a given security domain that must be protected. Table 1 below contains a summary of the typical types of information that can be found in the security classification guide. All information in the table below is notional since any real data would be classified. Included in the Table 1 is the category of sensitive information to be protected, its classification, clarification remarks and a sensitivity indicator. The sensitivity indicator is used for traceability purposes and is traced to the Rule Implementation Plan.

Sensitive Capability	Classification	Clarifying Remarks	Sensitivity Indicator
Fuel Consumption	Not Unclassified (NUC) to NUC-Special Access Required (SAR)	Full loiter capabilities are Special Access Required	FXMT.001
Total Flight Time	Not Unclassified (NUC) to NUC-Special Access Required (SAR)	Total flying time capability are Special Access Required	FXMT.002

Table 1 Example Security Classification Guide

Information gathered during the initial security working group meeting is critical in focusing efforts as part of this analysis. In some cases, one or more security classification guidelines may be required to fully understand the capabilities, performance factors, or features of a specific system on an Air Force MTC simulator that must be protected. The goal of this phase

of the process is to generate an English Language Rule Plan.

The English Language Rule Plan is a high level non-technical document that ultimately maps sensitive information from the security classification guide(s) to either a technical or non-technical rule. There are several intermediate steps in the process. First, each piece of sensitive information identified in the security classification guides is mapped to an indicator. An indicator is an event, object, or action which could reveal sensitive information relative to a capability, performance factor, or feature in the high side MTC. Once all items in the security classification guides have been mapped to indicators, the indicators are mapped to protection methods.

Protection methods are either technical or non-technical in nature. Technical methods are implemented in the DCDS and act on the DIS data stream automatically. Non technical methods are limitations placed on human operators such as being told not to make certain radio calls. Technical protection methods are mapped to one or more technical rules and non-technical protection methods map to one or more non-technical rules. Table 2 below is a sample notional table for technical protection methods contained in an English Language Rule Plan. Note the traceability of sensitivity indicators between the security classification guide and the English Language Rule Plan. The description provided in Table 2 below provides more fidelity concerning the program information that must be protected that can then be translated into a technical rule.

Protection Indicator	Description	Sensitivity Indicator	Technical Rule Mapping
PMTFX.001	Ensure time on station (loiter) does not reveal SAR information	FXMT.001 FXMT.002	ELTRFX.001 ELTRFX.002
PMTFX.002	Ensure actual flight times are protected	FXMT.001 FXMT.002	ELTRFX.001
PMTFX.003	Prevent MTC Instructor setup actions do not reveal SAR information	FXMT.001 FXMT.002	ELTRFX.003

Table 2 Example Technical Protection Methods

Technical rules are rules that will be coded into the DCDS Rule Set. Technical rules are developed to either block, guise, or allow a specific parameter associated with a capability, performance factor, or feature to pass unaltered from the high side security domain to the low side security domain. Technical rules for processing low side security domain DIS

PDUs bound for the high side security domain can also be developed. Non-technical rules are instructions given to pilots for a given security domain indicating specific actions that the pilot must or must not take. For example, a pilot may be instructed to not use certain systems or features of a system on a particular simulator. The English Language Rules Plan documents a number of technical or non-technical rules that address how the capability, performance factor, or feature will be protected. Table 3 below contains a notional example of a Technical English Language Rule. The descriptions contained in Table 3 are used to develop the pseudo code which is then used to code the Rule Set.

Rule	Description	Protection Indicator
ELTRFX.001	Block all DIS PDUs that contain data related to fuel consumption	PMTFX.001 PMTFX.002
ELTRFX.002	Block all DIS PDUs that reveal takeoff and landing of the FX	PMTFX.001 PMTFX.002
ELTRFX.003	Block Instructor generated resupply DIS PDUs	PMTFX.003

Table 3 Example English Language Technical Rules

Once the Rule Implementation Plan has been drafted, it is distributed to stake holders for review and analysis. Stake holders include data owners, simulator providers, subject matter experts, and DCDS Controlled Interface developers. Upon completion of this review a number of additional Rule Set working group meetings are held to review the Rule Implementation Plan and obtain concurrence from all stake holders before the Rule Implementation Report is developed. Depending on the complexity of the rules, a number of rules working group meetings may be required before the English Language Rule Plan is solidified and approved.

After concurrence of the English Language Rule Plan is obtained, the Rule Implementation Report is developed. The Rule Implementation Report builds on the Rule Implementation Plan documenting the implementation specifics that will be used to code the rule. For example, if one of the rules is to block a Fire PDU, then the rule implementation report will include pseudo code similar to the following:

```
If PDU_type = Fire
Then
        block Fire PDU;
```

If an English Language Rule requires modification of Fire PDUs from platform "A" to indicate platform "B," the implementation report will include pseudo code similar to the following:

```
If ((PDU type = Fire) and
(firing entity ID  = platform "A"))
Then
        Firing Entity ID = platform "B"
        Pass Fire PDU;
Else
        Continue processing Fire PDU;
```

If a rule in the English Language Rule Plan specifies to pass all Data PDUs, the implementation report will include pseudo code similar to the following:

```
If PDU type = Data
Then
        Pass Data PDU;
```

In addition to the pseudo code, the Rule Implementation Report also contains detailed non-technical rules and simulator vendor certifications. Certifications from the simulator vendor are required to proceed to the actual coding of the Rule Set. The certifications provide detailed information on how features are modeled in the simulator. For example, if we refer to the example given for modifying Fire PDUs from platform "A" to indicate platform "B", the simulator must certify how platform "A" and platform "B" are modeled in the simulator in order to ensure the DCDS Rule Set places the correct information in the Firing Entity ID field.

Once the Rule Implementation Report has been completed, the DCDS Rule Set is then coded. The rules coding process involves translating the technical rule and simulator vendor certification information contained in the Rule Implementation Report into rule code for the DCDS architecture. The coded rules are tested to ensure they are performing in accordance with the English Language Rules Plan. A variety of functional tests and analysis tools assist with validating that the coded rules protect the sensitive data. Depending on the complexity of the rules, the rules coding process can be iterative including refinements to the Rules Implementation Report, the coded rules themselves and the associated test.

Effective Training and Inference

Effective team training and security of program sensitive data are the two main factors when deciding if a rule should block, guise, or pass a DIS PDU. The Rule Set working group carefully scrutinizes the Rule Set to ensure it is neither too permissive nor too restrictive. Rule Set Working Groups include low side participants to confirm effective low side training in a cross domain training event.

In addition to ensuring that a Cross Domain Rule Set does not allow program information to pass from a high security domain to a low security domain while providing effective training, the Rule Set must also ensure that program information cannot be inferred from the information that is passed. Gaining information in this manner is referred to as inference. Inference causes a unique set of challenges for a cross domain system. One of these challenges is obtaining a balance between protecting program information and providing effective training to the low security domain participants. One method employed to minimize the potential for inference is to limit the types of DIS PDUs allowed to pass from the high security domain to the low security domain. Limiting the types of DIS PDUs allowed to pass is accomplished in the DCDS Rule Set by implementing an allow list. The DIS PDU types on the allow list consist only of those PDUs necessary for effective training. PDU types that contain information that is platform specific and are utilized to stimulate specific functions of the simulator of the high side MTC are not included on the allow list. A typical allow list consists of approximately ten PDU types that provide information about the battlespace and the high side entities, namely Acknowledge, Entity State, Fire, Detonate, Identification Friend or Foe (IFF), Designator, Electronic Emission, Signal, Receiver, and Transmitter. The possibility of inference is reduced but not eliminated by only passing PDU types on the allow list. PDUs on the allow list, while permitted to pass, may be modified by the Rule Set prior to passing to the low side. Technical rules that modify PDUs on the allow list are often designed to prevent inference.

PROCESS IMPROVEMENTS

Verifying the actions of a Rule Set handling millions of simulation data packets can be an overwhelming task. Process improvements to reduce the time and increase the accuracy of analysis were needed. Tools to run on the high and low side of the DCDS Controlled Interface were developed for analyzing PDU packets and for visualizing the battlespace. These tools are the Event Analysis Tool (EAT) and the Distributed Interactive Simulation Graphical User Interface (DISGUI), shown in Figure 1. In Figure 1 the high and low side security domains are displayed by color. In this case blue is the high side security domain and red is the low side security domain. In practice two tools are used when testing a rule, one connected to the high side network, the second connected to the low side network.
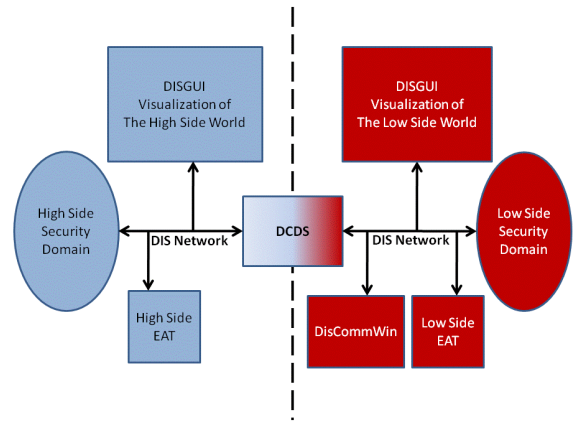

Figure 1 DCDS and Rule Set Verification Tools Conceptual Architecture

EAT provides a number by number comparison of PDUs based on the deployed rule. For example, if all comments PDUs are to be blocked, the EAT output and controlled interface output will both match. DISGUI provides more of a visual representation of the data flowing across the controlled interface and gives the rules developer a visual representation of how the rules are affecting the traffic flow between security domains.

In addition to EAT and DISGUI, a tool that allowed the verification and validation team the ability to hear the encoded audio carried in DIS Signal PDUs would be useful in the low side security domain. DisCommWin shown in Figure 1 was utilized to provide this functionality enabling the evaluators to hear the same information the low side security domain participants would hear during an operational event.

Event Analysis Tool

Event Analysis Tool (EAT) is an automated PDU packet analysis tool used for post event log examination. EAT is used to validate that the DCDS Controlled Interface is processing PDUs as defined by the Rule Set. The EAT's PDU recording and counting features assist the post event auditor with the precise post event log analysis.

Criteria for sorting and counting PDUs, which mimic the Rule Set used in the training event, are defined in the EAT Configuration File. EAT is run on each side of the Controlled Interface, high side and low side. The high side EAT recording predicts the PDU counts that the DCDS Controlled Interface will report while the low side EAT recording reports the PDUs actually received from the DCDS Controlled Interface. After an event, EAT counts of each PDU group based on the

criteria can be quickly compared to the DCDS Controlled Interface PDU counts.

The high side EAT data consists of the number and type of PDU packets received from the low side and the prediction of the PDU packets expected to be sent or dropped by the DCDS Controlled Interface. The EAT PDU packet numbers should match the DCDS Controlled Interface high-to-low and low-to-high statistics. The passed, blocked, and guised packet types should be consistent with the Rule Set implementation.

The EAT significantly improves post event log file analysis. Before EAT, the log file analysis activity was a lengthy and manually intensive exercise. This exercise consisted of separating log files by originating Internet Protocol (IP) address, importing into Microsoft Office Excel, using Excel filters to determine the count of each PDU type, and tallying the counts and comparing them to the Controlled Interface logs identifying Rule Set variations. A 90 minute training event may easily generate log files in excess of 4 Gigabytes, so data would require separating into multiple sheets due to Excel 2003's 65K or Excel 2007's 1 million row limit. Counts for any PDU being guised must be manually reviewed to ensure proper Rule Set functioning. This manual method takes many hours and can introduce human error.

Providing post event log file analysis increased the assurance to the DCDS Designated Approving Authority that the DCDS functioned as designed. The DCDS team provides post event analysis reports developed using EAT to the Designated Approving Authority.

Table 4 below provides an example of the type of data that is collected and reported by EAT during a cross domain test or training event. BL is blocked data, PU is passed unaltered data, and PA is passed altered data. For example, if a total of nine Fire PDUs were sent from the high side security domain bound for the low side security domain, the results for the high side EAT recording, the Controlled Interface, and the low side EAT would be as shown in Table 5 assuming the rule included code to block, pass unaltered, and pass altered certain Fire PDUs. All other PDU data contained in Table 4 would be completed in a real world event where additional rules and data would be included. For this example all other PDU data has not been completed.

PDU Type	High Side EAT			CI			Low Side EAT		
	BL	PU	PA	BL	PU	PA	BL	PU	PA
Acknowledge	–	–	–	–	–	–	–	–	–
Collision	–	–	–	–	–	–	–	–	–
Comment	–	–	–	–	–	–	–	–	–
Data	–	–	–	–	–	–	–	–	–
Data Query	–	–	–	–	–	–	–	–	–
Designator	–	–	–	–	–	–	–	–	–
Detonation	–	–	–	–	–	–	–	–	–
Electromagnetic Emission	–	–	–	–	–	–	–	–	–
Entity State	–	–	–	–	–	–	–	–	–
Fire	5	3	1	5	3	1	0	4	0
Receiver	–	–	–	–	–	–	–	–	–
Signal	–	–	–	–	–	–	–	–	–
Transmitter	–	–	–	–	–	–	–	–	–

Table 4 Example EAT Statistics

Even though it is obvious that fire PDUs were blocked, what was blocked and the effect it had on the low side security domain are not obvious. A visual representation could clearly show what was fired and the effects that were not realized on the low side security domain based on the fire PDU being blocked.

Visualization Tool – DISGUI

The Distributed Interactive Simulation Graphical User Interface (DISGUI) displays graphically as much as possible of the data in the DIS data stream so that if any part of the DIS data stream is modified it will be visible graphically in an intuitive manner and can therefore be identified easily. Comparison between the high side security domain and the low side security domain should reveal what data has been blocked, guised or passed.

The DISGUI connected to the high side DIS network is used to display the information that exists on the high side network. The DISGUI connected to the low side DIS network is used to display the information that exists on the low side network. By observing both DISGUI monitors, the differences between the high side world and low side world may be observed simultaneously in real-time. Experience has shown this to be a remarkably powerful approach that can highlight unexpected consequences of rules. Note that this approach requires that the DISGUI monitors be placed physically close to each other to allow

simultaneous viewing of the high and low side DISGUI applications.

Figure 2 shows side by side DISGUI monitors of the high and low side with a rule to block F-15s. The DISGUI allows the user to quickly visualize the effects of rules in real-time, within the context of the other players in the simulation. This allows the rules to be quickly evaluated and fine tuned. It also lends itself to showing unintended consequences of rules that need to be addressed.
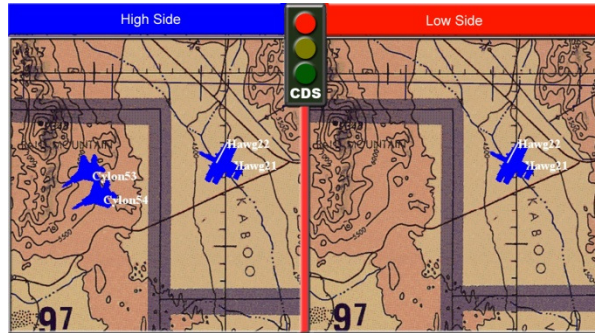

Figure 2 DISGUI Side-By-Side Display

DISGUI has the ability to display player icons and symbology on a map and map zooming functions as shown in Figure 3. There are a number of different symbols included that help the observer determine what is occurring at any given moment. Figure 4 shows examples of adding symbols to a base entity, in this example the F-16, to show avionics status. The DISGUI provides some data textually as labels to provide full details such as radar system types, radar system modes, and emitter information that may be desirable to look at, that do not lend themselves to a graphical representation.

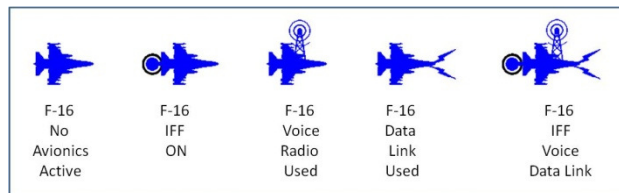**Figure 3 DISGUI Symbol Legend**

Figure 4 Example F-16s With Symbols

The three major unique features of DISGUI are:
- It is designed to help visualize engineering level details of the DIS data stream rather than for operational use.

- It can be configured to be controlled remotely from another DISGUI, which allows one user to manipulate both low side and high side DISGUIs and easily display the same view.
- Extensive filtering and labeling capabilities to allow the user to quickly focus on a specific aspect of the DIS data stream that is being modified or blocked by the DCDS.

The DISGUI currently supports the PDU types shown in Table 5.

PDU Type	Comments
Entity State	Additional symbology added to entity for avionics, marking, type, ID, altitude labels.
Fire	Pairing lines based on provided range and or estimated using shooter and target entity state data.
Detonate	Multiple symbols, results are added as number labels in miss symbol, Full Detonate PDU in window.
Emission	Every beam is drawn with unique colors, outline of each beam indicates a target in the target list of that beam. Detailed labels to show the modes of every beam. Range based on estimated detection range of a 1sq meter RCS target. Labels detail every system, and beam field in the PDU.
Transmitter	A shaded range ring. Color is driven by the frequency of the transmission. Size is proportional to the transmission power. The center symbol driven by the modulation type. Label shows Radio ID, frequency and status.
Signal	Encoded Audio (Voice) and Data Link symbology added to the entity that sends the Signal PDU. Label shows Radio ID, encoding type, and Tactical Data Link (TDL) type. Icons and symbology used to display Link16 information including J2 PPLI, J3 Surveillance tracks, J12.6 Target Sorting, J13.2 System Status, and J28.2 Text Messages.
IFF	Decoration symbology added to the entity transmitting the IFF. Label shows IFF system type and all IFF mode values.

Table 5 DISGUI PDU Types

DCDS Rule Set visualization is done in one of three modes.
1. Single or a few PDUs scripted that are expected to trigger a rule in the DCDS.
2. Systems are operating in controlled or small scale test mode.
3. Systems are operating in a full up mode in an operationally relevant scenario.

The DIS data in any of the three modes may be captured using a DIS data logger, and then be played back multiple times to allow visualization testing of rules as they are modified. Or it may be used to test notional rules for effectiveness. This iterative approach improves the likelihood of arriving at operational DCDS rules that are both effective at blocking the desired data while also passing the maximum amount of possible data to allow appropriate interactions between training participants on the high and low side of the security domains.

Audio Tool

The audio tool currently in use is DisCommWin and is a commercial product that converts encoded audio contained in the DIS protocol to actual audio. DisCommWin was chosen due to its availability in the testing and operational environments as well as its ease of configuration and user interface. Any product that provides this functionality could be used.

DisCommWin gives the DCDS team the ability to hear the encoded audio carried in DIS Signal PDUs on the low side security domain. The DCDS team uses the DisCommWin to verify the audio stream on the low side does not contain any sensitive high side data. If sensitive high side data is heard on the low side, then technical rules may need to be added or non-technical rules may need to be added or enforced.

RULE SET VERIFICATION TESTING

Verification of the rules involves four phases and four types of testing. The four phases of security testing include the following with each phase building on the other.

Phase 1 testing is unclassified and is conducted in an unclassified lab using an unclassified Rule Set. The objective of this testing is to verify compliance with configuration guidelines and to validate the operation of the integrated DCDS in an unclassified environment. Phase 1 configuration and functional test procedures provide for verification of the configuration, integration, and functioning of security features and

capabilities of the components contained in the DCDS in the unclassified environment.

Phase 2 internal testing in the DMON Operation Center focuses on functional testing using a classified Rule Set. Phase 2 testing is conducted using prepared test cases with known inputs and outputs or expected results and recorded play files. Since the expected results are known for the prepared test cases, these results are used for EAT tool configuration file verification purposes as well. The recorded play files are used to validate both the rules and the EAT verification tool. Results produced by EAT are validated against a proven manual process. The visualization tool DISGUI and the audio tool DisCommWin are used during Phase 2 for Rule Set verification purposes.

Phase 3 is single-level classified testing over the DMON between two Mission Training Center enclaves, one operating as the high side and one operating as the low side to emulate the cross domain environment (fake low). Phase 3 moves from the simulated DMON environment, where expected results and output are known before the test to an actual over-the-air test environment employing approved DMON participant sites. Phase 3 is where the EAT verification tool become imperative. Since the volume of data processed by the DCDS Controlled Interface is so large and the outputs are unknown at the time, processing this data manually is very time consuming and lends itself to errors. Using the EAT verification tool allows for almost instant and reliable verification of the rules process engine. DISGUI and DisCommWin tools can be used during Phase 3 for additional verification.

Depending on the complexity of the rule, multiple Phase 3 test events may be required to verify the rule and the EAT test tool configuration. Depending on the results of the Phase 3 testing efforts, additional subject matter expert input maybe required to validate that the rule implementation is consistent with the Rule Implementation Report. Once a number of successful test runs have been completed, verification efforts move to Phase 4.

From a rules testing prospective, the only difference between Phase 3 and Phase 4 testing is that the low side security domain is not a fake low, but a site actually operating in the actual low side security domain.

For each phase of testing, there are two primary categories of test that can be used. These two primary test categories include Controlled Interface Configuration Tests (CICTs) and Controlled Interface

Functional Tests (CIFTs). Configuration testing is used to examine the configuration of DCDS components for compliance with Department of Defense (DOD) or industry best practice configuration guides. Configuration test procedures are traced to Joint Air Force, Army, Navy (JAFAN) 6/3 Protection Level 3 security requirements and that mapping is documented in the DCDS Security Requirements Traceability Matrix. Configuration testing for DCDS is accomplished using the DCDS Controlled Interface Configuration Tests procedures. Configuration Test objectives are used to verify that all configuration related requirements as defined in the DCDS Security Requirements Traceability Matrix are implemented.

Functional verification testing demonstrates compliance with security and operational performance requirements and is the primary test used for rule verification. Functional test procedures are traced to JAFAN 6/3 requirements and that mapping is also documented in the DCDS Security Requirements Traceability Matrix. Functional testing for DCDS is accomplished using the DCDS Controlled Interface Functional Tests procedures. Functional test objectives include the following:

- Demonstrate that the DCDS Management System and DCDS Controlled Interface functions as specified in a given Rule Set.

- Demonstrate that all applicable rule processing security requirements as defined in the DCDS Security Requirements Traceability Matrix are implemented and operational.

- Demonstrate that the DCDS complies with all applicable risk management and Certification and Accreditation requirements outlined in JAFAN 6/3.

- Demonstrate that all DCDS functional requirements are satisfied.

System-level tests are performed to assess the operational performance of the DCDS in real-world or near real-world conditions. System-level testing is performed after configuration and functional tests are completed and the integrated DCDS configuration functions properly. System-level testing for DCDS is accomplished using the DCDS Controlled Interface Functional Test procedures.

Regression testing consists of configuration, functional, and if required system-level testing to assess the proper operation of DCDS components that have undergone modifications or upgrades including installation of vendor recommended patches.

Even though all four categories of testing are required to verify that the cross domain solution is functioning properly, the focus of the functional verification testing is to verify the Cross Domain Rule Set. There are five functional test sets used to verify a Rule Set and they are defined as the following:

CIFT1 – Functional rule tests include the following scenarios:
- DIS PDUs Passed Without Change – test procedures developed as part of this scenario demonstrate that PDUs that are allowed to pass as defined by the rule policy are passed in their entirety without error. PDUs that are included in this CIFT are defined in the applicable Rule Set.

- DIS PDUs Passed with Change – test procedures developed as part of this scenario demonstrate that some PDUs are modified before being allowed to pass from the high side to the low side. Specific changes to be made and the PDUs to be changed will be defined in the rule policy. PDUs that are included in this CIFT are defined in the applicable Rule Set.

- DIS PDUs Not Allowed to Pass or Blocked PDUs – test procedures developed as part of this scenario demonstrate that only explicitly identified PDUs are allowed to pass through the DCDS Controlled Interface from the high side to the low side. PDUs not explicitly allowed or that are always blocked are included in this CIFT as defined in the applicable Rule Set.

CIFT2 – Low-To-High – test procedures developed as part of this scenario demonstrate that DIS PDUs are allowed to pass through the DCDS Controlled Interface from low to high.

CIFT3 – Fail Open – test procedures developed as part of this scenario demonstrate that in the unlikely event that the DCDS Controlled Interface should fail, it will fail as an open circuit and no DIS PDU traffic will pass through the Controlled Interface while the Controlled Interface is in a failure state.

CIFT4 – Non-DIS PDU Traffic – verifies that non-DIS PDU Traffic is not allowed to pass through the DCDS Controlled Interface.

CIFT5 – High-To-Low – test procedures developed as part of this scenario demonstrate that during a typical mission training event only authorized PDUs as defined in the Rule Set are allowed to pass from the high side of the DCDS Controlled Interface to the low side of the Controlled Interface.

If a discrepancy is identified during any phase of testing that requires a system or software modification,

it is documented in a Problem Report and the test continues to the next step. When a testing cycle has been completed, Configuration Management provides an updated test baseline and all failed test scenarios are to be re-tested along with regression testing, as needed. The Test Lead is responsible for reviewing test activities, schedule, and problems, and for coordinating closure of discrepancy reports.

LESSONS LEARNED

Rule set development is normally a spiral process, and as such, use of automated tools makes this a much more effective and efficient process. In the past when much of the analysis had to be done manually, the time required to develop and verify a rule was much longer and lent itself to small accounting errors that in some cases were very difficult to identify. Much of the difficulty was purely based on the volume of data that had to be manually reviewed. Tools like the EAT and DISGUI have greatly improved this process both in efficiency and effectiveness. With these tools, data comparisons between the high side security domain, DCDS Controlled Interface, and low side security domain can be performed quickly and accurately.

Voice communications between the high side security domain and the low side security domain need to be monitored during test events and training events. Even though there are non-technical rules that address these voice communications, it is important that some type of verification be performed. Currently the most effective method for performing this verification is by having subject matter experts listen in during test and training events. These subject matter experts could, based on past experience in some cases, anticipate what pilots may say and circumvent these conversations, as required. Just the fact that pilots know someone will be listening to their verbal exchange of information in the cockpit should reduce the probability that program information could be inadvertently revealed to the low side security domain. Additionally, the subject matter expert could provide an audit function in case something was communicated between the high side security domain and the low side security domain that should not have occurred.

CONCLUSIONS

Designing, developing, and verifying a Cross Domain Rule Set is a multi step process that involves input and participation from all stake holders. Rule set development is also a spiral process and the number of iterations to the process is driven by the complexities of the Rule Set, accurate and timely information

regarding the program specific data that must be protected, availability of subject matter experts, coding complexities, and verification complexities.

In the past, Rule Set verification efforts have focused on analysis of PDU statistical data provided by the DCDS Controlled Interface and did not really consider all aspects including visual and audio effects. Statistical information is important; however, it does not show the cause and effect of modifications to the DIS PDU data on a simulation event. Tools like DISGUI give a clear visual representation of how changes to DIS PDU data between the high side security domain and the low side security domain effect the simulation both on the high and low side of the training event.

In order to effectively verify and validate a DCDS Rule Set, all aspects of the low side security domain environment must be considered. This includes PDU content and analysis, visual aspects of simulator and other participants, and aural aspects of low side security domain. It is imperative that tools are available that allow each of these aspects to be analyzed individually as there is too much data when viewed as a whole to thoroughly validate the Rule Set. In short, it is not feasible to successfully verify and validate a Rule Set by simply observing the low side security domain during a test event.

REFERENCES

Aldinger, M. & Keen, S. (2007), CAF DMO Standards-Based Approach for Achieving M&S Interoperability. Paper presented at NATO RTO Symposium on "*Improving M&S Interoperability, Reuse and Efficiency in Support of Current and Future Forces*," September 2007.

Danner, B., Muckenhirm, C., Valle, T., McElveen, C., Bragdon-Handfield, J., & Colegrove, A. (2002). Multilevel Security Feasibility in the M&S Training Environment. Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2002, Paper 167.

Danner, B., & Valle, T. (2005). Multilevel Security Assessment for the Distributed Mission Operations Network (DMON). Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2005, Paper 2165.

Danner, B., Valle, T., & McGregor, B. (2006). A DMON Cross Domain Solution (CDS) for Recurring Team Training. Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2006, Paper 2775.

Danner, B., & Djahandari, K. (2008). Cross Domain Solution Policy, Management, and Technical Challenges. Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2008, Paper 8343.

Danner, B. (2009). Cross Domain Solution (CDS) Certification and Accreditation for Persistent, Simulation Training. Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2009, Paper 9142.

Djahandari, K., Archer, J., & Danner, B. (2009). Cross Domain Solution Challenges Transitioning From Concept to Operations. Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2009, Paper 9133.

DMT O&I Contractor (2010). Combat Air Force Distributed Mission Operations (CAF DMO) Network Users Guide, Version 2.0. Retrieved May 26, 2010, from
https://secure.dmodmt.com/document.cfm?id=2248.

Johnson, W. (2008) Combat Air Force Distributed Mission Operations: Immersion Into Daily Training. *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2008,* Paper 8008.

DoD (2004). *Joint Air Force, Army, Navy (JAFAN) 6/3 Manual* (FOUO).